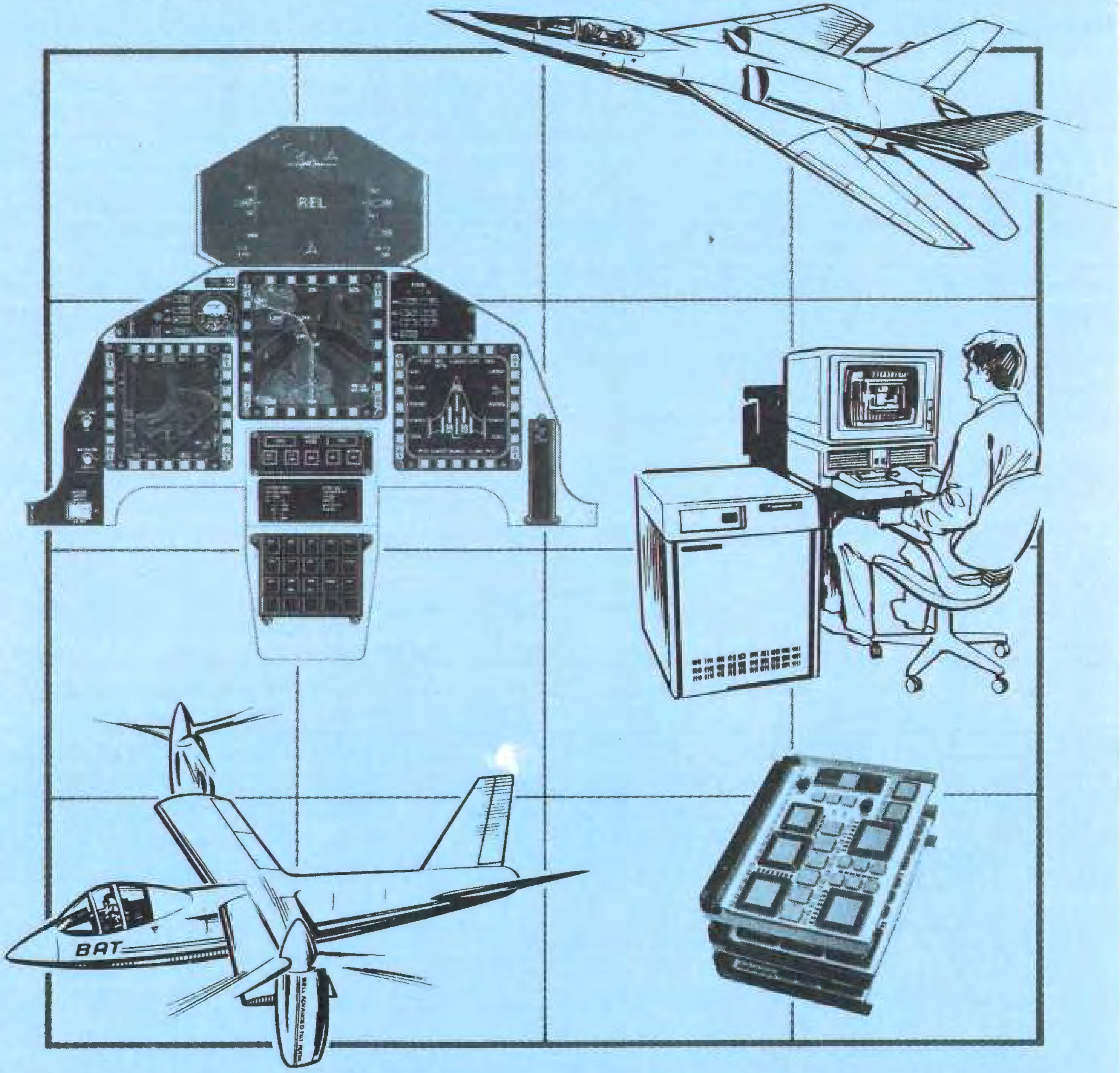
AIAA/IEEE 6TH DIGITAL AVIONICS SYSTEMS CONFERENCE



For permission to copy or republish,
contact the American Institute of Aeronautics and Astronautics,
1633 Broadway, New York, NY 10019

PROCEEDINGS

OF THE

AIAA/IEEE

6th

DIGITAL AVIONICS

SYSTEMS CONFERENCE, 6th, 1984.

DECEMBER 3-6, 1984/BALTIMORE, MARYLAND

AMERICAN INSTITUTE OF
AERONAUTICS AND ASTRONAUTICS
- TECHNICAL COMMITTEE
— DIGITAL AVIONICS

INSTITUTE OF ELECTRICAL
AND ELECTRONIC ENGINEERS
- AEROSPACE AND ELECTRONICS
SYSTEMS SOCIETY

THE CONFERENCE AT A GLANCE

MONDAY, DECEMBER 3	TUESDAY, DECEMBER 4	WEDNESDAY, DECEMBER 5	THURSDAY, DECEMBER 6
9:00-12:00 TUTORIALS A. Introduction to Digital Avionics. (307) B. Introduction to **ADA**™. (308) C. Digital Signal Processing. (309)	9:00-12:00 PLENARY SESSION TOPIC: Digital Avionics Requirements for the 1990s SPEAKERS: B.A. Zempolich P.O. Brown R.N. Longuemare E.C. Machacek	9:00-12:15 TECHNICAL SESSIONS 6. Systems and Software-Development and Evaluation Tools. (307) 7. General Aviation Avionics. (301) 8. Fault Tolerant Avionics. (308) 9. Signal Processing. (310) 10. Crew Systems-Advanced Control and Display Technology. (309)	9:00-12:15 TECHNICAL SESSIONS 16. Systems and Software-**ADA**™. (307) 17. On-Board Monitoring and Test. (301) 18. VLSI Design and Testing. (308) 19. Data Link Systems Applications. (309) 20. Merged Digital Map Techniques. (310)
OPEN	12:00-2:00 LUNCHEON Dutch Treat in Exhibit Area (Hall C)	12:15-2:15 LUNCHEON Digital Avionics Award (Hall C)	12:15-2:15 LUNCHEON Speaker - Jack Jackson (Hall C)
2:00-5:00 TUTORIALS D. Voice Interactive Systems Applications and Implementation. (307) E. **ADA**™ for Project Managers. (308) F. Overview of Artificial Intelligence. (309)	2:15-5:30 TECHNICAL SESSIONS 1. Systems and Software-Development Methods. (307) 2. Commercial Transport Avionics. (301) 3. Advanced Avionic Sensor Systems. (308) 4. Crew Systems-Human Factors-Artificial Intelligence. (309) 5. Digital Flight Controls. (310)	2:15-5:30 TECHNICAL SESSIONS 11. Systems and Software-Verification and Test Techniques. (307) 12. Rotorcraft Avionics. (301) 13. Data Bus-Concepts and Practices. (308) 14. Crew Systems-Systems, Development and Integration. (309) 15. Communication, Navigation, and Identification Terminals. (310)	2:15-5:30 TECHNICAL SESSIONS 21. Advanced Digital Integrated Circuits. (307) 22. Airborne Separation Assurance. (308) 23. The All Electric Airplane. (309) 24. Standardized Modular Avionics. (310) 25. Digital Propulsion Control and Monitoring. (301)
OPEN	EVENING	EVENING	EVENING
	6:00-8:00 EXHIBITORS RECEPTION Hors d'Oeuvres Cash Bar (Hall C)	6:00-8:00 PANEL DISCUSSIONS A. National Airspace Systems Plan. (307) B. VHSIC Insertion and Its Implications. (309)	CRAB FEAST Relaxed Evening Casual Attire
	12:00-8:00 EXHIBITS OPEN	9:00-8:00 EXHIBITS OPEN	9:00-2:00 EXHIBITS OPEN

Ada is a registered trademark of the U.S. Government, Ada Joint Program Office

TECHNICAL SESSIONS

SYSTEMS AND SOFTWARE — DEVELOPMENT METHODS 1

COMMERCIAL TRANSPORT AVIONICS 2

ADVANCED AVIONIC SENSOR SYSTEMS 3

CREW SYSTEMS — HUMAN FACTORS AND ARTIFICIAL INTELLIGENCE 4

DIGITAL FLIGHT CONTROLS 5

SYSTEMS AND SOFTWARE — DEVELOPMENT AND EVALUATION TOOLS 6

GENERAL AVIATION AVIONICS 7

FAULT TOLERANT AVIONICS 8

SIGNAL PROCESSING 9

CREW SYSTEMS — ADVANCED CONTROL/DISPLAY TECHNOLOGY 10

SYSTEMS AND SOFTWARE — VERIFICATION AND TEST TECHNIQUES 11

ROTORCRAFT AVIONICS 12

DATA BUS CONCEPTS AND PRACTICES 13

CREW SYSTEMS — SYSTEMS DEVELOPMENT AND TEST 14

COMMUNICATION, NAVIGATION, AND IDENTIFICATION (CNI) TERMINALS 15

SYSTEMS AND SOFTWARE — ADA™ 16

ON-BOARD MONITORING AND TEST 17

VLSI DESIGN AND TESTING 18

DATA LINK SYSTEMS APPLICATIONS 19

MERGED DIGITAL MAP TECHNIQUES 20

ADVANCED DIGITAL INTEGRATED CIRCUITS 21

AIRBORNE SEPARATION ASSURANCE 22

ALL ELECTRIC AIRCRAFT 23

STANDARDIZED MODULAR AVIONICS 24

DIGITAL PROPULSION CONTROL AND MONITORING SYSTEMS 25

Exhibitors

ALLIED BENDIX AEROSPACE

APPLIED SCIENCE LABS

ARINC RESEARCH CORP.

BOEING AEROSPACE CO.

CANADIAN MARCONI CO.

TABLE OF CONTENTS

SESSION 1: SYSTEMS AND SOFTWARE — DEVELOPMENT METHODS

Chairmen:

G. Henley
Intermetrics Inc.,
Cambridge, MA

P.E. Gartz
Boeing Commercial Airplane Co.
Seattle, WA

AIAA-84-2598-CP
Developing Safe Software for Critical Airborne Applications 1
W.B. Noble, *Hughes Aircraft Co., Ground Systems Group*

AIAA-84-2595-CP
The Use of Structured Methods in the Development of Large Software-Based Avionics Systems 6
D.J. Hatley, *Lear Siegler Inc., Instrument Avionics Div.*

AIAA-84-2600-CP
A Conceptual Design Methodology Used to Design a Support Computer Program 16
J.A. Mozier, D.J. Berg and R.L. Young, *McDonnell Douglas Corp.,*

SESSION 2: COMMERCIAL TRANSPORT AVIONICS

Chairmen:

R.L. Heimbold
Lockheed Corp.
Burbank, CA

J.H. Shannon
Douglas Aircraft Co.
Long Beach, CA

AIAA-84-2602-CP
Digital Fuel Quantity Indicating System for Aircraft 21
R.L. Newport, D.J. Nelson and M.T. Manfred, *Honeywell Inc.*

AIAA-84-2603-CP
Architectural Solutions to Safety Problems of Digital Flight-Critical Systems for Commercial Transports 28
L.J. Yount, *Sperry Corp.*

AIAA-84-2604-CP
Applications of Air-Ground-Air Digital Communication System ACARS/AIRCOM 36
C.A. Bennett, *Teledyne Controls*

AIAA-84-2605-CP
An Overview of the Digital Avionics Assessment Activities Being Conducted by the Federal Aviation Administration at NASA-Ames Research Center 40
W. Larsen, *NASA Ames Research Center,* and D. Eldredge and E. Hitt, *Battelle Columbus Labs,* and D. Mulcare and W. Ness, *Lockheed-Georgia Co.*

AIAA-84-2606-CP
Recent Advances in Aircraft On-Board Weight and Balance Systems 47
J.P. O'Brien, *Sundstrand Data Control Inc.*

AIAA-84-2607-CP
TIME FLIES, An In-Service Evaluation of a 4-D Flight Management System 54
D.A. Moor, *Lockheed-California Co.*

SESSION 3: ADVANCED AVIONIC SENSOR SYSTEMS

Chairmen:

E.A. Hopper
Westinghouse
Dayton, OH

R.B. Longbrake
ASD/ENA
Wright-Patterson AFB, OH

AIAA-84-2613-CP
Imaging Sensor Autoprocessor 60
S.C. Sawtelle and R.J. Jennewine, *AFWAL/AARI, Wright-Patterson AFB*

AIAA-84-2610-CP
Infrared Search and Track System Development 65
W.E. Moore, M. Spector and V.D. Best, *AFWAL/AART, Wright-Patterson AFB*

AIAA-84-2609-CP
The APG-66 Radar and Its Derivative Applications 68
L.J. Kuchinski and T.R. Patton, *Westinghouse Electric Corp.*

AIAA-84-2614-CP (Alternate)
Integrated Tracking Software for Multimode Operation 76
J.L. Farrell and D.A. Hedland, *Westinghouse Electric Corp.*

Exhibitors

CIRCUIT TECHNOLOGY INC.

COLLINS AIR TRANSPORT DIVISION,
ROCKWELL INTERNATIONAL

COLLINS GOVERNMENT AVIONICS DIVISION,
ROCKWELL INTERNATIONAL

CONTROL DATA

DELCO SYSTEMS OPERATIONS,
GENERAL MOTORS CORP.

SESSION 4: CREW SYSTEMS: HUMAN FACTORS AND ARTIFICIAL INTELLIGENCE

Chairmen:

R.N. Bretoi
NASA Ames Research Center
Moffett Field, CA

D. Cole
Air Force Aerospace Medical
Reesearch Lab.
Wright-Patterson AFB, OH

AIAA-84-2616-CP
The Evaluation of Display Symbology: A Chronometric Study of Visual Search 84
R. Remington, *NASA Ames Research Center,* and D. Williams, *Psycho Linguistics Institute*

AIAA-84-2617-CP
Artificial Intelligence Implications for Advanced Pilot/Vehicle Interface Design 90
K.J. Maxwell and J.A. Davis, *General Dynamics Fort Worth Div.*

AIAA-84-2619-CP
Model-Based Reasoning in Expert Systems: An Application to Enroute Air Traffic Control 95
S.E. Cross, *Air Force Institute of Technology, Wright-Patterson AFB*

AIAA-84-2620-CP
Mission Scenarios for Cockpit Automation Technology 102
G.G. Kuperman and P.V. Kulwicki, *Air Force Aerospace Medical Research Lab., Wright-Patterson AFB*

SESSION 5: DIGITAL FLIGHT CONTROLS

Chairmen:

C.R. Abrams
Naval Air Development Center
Warminster, PA

G. Vetsch
McDonnell Douglas Corp.
St. Louis, MO

AIAA-84-2622-CP
Backup Modes for the F/A-18 Digital Flight Control System 108
H.E. Harschburger, B. Glaser and J.R. Hammel, *McDonnell Douglas Corp.*

AIAA-84-2623-CP
Integrated Flight Control/Navigation Sensors 116
R.E. Ebner and S.Y. Wei, *Litton Systems Inc.*

AIAA-84-2624-CP
A Highly Monitored Digital Flight Control System for the AV-8B Harrier II 124
G.G. Gaston, *McDonnell Aircraft Corp.,* and L.E. Costlow and J.R. Perry, Jr., *Sperry Flight Systems*

AIAA-84-2625-CP
Multivariable Control System Design Techniques: An Application to a Short Take Off and Landing Aircraft 132
R.J. Landy and D.B. Kim, *McDonnell Aircraft Co.*

AIAA-84-2627-CP
Aircraft Automatic Digital Flight Control System with Inversion of the Model in the Feed-Forward Path 140
G.A. Smith and G. Meyer, *NASA Ames Research Center*

SESSION 6: SYSTEMS AND SOFTWARE — DEVELOPMENT AND EVALUATION TOOLS

Chairmen:

E. Waltz
Bendix Communications Div.
Baltimore, MD

D. Brown
General Dynamics
Fort Worth, TX

AIAA-84-2629-CP
Development of a Support Software System for Real-Time HAL/S Applications 151
R.S. Smith, *NASA Langley Research Center*

AIAA-84-2633-CP
Software Traceability, Requirements Testability, and Auditing Model 159
J. Sciortino and D. Dunning, *Arinc Research Corp.*

AIAA-84-2635-CP
Development Tools - Case Study for Large Systems 167
K. Hornbach, *Lear-Siegler/Instrument Div.*

AIAA-84-2631-CP
Multi-Sensor Surveillance Performance Evaluation Tool............................ 175
C.L. Bowman and G. Snashall, *Verac Inc.*

SESSION 7: General Aviation Avionics

Chairman:

M.W. Kelly
Beech Aircraft Corp.
Wichita, KS

AIAA-84-2637-CP
General Aviation Data Bus Update 180
D.L. Stanislaw, *Cessna Aircraft Co.*

Exhibitors

GENERAL DYNAMICS, FORT WORTH DIVISION

GENERAL ELECTRIC CO.

ILC DATA DEVICE CORP.

ITT AVIONICS DIVISION

LEAR SIEGLER INC.

AIAA-84-2639-CP
Systems Concept for Speech Technology Application in General Aviation 184
R.A. North, *Honeywell Systems Research Center* and H. Bergeron, *NASA Langley Research Center*

AIAA-84-2641-CP
Effects of Precipitation Static on General Aviation Digital Avionics 190
L.O. Hendry, *Beech Aircraft Corp.*

AIAA-84-2642-CP
Current and Future General Aviation EFIS Developments 193
J.A. Ogann, *Rockwell International*

SESSION 8: Fault Tolerant Avionics

Chairmen:

B.L. Dove
NASA Langley Research Center
Hampton, VA

D.B. Mulcare
Lockheed-Georgia Co.
Marietta, GA

AIAA-84-2644-CP
Advanced Information Processing System 199
J.H. Lala, *Charles Stark Draper Lab.*

AIAA-84-2646-CP
A Fault-Tolerant Software Strategy for Digital Systems 211
E.F. Hitt and J.J. Webb, *Battelle Columbus Labs*

AIAA-84-2647-CP
Evaluation Methodologies for an Advanced Information Processing System 217
R.S. Schabowsky Jr., E. Gai, P. Motyka, J.H. Lala, and B.K. Walker, *Charles Stark Draper Lab.*

AIAA-84-2648-CP
Validation Methodology for Fault Tolerant Clock Synchronization 225
S.C. Johnson and R.W. Butler, *NASA Langley Research Center*

AIAA-84-2649-CP (Alternate)
Test Experience on an UltraReliable Computer Communication Network 233
L. Abbott, *NASA Ames Research Center/Dryden Flight Research Facility*

AIAA-84-2650-CP (Alternate)
UltraReliable Fault-Tolerant Control Systems 239
L.D. Webster, R.A. Slykhouse, L.A. Booth, T.M. Carson, G. J. Davis, and J.C. Howard, *NASA Ames Research Center*

SESSION 9: Signal Processing

Chairmen:

G.J. Palatucci
U.S. Naval Air Development Center
Warminster, PA

B. Bjerede
Linkabit Corp.
San Diego, CA

AIAA-84-2651-CP
Signal Processing Using Digital Lattice Filters 247
L. Chin, *Naval Air Development Center*

AIAA-84-2652-CP
Application of Digital Signal Processing to a Low Rate Communications Receiver 252
R.G. Henderson and P. Lafrance, *The MITRE Corp.*

AIAA-84-2653-CP
The Impact of Advanced Computational Techniques on Digital Avionics Signal Processor Performance 257
W.M. Vojir, *Grumman Aerospace Corp.*

AIAA-84-2654-CP
Self Adaptive Filtering of Environmental Noises from Speech 263
D. Graupe, J. Grosspietsch and S. Basseas, *IntelliTech Inc.*

AIAA-84-2655-CP
A Frequency Hopped Demand Assigned Multiple Access Scheme 270
K. Joseph and S.S. Rappaport, *State Univ. of New York*

AIAA-84-2656-CP
Enhanced TCAS-II Signal Processor Development 277
B.J. Lyons and M.A. Martin, *Bendix Communications Div.*

SESSION 10: CREW SYSTEMS: ADVANCED CONTROL/DISPLAY TECHNOLOGY

Chairman:

J.J. Hatfield
NASA Langley Research Center
Hampton, VA

AIAA-84-2657-CP
Software Controlled Display/Pushbutton 285
D.A. Nicholson, *MICRO SWITCH Div. of Honeywell* and R.J. Spiger, *Boeing Aerospace Co.*

Exhibitors

LITTON SYSTEMS CANADA, LTD.

MCDONNELL DOUGLAS ASTRONAUTICS, CO.

NORDEN SYSTEMS

PLESSEY AVIONICS

SMITHS INDUSTRIES AEROSPACE
AND DEFENSE SYSTEMS, INC.

AIAA-84-2663-CP
Recent Advances in Electroluminescent Displays Applicable to Future Crew-Station Interfaces 291
M.R. Miller and E. Schlam, *U.S. Army-ERADCOM*, and J.B. Robertson and J.J. Hatfield, *NASA Langley Research Center*

AIAA-84-2659-CP
α-Si Addressed Liquid Crystal Aircraft Display 296
D.E. Castleberry, J.E. Bigelow, H.G. Parks, W.W. Piper and G.E. Possin, *General Electric Co.*

AIAA-84-2660-CP
Applications of Voice Interactive Systems - Military Flight Test and the Future 301
C.A. Moore and R.D. Moore, *VERAC Inc.*, and J.C. Ruth, *McDonnell Douglas Electronics Co.*

SESSION 11: SYSTEMS AND SOFTWARE VERIFICATION AND TEST TECHNIQUES

Chairmen:

E.A. Delanty
The Boeing Co.
Seattle, WA

J.G. Weber
VERAC Inc.
San Diego, CA

AIAA-84-2669-CP
Verification Techniques for Improving Software Quality Through Automated Requirements Data Bases 309
G. Lueders, *Sperry Corp., Flight Systems Div.*

AIAA-84-2667-CP
Automated Benchmark Generation Based Upon a Specification Language 312
N. Rajan, S.E. Feteih, *Stanford Univ.*, and J. Saito, *NASA Ames Research Center*

AIAA-84-2665-CP
Automated Software Test System for The 737-300 Flight Management Computer 319
S.C. Runo, *The Boeing Co.*

AIAA-84-2664-CP
A Method for Testing a Digital Flight Control System Without the Use of Ground Support Equipment 325
H.E. Hansen, *McDonnell Aircraft Co.*

AIAA-84-2666-CP
Real Time Data Processing for Avionics Testing on the A-6E 332
P.T. Richards and J. Lehmann, *Grumman Data Systems Corp.*

AIAA-84-2670-CP (Alternate)
Safety of Flight and Qualification Testing for Avionic Systems 339
W.J. Hall, Jr., *General Dynamics Fort Worth Div.*

AIAA-84-2726-CP
Executable Assertions and Flight Software 346
A. Mahmood, D.M. Andrews and E.J. McCluskey, *Center for Reliable Computing, Stanford Univ.*

SESSION 12: Rotorcraft Avionics

Chairmen:

D.G. Denery
NASA Ames Research Center
Moffett Field, CA

G. Stech
Avionics Research and Development Activity
Fort Monmouth, NJ

AIAA-84-2672-CP
Integrated Avionics for Advanced Army Rotorcraft 352
W.L. Eversole, W.F. Kiczuk, J.A. Lambrecht, R.L. Rivard, J.J. Williams, and R.E. Branstetter, *Texas Instruments Inc.*

AIAA-84-2673-CP
Evaluation of a Real-Time Predictive Guidance Law for Landing VTOL Aircraft at Sea 359
C.H. Paulk Jr., *NASA Ames Research Center*, and A.V. Phatak, *Analytical Mechanics Associates Inc.*

AIAA-84-2674-CP
Digital Avionics and Flight Path Director Functions of the HH-60 Helicopter 372
F.G. Kilmer, R.L. Kilmer and B.A. Thalacker, *IBM Federal Systems Div.*

AIAA-84-2676-CP
Application of Differential GPS to Civil Helicopter Terminal Guidance 379
R.P. Denaro, *TAU Corp.*

AIAA-84-2621-CP
Pilot Command Interfaces for Discrete Control of Automated Nap-of-the-Earth Flight 386
S.J. Mountford, R. Penner and P. Bursch, *Honeywell SRC/MMS*

Exhibitors

ROLM CORP.

SPERRY DEFENSE SYSTEMS

SUNDSTRAND DATA CONTROL, INC.

TRW ELECTRONICS AND DEFENSE

VERAC, INC.

SESSION 13: DATA BUS CONCEPTS AND PRACTICES

Chairmen:

R.M. Hartman
McDonnell Douglas Astronautics Co.
Huntington Beach, CA

M.A. Geyer
Westinghouse Electric Corp.
Baltimore, MD

AIAA-84-2679-CP
Standard Computer Bus for MIL-STD-1750A Avionics Computer 393
S. Levy and D. Penn, *Israel Aircraft Industries, and E. Loker, Elbit Computers Ltd.*

AIAA-84-2681-CP
A Wavelength Division Multiplexed (WDM) Optical Data Bus for Future Military Applications 399
K.K. Chow and W.B. Leonard, *Lockheed Palo Alto Research Lab.*, and T.C. Yang, *Feng Chia Univ.*

AIAA-84-2682-CP
Data Bus Interface MIL-STD-1553B/ARINC 561 405
J. Spieth, *ASD, U.S. Air Force*

AIAA-84-2757-CP
AI Concepts in MUX BUS Control: Layered Reservation Access 412
B.J. MacNamera, *ARINC Research*

AIAA-84-2756-CP
High-Speed Bus Structures for Multiprocessing 417
Q.E. Dolecek, *The Johns Hopkins Univ.*

SESSION 14: CREW SYSTEMS: SYSTEMS DEVELOPMENT AND TEST

Chairman:

W.D. Smith
Boeing Commercial Airplane Co.
Seattle, WA

AIAA-84-2686-CP
An Advanced Media Interface for Control of Modern TransportAircraft Navigational Systems 421
D.R. Jones, R.V. Parrish and L.H. Person, *NASA Langley Research Center*, and J.L. Old, *Research Triangle Institute*

AIAA-84-2687-CP (Alternate)
A Simulator Application of a Hands-On Throttle and Stick Concept to a Transport Pilot/Autopilot Interface 429
A.M. Busquets, R.V. Parrish and T.W. Hogge, *NASA Langley Research Center*

AIAA-84-2688-CP
Micro-Based Control and Display Research 435
J.M. Reising, G.D. Lizza and D.A. Sobota, *AFWAL/FIGR, Wright-Patterson AFB*

AIAA-84-2689-CP
Certification of a Holographic Head-Up Display for Low Visibility Landings 441
J.P. Desmond and D.W. Ford, *Flight Dynamics Inc.*

SESSION 15: COMMUNICATION, NAVIGATION, AND IDENTIFICATION (CNI) TERMINALS

Chairmen:

D.B. Cox, Jr.
C.S. Draper Lab.
Cambridge, MA

F.W. Smead
ITT Avionics
Nutley, NJ

AIAA-84-2693-CP
Customizing ICNIA for Specific Applications .... WITHDRAWN
F.W. Smead, *ITT Avionics*

AIAA-84-2691-CP
An Experimental Aeronautical Satellite Data Link 454
S. Anderson, *The MITRE Corp.*

AIAA-84-2698-CP (Alternate)
An Adaptive Airborne VLF Communications Array Concept 462
S.R. Schmitt, *Naval Air Development Center*

SESSION 16: SYSTEMS AND SOFTWARE: ADA™

Chairmen:

D.B. Mulcare
Lockheed-Georgia Co.
Marietta, GA

T.V. McTigue
McDonnell Aircraft Co.
St. Louis, MO

AIAA-84-2701-CP
Directed Graph Methodology© 467
E.M. Lanier and M.E. Hinkey, *Westinghouse Electric Corp.*

Exhibitors

W.W. GAERTNER RESEARCH INC.

SCI SYSTEMS, INC.

* AMERICAN ASSOCIATION FOR
THE ADVANCEMENT OF SCIENCE

DIGITAL TECHNOLOGY, INC.

* IEEE-AEROSPACE
AND ELECTRONIC SYSTEMS SOCIETY

AIAA-84-2702-CP
Real Life Considerations of ADA™ Runtime Organizations for Real-Time Applications 472
J.M. Kamrad II, *Alsys*

AIAA-84-2703-CP
Using Ada™ for a Distributed, Fault-Tolerant System 477
J.B. DeWolf, N.M. Sodano and R.S. Whittredge, *Charles Stark Draper Lab. Inc.*

SESSION 17: ON-BOARD MONITORING AND TEST

Chairmen:

L.M. Carrier Jr.
Rockwell International Corp.
Lakewood, CA

D. Pieratt
ASD/BIEE
Wright-Patterson AFB, OH

AIAA-84-2706-CP
Maintenance Assist Functions Embedded Within the 737-300 Flight Management System 485
G.F. Ellis, *The Boeing Co.*, and H.E. Hofferber, *Sperry Flight Systems*

AIAA-84-2709-CP
Integrated On-Board Monitoring and Test Concepts 492
P.C. Jenkins and C.J. Ong, *Rockwell International Corporation*

AIAA-84-2708-CP
Avionics Hardware Design for Testability 498
B.L. Ferrell and S.L. Over, *General Dynamics Fort Worth Div.*

AIAA-84-2677-CP
Remote Maintenance Monitoring Using a Digital Data Link 503
D. Dowling and R.A. Lancaster, *ARINC Research Corp.*

AIAA-84-2705-CP
Crash Survivable Flight Data Recorder 508
G.E. Davis, *General Dynamics Fort Worth Div.*

SESSION 18: VLSI DESIGN AND TESTING

Chairman:

C.H. Huang
Lockheed Research and Development Div.
Palo Alto, CA

AIAA-84-2712-CP
Semi-Custom Devices and Design 513
C.W. Stern, *Silicon Compilers Inc.*

AIAA-84-2711-CP
Gate Array, Standard Cell, and Fully Custom: Building Blocks for the Digital Avionics Designer 517
H.L. Owen III and M.T. Kopp, *Georgia Institute of Technology*

AIAA-84-2713-CP
VLSI Design for Testability 523
E.J. McCluskey, *Center for Reliable Computing, Stanford Univ.*

SESSION 19: DATA LINK SYSTEMS APPLICATIONS

Chairmen:

D.G. Botha
AFWAL/AAAI
Wright-Patterson AFB, OH

D.G. Evans
PME/PMA
Washington, DC

AIAA-84-2716-CP
The Navy's Tactical Data Network 531
W.J. Schoppe, *Naval Air Development Center*

AIAA-84-2719-CP
Implications of JTIDS/TADIL J on Navy Aircraft Weapons Systems 539
R.W. Kuhn, *McDonnell Aircraft Co.*

SESSION 20: MERGED DIGITAL MAP TECHNIQUES

Chairman:

J.W. Weber
Hughes Aircraft Co.
Los Angeles, CA

AIAA-84-2724-CP
Data Compression Techniques Digital Map ... 549
P.L. Poehler, *DBA Systems Inc.*

AIAA-84-2723-CP
A Terrain Data/Digital Map System for LHX ... 557
G.O. Burnham, C. Benning and R.L. Rivard, *Texas Instruments Inc.*

SESSION 21: ADVANCED DIGITAL INTEGRATED CIRCUITS

Chairmen:

D.B. McBrayer
LTV Vought Missiles and Advanced Programs
Dallas, TX

W.R. Hutchins
Sanders Associates
Nashua, NH

AIAA-84-2728-CP
MIL-STD-1553 Dual Redundant Remote Terminal Superhybrid 563
S.N. Friedman, *ILC Data Device Corp.*

Exhibitors

FAIRCHILD INDUSTRIES

AIAA-84-2730-CP
VLSI Chip Set for High-Performance Avionic Computers 569
S.J. Forde and M.A. Hilmantel, *Sanders Associates Inc.*

SESSION 22: AIRBORNE SEPARATION ASSURANCE

Chairmen:

F.S. Chandler
Sperry Dalmo Victor Inc.
Belmont, CA

J.J. Fee
Federal Aviation Administration
Washington, DC

AIAA-84-2734-CP
Evolution of the Collision Avoidance System in the Cockpit 573
W.L. Hyland, *Federal Aviation Administration*

AIAA-84-2738-CP
Enhanced TCAS-II Tracking Accuracy 577
A.I. Sinsky and J.E. Reed, *Allied Bendix Aerospace, Bendix Communications Div.*, and J. Fee, *Federal Aviation Administration*

AIAA-84-2737-CP
New Concepts in Collision Avoidance Logic . . 586
W.D. Love and A.D. Zeitlin, *The MITRE Corp.*

AIAA-84-2736-CP
Improved TCAS I for Pilot Warning Indication 593
J.D. Welch and W.H. Harman, *M.I.T. Lincoln Lab.*

AIAA-84-2735-CP
Secondary Surveillance Radar Mode S - Potential Applications and Future Requirements in Europe 597
R.C.G. Jenyns and M.E. Cox, *European Organization for the Safety of Air Navigation*

SESSION 23: ALL ELECTRIC AIRCRAFT

Chairman:

C.R. Spitzer
NASA Langley Research Center
Hampton, VA

AIAA-84-2739-CP
Deja-Vu of All Electric/All Digital Aircraft 606
M.J. Cronin, *Lockheed-California Corp.*

AIAA-84-2740-CP
An Assessment of the 1990 Digital/Electric Airplane 615
G.E. Tagge, *Boeing Commercial Airplane Co.*

AIAA-84-2742-CP
A Minimum Interruption AC Electric Power Generating System for Avionics 619
B.R. Mehl and G.R. Pierce, *Sundstrand*

SESSION 24: STANDARDIZED MODULAR AVIONICS

Chairmen:

R.K. Ricker
USAF
Wright-Patterson AFB, OH

J.A. Wyatt
Dept. of Defense (OUSDRE)
Falls Church, VA

AIAA-84-2745-CP
Common Modular Avionics: Partitioning and Design Philosophy 624
D.M. Scott and S.P. Mulvaney, *General Dynamics Corp.*

AIAA-84-2747-CP
Modular Standards for Emerging Avionics Technologies 629
W.E. Radcliffe, *ARINC Research Corp.*, and D.J. Boaz, *Air Force Avionics Lab., Wright-Patterson AFB*

AIAA-84-2749-CP
High Density Modular Avionics Packaging 634
F.J. Poradish, *Texas Instruments Inc.*

AIAA-84-2746-CP
Modular Avionics Packaging Standardization 641
M.L. Austin and J.K. McNichols, *Naval Avionics Center*

AIAA-84-2748-CP
General Dynamics Convair Division Totally Reconfigurable Embedded Computer 646
L.G. Markert, J.L. Kusek, and P.A. Hedtke, *General Dynamics Convair Div.*

SESSION 25: DIGITAL PROPULSION CONTROL AND MONITORING SYSTEMS

Chairmen:

J.C. Richards
General Electric Co.
Cincinnati, OH

P. Adams
AFWAL/POTC
Wright-Patterson AFB, OH

AIAA-84-2753-CP
Accuracies of Digital Engine Monitoring Proven by Performance 653
J. Bluish, *Allied Bendix Aerospace*, and J. Balazic, *Boeing Commercial Airplane Co.*

AIAA-84-2754-CP
The F110 Engine Monitoring System 661
M.J. Ashby and R.J.E. Dyson, *General Electric Co.*

6TH
DASC

MESSAGE FROM THE CHAIRMAN

IRVING R. REESE
Boeing Commercial
Airplane Co.

Welcome to Baltimore and the AIAA/IEEE 6th Digital Avionics Systems Conference, otherwise known as the "6th DASC." It has been only one short year since the 5th DASC in Seattle. Yet, the 6th DASC will have even more technical sessions, panel discussions and exhibits than ever before. This, I believe, is indicative of the exciting progress we are experiencing in digital avionics.

Digital avionics are contributing to higher performance, new mission capabilities, improved crew interface and greater reliability for both military and civil aircraft. Several of these realized benefits were reported at the 5th DASC in Seattle last year. However, this is not a time to rest on our laurels. Even greater challenges and opportunities lie ahead.

Randy Moore and the Technical Program Committee have prepared a full agenda of tutorials, technical sessions and expert panel discussions designed to educate, share information and stimulate debate.

An outstanding array of technical exhibits will provide a rich opportunity to see, hear and operate state-of-the-art equipment and components. You would need a large travel budget to see even a fraction of this technology in its normal environment...so plan to spend a few hours browsing and talking with exhibitors.

The technical and exhibits programs are complemented by social events where you can visit with your colleagues in a relaxed atmosphere. A program of tours and information on local activities will make the 6th DASC memorable for spouses, too.

On behalf of the conference committee, participants and the sponsoring societies — welcome to the 6th DASC.

EXECUTIVE COMMITTEE

GENERAL CHAIRMAN

Irving R. Reese
Boeing Commercial Airplane Co.

EXECUTIVE VICE CHAIRMAN

Johnnie L. Pearson
Westinghouse Electric Corp.

TECHNICAL PROGRAM CHAIRMAN

Randal K. Moore
General Dynamics, Fort Worth Div.

DEPUTY EXECUTIVE VICE CHAIRMAN

John G. Gregory
Westinghouse Electric Corp.

FINANCE CHAIRMAN

Frank W. Smead
ITT Avionics

PUBLICITY CHAIRMAN

Frank C. White
Aviation Consultant

DEPUTY PUBLICITY CHAIRMEN

Wendie Chapman
FAA

Arthur D. McComas
Allied Bendix Aerospace

SECRETARY

William L. Hyland
FAA

PUBLICATIONS CHAIRMAN

Jean M. Eason
General Dynamics, Fort Worth Div.

EXHIBITS CHAIRMAN

Harold H. Fink
ARINC AEEC

ARRANGEMENTS CHAIRMAN

Ralph D. Ormsby
Allied Bendix Aerospace

SPECIAL EVENTS CHAIRMAN

Richard D. Porter
Westinghouse Electric Corp.

DIGITAL AVIONICS SYSTEMS CONFERENCES

THE HERITAGE

	Technical Program		Registration		Exhibits		Tutorials	
	Papers	Sessions	Paid	Exhibits & Other	Firms	Booths	Courses	Registratio
AIAA 1ST DIGITAL AVIONICS SYSTEMS CONFERENCE APRIL 2-4, 1975 BOSTON General Chairman: C. Eric Ellingson, *Mitre*	53	9	175	—	—	—	—	—
AIAA 2ND DIGITAL AVIONICS SYSTEMS CONFERENCE NOVEMBER 2-4, 1977 LOS ANGELES General Chairman: William M. Pulford, *Bendix*	78	15	228	—	—	—	—	—
IEEE/AIAA 3RD DIGITAL AVIONICS SYSTEMS CONFERENCE NOVEMBER 6-8, 1979 FT. WORTH General Chairman: Daniel S. Goldin, *TRW*	88	16	396	210	14	29	—	—
AIAA/IEEE 4TH DIGITAL AVIONICS SYSTEMS CONFERENCE NOVEMBER 16-19, 1981 St. LOUIS General Chairman: John C. Ruth, *General Dynamics*	117	23	494	650	37	56	6	150
IEEE/AIAA 5TH DIGITAL AVIONICS SYSTEMS CONFERENCE OCTOBER 31-NOVEMBER 3, 1983 SEATTLE General Chairman: Cary R. Spitzer, *NASA*	135	25	655	825	33	61	6	285

THE HISTORY

The Conference began in the fall of 1973 when discussions among key members of the AIAA Technical Committee in Communications surfaced a need for a conference to address emerging digital avionics technologies.

A proposal was generated and sent to the AIAA headquarters staff to try an experiment by having a conference called the Digital Avionics Systems Conference, to be held in the Spring of 1975 in Boston, Massachusetts.

There was a very strong interest by personnel from Mitre, Draper Labs, Lincoln Labs and Bendix to pull together the elements of this first conference. The conference that year was very modest in the number of presentations, speakers, and attendees. The proceedings were a very meager collection of papers and the motivation of the conference was to provide for very good presentations and open "birds of a feather" discussion periods. The concensus of the people attending was that this conference should become a permanent structure within the AIAA. The AIAA Technical Activities Committee concurred and decided that in 1977 a second conference should be held.

An innovative feature of the 1977 conference, which was then called the Second Digital Avionics Systems Conference, was the addition of an exhibits program. There were four exhibits displayed in the hallways of the conference hotel. They were small but demonstrated the value of an exhibits program to complementing a very good technical program.

Another landmark decision was made to include the IEEE as a silent junior partner in the 1977 conference. This partnership proved to be so successful that following the second conference, a representative of the IEEE (AESS) met with the AIAA Technical Committee on Communications and the conference organizers to discuss the role of the IEEE in future conferences. The final result of these meetings was a signed memorandum of understanding between the IEEE (AESS) and the AIAA Technical Committee on Communications to jointly sponsor the Digital Avionics Systems Conference in the future. Conferences would continue to be every two years, however, the overall conference responsibility, administrative, and financial, would alternate between the two societies. The Technical Program would always be the responsibility of the Society which was not responsible for the overall conference and, to help ensure continuity of the conference function, the technical program chairman of one conference would become the general conference chairman next conference.

The Third Digital Avionics Systems Conference occurred in the fall of 1979 in Fort Worth, Texas, and was the first of the IEEE run conferences with AIAA pulling together the technical program. This was a very interesting experiment and brought a new flavor and mode of operation to the conference.

Another novel feature of the Third DASC was a full commitment to an extensive large-scale exhibits program highlighting larger contractor exhibits. There were 15 exceptional exhibits, which provided an excellent balance to the technical program. This success put in motion the game plan for including the exhibits program as an integral part of the conference in the future. The exhibitors themselves, although expressing concern at first that this was a loss leader investment, were exceptionally delighted with the quantity and quality of people attending their exhibits, and 14 of the 15 expressed an immediate interest to be included in the next conference. Over 600 people attended the exhibits and shows, and the number of technical sessions supported a full three-day program with an evening panel session.

The Fourth Digital Avionics Systems Conference was in 1981 in St. Louis, Missouri. McDonnell Douglas acted as a local industrial sponsor. History had already taught the conference committee some very valuable lessons concerning the industrial support for conferences. In addition to willing and energetic volunteers, the presence of a large aerospace company to provide resources to support the conference was invaluable. General Dynamics provided this influence in Fort Worth, and Boeing had already agreed to support the conference in 1983 in Seattle. The conference had made its breakthrough. Over 1,000 people attended the excellent exhibits provided by over 30 companies. There were almost 500 paid registered attendees at the conference. As usual, another innovative feature was attempted. Six 3-hour tutorials were held on the day before the actual start of the conference. The tutorials presented information on relevant digital avionics topics and proved to be very successful.

The Digital Avionics Systems Conference was now in full stride. The fifth conference occurred in Seattle and was highly successful. Over 600 people attended the five parallel technical sessions and more than 60 exhibit booths were filled. The digital avionics community was waiting for this meeting to occur and anticipated it well. A decision was made at this time to hold the 6th DASC one year later in Baltimore. This accelerated schedule took the DASC out of sync with the closely related computers in Aerospace Conference. Westinghouse, FAA, and Bendix follow Boeing as the industrial aerospace sponsors and everything points to an unbelievably successful 1984 Sixth Digital Avionics Systems Conference.

MESSAGE FROM THE TECHNICAL PROGRAM CHAIRMAN

Randal K. Moore
General Dynamics, Fort Worth Div.

Welcome to the 6th Digital Avionics Systems Conference. Please permit me a few paragraphs to outline for you this year's technical program.

The tutorials will repeat some past favorites, sometimes with new instructors and different material. In addition, there will be expanded coverage of Ada™, as well as new topics in voice systems and artificial intelligence. Sylvia Blair has done an excellent job of obtaining tutorials that will be both interesting and informative.

The plenary session on Tuesday morning will take a forward look at digital avionics from four different perspectives. The plenary session theme "Digital Avionics Requirements for the 1990's" would be just as appropriate for the entire conference. The aircraft of the next decade will be built upon the lessons we learn today and the technologies now emerging.

As in the past, we have attempted to strike a balance between defense, commercial, and general aviation interests. This is reflected in the technical sessions, the plenary session, and particularly in the panel sessions. Ken Chow and Stew Baily have organized two superb panel discussions in order to better address different areas of current concern — VHSIC insertion for the defense community and the National Airspace Systems Plan for commercial and general aviation.

I am confident that the twenty-five different technical sessions will provide many papers to capture your interest. The session organizers, authors, and speakers have done a tremendous job. The papers cover technologies and end-item applications, components and systems, software and hardware, fixed-wing airplanes and rotorcraft. Yet, there is method to the diversity, and your selective attendance at the technical sessions will let you choose a view of digital avionic systems as wide or as narrow as you might wish.

I am proud to have been associated with the many people who contributed their time and talents to the 6th DASC technical program. I think you will agree that their contributions are reflected by the excellence of the result. Please join me as I say "Thanks" for a job well done.

Randal K. Moore
Technical Program Chairman
6th DASC

TECHNICAL PROGRAM COMMITTEE

TECHNICAL PROGRAM CHAIRMAN

Randal K. Moore
General Dynamics, Fort Worth Div.

DEPUTY TECHNICAL PROGRAM CHAIRMEN

AIAA	**K.K. Chow** Lockheed R&D Div.
IEEE	**Stewart Baily** ARINC Research Corp.
TUTORIALS	**Sylvia H. Blair** General Dynamics, Fort Worth Div.

TUTORIALS PROGRAM

The 6th DASC tutorials provide relevant and informative sessions on topics of concern to engineers and managers. These six sessions cover many of the latest developments in the avionics industry. The material is introductory in nature to provide a founda tion from which the student can evolve toward his ow specific applications.

INTRODUCTION TO DIGITAL AVIONICS

Richard A. Maher
VERAC, Inc.

Mr. Maher is currently Assistant Manager of the Avionics Group at VERAC Inc. in San Diego CA. He has over 25 years experience in development, evaluation, and the support of missile and avionic systems employing extensive use of advanced digital technologies. For the past 10 years at TRW, he was responsible for the development of advanced avionics concepts for electronic warfare and integrated avionics. Recent avionics program experience includes DAIS/-Pave Pillar, EW Area Reprogramming Capability (ARC), ICNIA, INEWS, and several VHSIC technology insertion applications.

Dr. John G. Weber
VERAC, Inc.

Dr. Weber is manager of the Advanced Avionics Systems Department at VERAC Inc. San Diego CA. He has over 20 years experience in the design, development, and testing of digital systems. His design and development experience has been principally associated with the field of digital avionics. He was the chief designer and program manager for the Digital Avionics Information System (DAIS) Program where he made major contributions in developing a number of current avionics standards. His testing experience has involved data reduction and analysis for the Minuteman ICBM program and the development of the integration facility for avionics support testing (IFAST) for the Air Force Flight Test Center.

Course Description -

This course provides an overview and orientation to digital avionics systems architecture, hardware/software elements and support systems including: technology application areas; software discipline; interface standardization efforts; and developments in real time simulation support systems. The course will begin with a historical summary of digital avionics; describe on-going programs to demonstrate systems architecture and interface standards; and discuss future trends in avionics technology including VHSIC processing, software, voice interactive systems, artificial intelligence, and reprogrammable support systems.

VOICE INTERACTIVE SYSTEMS APPLICATIONS AND IMPLEMENTATION

Dr. John C. Ruth
McDonnell Douglas Electronics (

Dr. John C. Ruth is presently the Vice President McDonnell Douglas Electronics Company, Marketing ar New Business Development where he is responsible f market and advanced program planning/development ar sales. Prior to moving to MDEC, he worked at Gener Dynamics, Fort Worth Division where he was responsible f advanced avionics new business development plans ar strategies. Dr. Ruth is retired from the Air Force where I served in numerous management positions dealing wi digital avionics including Director of the Digital Avioni Information Systems (DAIS) Program Office of the Air For Avionics Lab.

Carolyn A. Moore
VERAC, Inc.

Ms. Carolyn A. Moore is presently the task manager for t Avionics Displays and Control Simulator at Verac Inc. She responsible for the rapid prototyping system design simul tor for the F-16 C/D Program. Prior to coming to Verac, s was the senior design engineer for Phase I and Phase II AF F-16 Voice Command Program at General Dynamics, F Worth Division. She also served on the F-16 program as t Up-Front Controls Element Manager.

Course Description -

The concept of the "pilot as a manager" evolves as avi nic systems become more and more sophisticated. The pi will not only need to monitor and direct the various fun tions, but will need to access data or ask a subsystem relational data. This tutorial addresses three distinct areas voice interactive systems, starting with the reasons for t use of interactive voice recognition and synthesis as a v ble tool in current and advanced aircraft. Next, various typ of voice applications will be discussed, including a br history of the AFTI/F-16 Phase I flight test effort anc discussion on possible interactive voice applications advanced aircraft. Finally, a detailed systems approach how interactive voice systems can be implemented in cu rent and future aircraft systems will be described.

INTRODUCTION TO ADA™

Richard E. Bolz
Modern Programming Languages

Mr. Richard E. Bolz was an Associate Professor of Computer Science for the Air Force Academy in Colorado Springs for ten years. He was the code developer of the model course for the Department of Defense for Software Engineering with Ada™. He has been teaching Ada™ for the last four years and is a member of AdaTech and Ada™/Jovial User's Group. Mr. Bolz has presented this tutorial on numerous occasions including for Adatech and the Aerospace Engineering Conference and Show with excellent results.

Course Description -

This course introduces Ada™, the Department of Defense's new programming language. The purpose of the course is not to teach the language in detail, but rather to provide a flavor for the power and form of the language in the perspetive of modern software methodologies. The session begins with a brief history of the language, followed by detailed examples of packages, exception handlers, generic units, representation specification, and tasking.

DIGITAL SIGNAL PROCESSING

Dr. Leonard Chin
Naval Air Development Center

Dr. Chin has been with the Navy Department since 1967, working on a wide range of systems research and development projects including estimation, control, and digital signal processing of self-contained navigation systems for submarine, surface ship, and aircraft. Project assignments have included work on Global Positioning Systems (GPS) and Joint Tactical Information Distribution System (JTIDS).

Course Description -

The main objective of this tutorial is to present an overview of digital signal processing (DSP) techniques and applications with emphasis on the understanding of fundamental concepts and practical utilizations. The building blocks from which DSP techniques were developed will be reviewed first. This is followed by a comprehensive survey of familiar DSP techniques. Next an overview of DSP applications will be presented, encompassing topics such as filtering, spectral analysis, detection, signal reconstruction, image processing, etc. Finally for completeness, a brief report on future trends in DSP techniques and applications will be made.

ADA™ FOR PROJECT MANAGERS

Anthony B. Gargaro
Computer Sciences Corp.

Mr. Anthony B. Gargaro is a lead scientist with Computer Sciences Corporation (CSC) Defense Systems Division and has been a principal contributor to the Ada™ program since 1978. Mr. Gargaro has participated in activities in both the implementation and use of the Ada™ language for some years. Currently, he is the vice chairperson of SIGAda and a member of the KAPSE interface team from industry and academia. He has presented this tutorial on many occasions, including to Adatech national conferences.

Course Description -

This course will address what a project manager, who is implementing a program using Ada™, needs to know about the language and how to get the most out of Ada™. Emphasis will be placed on software productivity and developing quality programs, including presentation, abstraction, encapsulation, synthesis, instantiation, synchronization, and representation. Ada™ is such a powerful programming language that without an adequate understanding, a project manager could find the program managing the manager. With understanding, Ada™ can simplify the complexity of managing the program and reduce the cost of implementing the application.

OVERVIEW OF ARTIFICIAL INTELLIGENCE

Dr. William B. Gavarter
Research Scientist

Dr. Gavarter has spent several years developing an in-depth overview of artificial intelligence (AI) and robotics. He has just published a book entitled "Intelligent Machines", Prentiss Hall, covering this material. He was formerly the manager of Automation Research at NASA Headquarters and is now doing research and development on expert systems for NASA Ames Research Center. Dr. Gavarter is a past chairman of the Washington D.C. Chapter of IEEE, Systems, Man, and Cybernetics Society. He is published extensively on AI and robotics.

Course Description -

Artificial Intelligence is an emerging technology that has recently attracted considerable publicity. AI is a discipline devoted to developing and applying conceptual approaches to intelligent behavior, or simply put, it's an approach to developing smart computers. Many AI groups and organizations have been formed and applications are under development. This tutorial will provide an overview of Artificial Intelligence which will cover the foundation of AI, the techniques utilized, the applications, the participants, and finally, the state of the art and future trends.

* Ada is a registered trademark of the U.S. Government, Ada Joint Program office

PLENARY SESSION

"DIGITAL AVIONICS REQUIREMENTS FOR THE 1990'S ..."

"... TECHNOLOGY INSERTION ROADMAP"

Bernard A. Zempolich
Naval Air Systems Command
Headquarters

Since joining the Naval Air Systems Command in 1963, Mr. Zempolich has held positions of increasing responsibility in the areas of computer systems and related software. He is currently acting director of the avionics division at NavAir and technical manager for advanced developments in computer technology, hardware, software, integrated avionic systems and artificial intelligence.

Mr. Zempolich has a bachelors degree in electrical engineering from Catholic University of America. He graduated with distinction from the Industrial College of the Armed Forces and currently teaches computer science at the University of Maryland.

Todays advances toward smarter, faster, more modular avionics are pushing the system integrator role into the spotlight. Mr. Zempolich will discuss the integration of avionics with consideration of technology, information fusion, and advanced system architecture impacts. The talk will address key issues such as fault tolerance and reconfiguration, integration of subsystems from an electronic/mechanical viewpoint, and the utilization of VLSI components.

"... TOTAL WEAPON SYSTEM VIEW"

Phillip O. Brown
McDonnell-Douglas Corp.

Mr. Brown is Chief Program Engineer, Electronic Systems. He holds a BSEE, MSEE, and Professional Degree in Electrical Engineering from the University of Missouri at Rolla. As ASA Project Manager, Mr. Brown also has cognizance of VHSIC insertion (Pave Sprinter) and related IRAD effort. His previous areas of responsibility have been in the avionics systems associated with advanced USAF and USN aircraft; and with electronic attack and radiating sybsystem technologies for advanced concepts. Mr. Brown had direct responsibility for the F-15 avionics system performance analyses, as well as the development of equations for the integrated control of sensors, displays, and weapons in the many modes required for weapon system effectiveness. Mr. Brown is Chairman, Avionics Section of the Air Armament Division of the American Defense Preparedness Association, a member of the IEEE, Tau Beta Pi, Eta Kappa Nu, and the Academy of Electrical Engineers at University of Missouri at Rolla.

An Avionics System definition derived from Weapon System Mission Requirements and the subsequent development of the Weapon System is of vital interest to the aircraft prime inasmuchas the total Weapon System Performance Responsibility resides with the prime. This interest is brought about by the strong effect the Avionics System has on the performance, availability, combat effectiveness, and life cycle cost associated with the Weapon System. Key to countering the 1990's threat while remaining compatible with the projected economics and human resources available will be the on-board digital Avionics Systems. This presentation addresses the significance of the 1990 digital avionics in the areas of: 1) Development - hardware/software complexity, testability; 2) Production - new technology integration, standardization; 3) Supportability - on-off aircraft maintenance, reduced spares types; and 4) Affordability - all of the previous areas.

"... THE FUTURE OF DEFENSE AND SPACE SYSTEMS"

R. Noel Longuemare
Westinghouse Electric Corp.

Since joining Westinghouse in 1952, Mr. Longuemare has held positions in design engineering, line and project management, and has played a leading role in the development of modern radar and avionics systems for airborne and land mobile applications.

He was heavily involved in managing the introduction of digital computers and digital signal processors into these systems, and in formulating the systems architecture and philosophy for the Westinghouse Modular Radar Series now successfully implemented on several frontline Westinghouse programs.

His current management responsibilities include the development of advanced sensors and avionic systems, as well as key technology programs such as VHSIC and advanced signal processing. He is also responsible for managing the engineering department at the Westinghouse Defense complex near BWI Airport. He holds 8 patents and 17 patent disclosures, and is active in numerous technical and industrial societies in the avionics and military electronics field.

Mr. Longuemare holds a BSEE from the University of Texas at El Paso and an MSE from Johns Hopkins University. He has also completed the Stanford University executive program.

As we move into next generation Avionic Systems for the Department of Defense and other Government Agencies we see requirements for unattended systems, faster system reaction times, lower life cycle costs, shorter concept-to-deployment times, and fault reconfigurable systems for greater operational availability. These requirements demand new architectural approaches beyond those of todays digital Avionics Systems. New information flow concepts for interconnecting computing and processing subsystems, sensors, control and displays must be developed. Studies underway indicate the need for several levels of bussing, some with bandwidths several orders of magnitude higher than todays bussing speeds. To lower life cycle costs, standard modules compatible with generic system functions and partitioned to permit built in fault tolerance through rapid reconfigurable functions, will evolve. A major thrust will be the development of survivable systems emphasizing sensor fusion and integrated functions using a common processing approach.

Eugene C. Machacek
Rockwell International

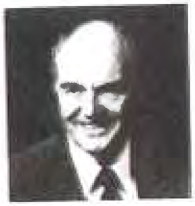
1ST DIGITAL AVIONICS SYSTEMS AWARI

CITATION

Spradlin, Sutcliffe, Peak and McDonald were the senior managers in the 767 Flight Management Systems Program organization. They were responsible for leading the combined project Engineering and Staff Task Force that directed the specification, design, development, test and certification of the new 757/767 digital avionics subsystems.

Richard E. Spradlin
Chief Design Engineer-Avionics/
Everett Division

Dick joined Boeing in 1954. He has held staff positions on the SST, 707, 727, 737. He is currently Senior Staff manager responsible for the design, development and certification of Flight Management Systems for the 757 and 767 airplane.



Peter L. Sutcliffe
Chief Design Engineer-FMS/
Avionics

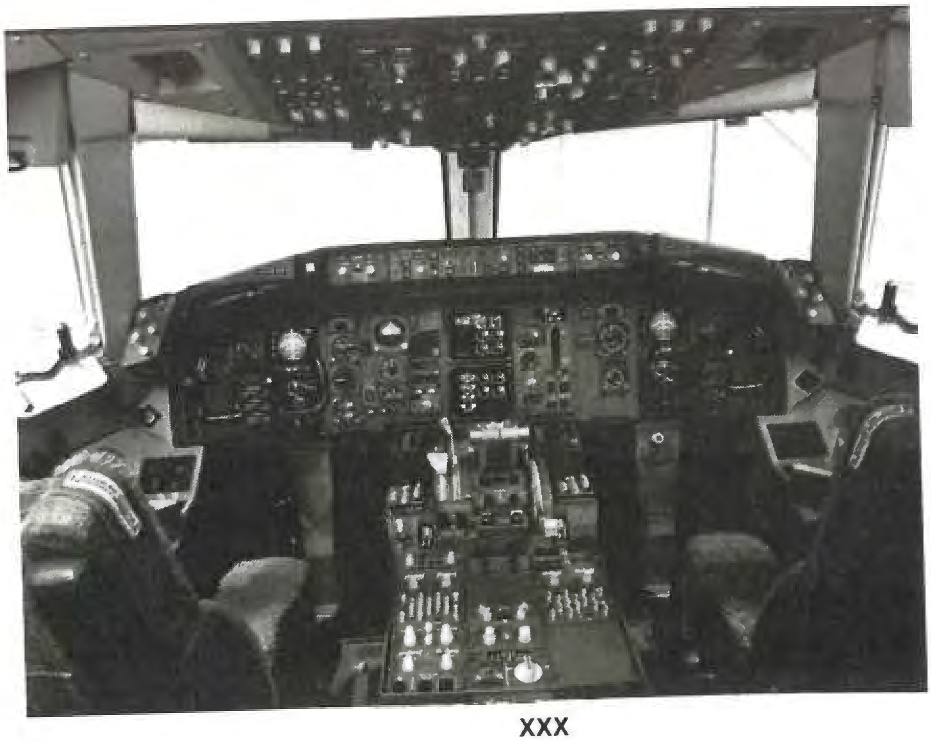
Peter joined Boeing in 1974 as Manager of Advanced design. In 1980, he became Deputy Manager-Systems Integration of 757/767 Flight Management Systems and later became Program Manager-757/767 Flight Management Systems.

Richard A. Peal
Chief Design Engineer - Avionic
Renton

During his 27 years of service, Dick has worked on the following programs: B52, WS-324A, C5A, SST, 707/727/737, 747, 757 and 767. He was Senoir Project engineer on the FMS Systems, E/E Systems, and Flight Deck.

Robert E. McDonald
Director-Engineering Researcl

During his 25 years at Boeing, Bob has worked o the 747 Program as Chief of Test and Director Techni cal Integrity-New Airplane Program. In 1979 Bob wa assigned as Chief Engineer responsible for develop ment of a common Flight Management System for th 757 and 767.



757/767 Flight Dec

PANEL DISCUSSION

NATIONAL AIRSPACE SYSTEMS PLAN —

ITS IMPACT ON DIGITAL AVIONICS SYSTEMS REQUIREMENTS

The avionics of the 1990's will become more integrated in terms of on-board architectures and with respect to the air traffic management environment. What impact will the initiatives undertaken by the FAA in the National Airspace Systems Plan have upon avionics system requirements? What will be the roles of the government, industry, operators, and the international aviation organizations in establishing these requirements? To address these issues we have assembled a panel drawn from diverse interested organizations in the aviation community:

MODERATOR

Stewart Baily
ARINC Research Corp.

Stu Baily is Manager, Advanced Systems Program, ARINC Research Corporation. He directs the company's advanced programs in aviation, command and control, and electronic warfare systems. Included in these programs are consulting efforts on the microwave landing system, modular avionics standardization, and variety of telecommunications, data link and voice radio system architectural investigations. Mr. Baily is a Senior Member, IEEE and a Governor of the Aerospace and Electronic Systems Society (AESS).

B.R. Climie
Aeronautical Radio, Inc.
(ARINC)

Rick Climie has been active in requirements definition and specification development for commercial avionics for over 25 years. He is a former chairman of the Airlines' Electrical Engineering Committee (AEEC). He currently coordinates, on behalf of ARINC and its airline ownership, the airlines' technical interests in system definition and standardization in the national and international communities in many industry and quasi-governmental groups including ICAO and ITU. Rick is Past President of the Aerospace and Electronic Systems Society (IEEE).

Peter C. McHugh
Aircraft Owners and Pilots Assn.
(AOPA)

Pete McHugh is responsible for keeping AOPA abreast of the developments in communications and navigation systems. He is a representative to RTCA and the future aircraft navigations systems committee of ICAO. Pete is also responsible for AOPA inputs to NOS and FAA aeronautical charts and publications. He was previously with the Aeronautical Systems Office, which performs similar functions for DCS, Operations, U.S. Army.

Robert Dunn
Boeing Commercial Aircraft Co.

Bob Dunn is responsible for design aspects for all avionics projects within Boeing Commercial Aircraft Company. His organization is also responsible for technical interface with BCAC suppliers, as well as for customer requirements. Software standards and control also fall within Bob's purview. Prior to his current position, Bob was Chief Engineer — Technology for the Boeing 747 aircraft program.

Martin T. Pozesky
Federal Aviation Administration

Marty Pozesky is responsible for "cradle-to-grave" oversight of National Airspace System Plan projects. These include a variety of air traffic control improvements, communications system upgrades, ground-to-air surveillance radar programs, and the microwave landing systems program. Prior to his present position, Marty was Deputy Director, Systems Research and Development Services, ARD-2.

WEDNESDAY LUNCHEON

"to recognize outstanding achievement in technical management and/or implementation of digital avionics in space or aeronautical systems to include system analysis, design, development of application."

Featured at the Wednesday luncheon is the presentation of the first Digital Avionics System Award, sponsored by the AIAA Digital Avionics Technical Committee. Nineteen-eighty-four marks the introduction of the award which is to be presented every two years at the Digital Avionics Systems Conference. In addition to this recognition, the award recipient also receives a medal and certificate of honor.

Selection of the recipient from the group of many worthy nominations involves assessment of each candidate's achievement or contribution with particular emphasis given to the degree of success in its practical application.

This 1st Digital Avionics Systems Award is being presented to the Boeing Flight Management System Program organization for their outstanding achievement on the 757/767 avionic system. The Boeing organization is represented by Richard E. Spradlin, Peter L. Sutcliffe, Richard A. Peal and Robert E. McDonald.

The Flight Management System equipment represents a significant step forward in commercial avionics. The equipment is now in service with both the 757 and 767 fleets.

PANEL DISCUSSION

VHSIC INSERTION AND ITS IMPLICATIONS

VHSIC Insertion is an ongoing DOD program to promote the insertion of VHSIC technology into operational systems. Very soon, VHSIC technology will be required for new DOD programs and add-ons. The panelists will discuss the intentions, mechanics, and experience of the VHSIC Insertion program. Although reference may be made to published accounts of VHSIC in the open literature, VHSIC chips and brassboards will not be presented due to ITAR restrictions. Conference attendees having experience with VHSIC Insertion or wanting to participate in the Insertion program are encouraged to participate in the ensuing discussion.

MODERATOR

Dr. K.K. Chow
Lockheed R&D Div.

Dr. Ken Chow has been working in the fields of electronic and opto-electronic devices and subsystems for twenty-five years, the last eleven years in management positions. His interests include communications, signal processing, and advanced concepts. Currently, he is manager of the Advanced Electronics Laboratory, Lockheed R&D Division; among his duties are the management of custom VLSI design and VHSIC Insertion.

LTC Nicholas J. Babiak
OUSDRE

LTC Babiak has been involved with high speed computing technologies for the past twelve years. Currently he is the VHSIC Deputy Director for Technology Insertion, Office of the Secretary of Defense, Office for Research and Advanced Technology, and is responsible for selecting and funding VHSIC Insertion programs. Previously, LTC Babiak was the Principal Advisor to HQ USAF for mission-critical computers.

John G. Gregory
Westinghouse Electric Corp.

Mr. Gregory has been in the computing field since the ENIAC/EDVAC days. For the past twenty years, he has worked in military digital systems, computers, and software development. Presently, he is manager of Digital Programs at Westinghouse; among his duties is the management of VHSIC Insertion programs.

Richard A. Maher
VERAC, Inc.

Mr. Maher has twenty-five years of industrial experience in system development, test and evaluation and logistics support for DOD programs. His work at TRW during the last twenty years was in missile guidance and avionics, all requiring extensive use of advanced digital technology. For the past few years, he has been involved in VHSIC Insertion into EW and communication systems. He recently joined VERAC as assistant manager of the Avionics Group.

Don Staake
Johns Hopkins University

Mr. Staake has been active in the requirements definition, design, and development of radar systems, primarily for missile guidance, for over thirty years. He is on the Special Assignment staff of the Fleet Systems Department at JHU/APL, and is also a member of the staff of VHSIC Workshop lecturers. Currently he is involved with planning for the insertion of VHSIC into various Navy programs. Previously, Mr. Staake was the Operational Systems Development Branch Supervisor at JHU/APL, where he was responsible for the development of fleet radars for the surface Navy.

Charles Caposell
ODASN

Mr. Caposell has been in the high technology field for over fifteen years. Currently he has split responsibilities as both the Assistant to the Deputy for Electronic & Physical Sciences in the Office of the Secretary of the Navy, and as the Technology Manager of the Avionics Division at NAVAIR. He has been associated with the VHSIC program since its inception and is currently responsible for all aspects of VHSIC within the Navy directorate, especially technology insertion programs.

THURSDAY LUNCHEON SPEAKER

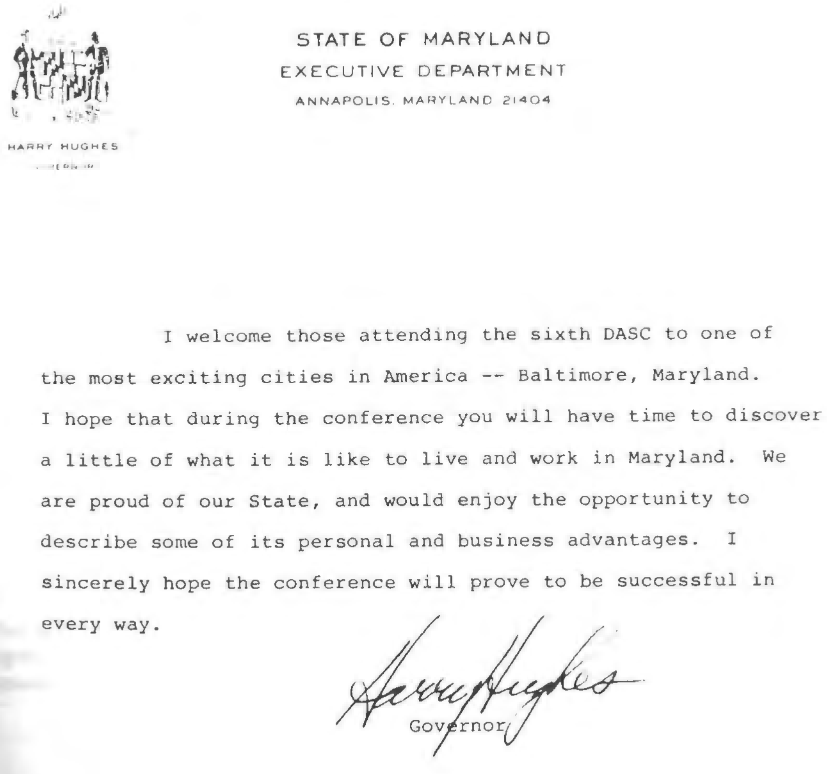
"The Price of Professionalism"

Jack Jackson
Leadership Development Institute Inc.

"Jack" Jackson is on the staff of Leadership Development Institute, Inc., in Fort Worth, Texas, and is chairman of the board of Jack Jackson and Associates. Prior to his association with LDI, Jack was an instructor with American Airlines for 23 years. During the last 13 years of that time he was also a goodwill ambassador for the company and traveled and spoke nationwide. He began his career as a member of the United States Air Force and upon completion of his duty, moved into Civil Service. He later entered private industry, spending several years with the Boeing Company prior to joining American Airlines' Flight Academy. Jack has become an institution with the DASC. His talks are always relevent, thought provoking and highly entertaining.

MESSAGE FROM THE GOVERNOR OF MARYLAND

Harry Hughes
Governor



STATE OF MARYLAND
EXECUTIVE DEPARTMENT
ANNAPOLIS, MARYLAND 21404

HARRY HUGHES
GOVERNOR

I welcome those attending the sixth DASC to one of the most exciting cities in America -- Baltimore, Maryland. I hope that during the conference you will have time to discover a little of what it is like to live and work in Maryland. We are proud of our State, and would enjoy the opportunity to describe some of its personal and business advantages. I sincerely hope the conference will prove to be successful in every way.

Harry Hughes
Governor

SESSION 1 SYSTEMS AND SOFTWARE - DEVELOPMENT METHODS

Chairmen:

Gordon Henley
Intermetrics, Inc.

Paul E. Gartz
Boeing Commercial Airplane Co.

This session examines time-proven and emerging system and software engineering techniques having application to military, commercial, and governmental sectors. Included are practical assessments of results and lessons learned.

DEVELOPING SAFE SOFTWARE FOR CRITICAL AIRBORNE APPLICATIONS

84-2598

William B. Noble
Hughes Aircraft Co.
Ground Systems Group
Fullerton, California

Abstract

Research and development work over the past few years has yielded several promising approaches to integrating the traditional hardware design and analysis techniques into the software development process so that the safety and reliability of complex software driven processes could be assured. This paper addresses a number of these areas, with emphasis on those techniques which can be used to ensure the safety of software driven embedded systems. This paper is concerned with design and analysis techniques, and does not address the area of control law validation or statical simulation methods which can be used to evaluate the fault-free performance of a control system.

Introduction

Automatic systems are being used for increasingly complex and critical applications. In the not too distant past, the scope of control, or authority, of control systems could be so limited that faults were tolerable and the system was fail soft. This started to change with the development of full time flight control systems which included critical phases such as climb-out and landings. Initially, these systems were designed to be fail passive, allowing the pilot to assume control in the event of a malfunction. However, the application of computers for active load alleviation, stability enhancement and fly-by-wire, and the increasing requirement for automatic landing capability in low visibility situations, now places the system in a situation in which manual recovery is no longer a feasible alternative.

As these applications evolved, analytic techniques were developed and perfected to address the reliability and safety issues for such systems, while hardware advances allowed the construction of inherently more reliable equipment and the practical application of redundancy to ensure fault tolerance. However, the system design implementation, represented by the hardware and software, poses a common thread amongst redundant channels which can obviate the intended fault tolerance.

This paper is directed primarily at flight control applications, but the concepts presented will be applicable to other applications which are similar in size and repetition rate. In general, a flight control program consists of an infinite loop which, once entered, executes so long as power is applied to the computer. This loop must repeat at a specified interval (called the cycle time) to implement the control laws and logic. Typical cycle times range from 15 to 50 milliseconds, although values outside of this range are not uncommon. The program itself may range from a few thousand to thirty or forty thousand instructions, about evenly divided between mathematical operations and boolean logic, and is usually stored in non-alterable memory. The program is embedded within the computer system, and cannot be accessed or altered without special equipment. The user (pilot) can interact with the program through a control panel, but the operation of program itself is usually a total mystery to the pilot who is interested in controlling the airframe, not in operating software.

The Concept of Safety

Safety is an interesting term because it has two aspects: SAFETY, which is a philosophical concept that relates to the perceived risk that the system can impose upon the mission for which the system is intended, or the risk that can result to passengers, operators, or the general public from the operation of the system. Safety (with lower case letters) is the other aspect, and is what can be analyzed and quantitatively evaluated. A system is considered safe if it avoids unsafe states or conditions. Unsafe conditions, or hazards are specifically defined undesirable occurrences (such as a control surface hardover, or an engine overspeed). The hazards represent a mapping between SAFETY and safety which allows the precise analysis of a rather imprecise concept. However, the mapping, or hazard definition is a manual effort which is based on the skill of the analyst and prior experience with similar systems, and is not guaranteed to be exhaustive. Thus it is perfectly possible to show that a system is safe, eg., it meets all the specified safety criteria, and still have the system enter a state which, in the light of hindsight is clearly not SAFE.

Understanding the difference between safety and SAFETY is critical to understanding the value and limitations of the system safety discipline. Even though the specified hazards which are analyzed may not be exhaustive, it is clear that a system which can be shown to avoid known hazards is at least as safe as a system

*Member AIAA, IEEE

which has had no such analysis. Moreover, in the realm of flight control, the nature of the hazards are generally well known, and thus the likelyhood of a previously unknown hazard leading to a loss of SAFETY is quite unlikely.

It is also important to recognize that safety and correctness are related, but not overlapping concepts. A system can be in one of four states:

- safe and correct: this is the ideal operating state of the system.
- safe and incorrect: a car which is stopped with the brakes locked is safe, but not of much use if the purpose of the vehicle was transportation.
- unsafe and correct: a particularly nasty state that usually results from the system designer misunderstanding the environment or the safety requirements. A control law that does not complement a vehicle's aerodynamics can result in instability. This can easily be unsafe, yet the system is performing in the specified manner.
- unsafe and incorrect: a car in motion with no brakes, or a control system that fails hardover are both unsafe and incorrect.

Clearly, the object of the system designer is to create a system which remains in the first state as much as possible and never enters the second or fourth state.

Two other areas are related to safety: reliability and availability. Reliability is usually associated with failure rates or mean time between failure (MTBF) figures, and can best be thought of as the amount of time between incorrect system operations. With software, reliability can be thought of as the time between the discovery of errors. Software is considered reliable if it performs the specified functions when it is supposed to. Availability is the probability that the system will perform the intended function when it is supposed to. Mathematically, availability is MTBF/(MTBF+MDT) where MDT or mean down time is the time between the occurrence of a failure which incapacitates the system, and the restoration of the system to a functional state. It can also be thought of as "up-time" divided by total time. Fly by wire systems must provide high availabilities (or very low probability of loss of function) due to the safety implications of loss of control.

Hardware Approaches

The usual approach in hardware to achieving high availability and overall system reliability is to add redundancy, on the presumption that failures are independent, and thus the probability of multiple channels having simultaneous failures can be calculated based upon the product of the individual channel failure rates. The number of channels (the degree of redundancy) influences the survivability of the system, thus a system may be fail safe, or fail operational. Similarly, safety is generally achieved by adding independent hardware to monitor the relevant parameters and disconnect the affected channels or limit the commands to a suitable value.
The approaches described above all rely on the independence of events. Because of this, it is necessary to show, by analysis that independent channels are in fact independent. A "common mode" analysis performs this function by verifying that a single failure will not propagate into independent areas and thus cause multiple channels to fail simultaneously. This analysis looks for physical separation between allegedly independent channels. Where the channels interconnect, the interconnection is analyzed for suitable buffering or interlocks which can ensure that a failure will not propagate across the interconnection.

A sequence of failures can still lead to an unsafe condition if for example a monitor were to fail, and that failure were followed by a second failure which should have been detected by the monitor (a power supply monitor fails in the always good condition, followed by a failure of the power supply). Such sequences of failures are analyzed with fault trees, which begin with a specified hazard, and determine the ways the system can cause the hazard. Each of these ways is then decomposed until a quantifiable condition is reached (such as a component failure, or an equipment failure). The tree is then evaluated, and the result compared to the remotness requirement for the hazard. If the goal is not met, then the decomposition continues until it is, or until it becomes apparent that the system design is deficient and must be revised. Figure one shows the top levels of a fault tree for a typical fail-passive flight control system with two independent channels, and one servo to position the control surface.

The Software Problem

A classic concern among designers of software driven multi-channel systems is the generic error. A generic error is an error of the design which is not uncovered during testing,and which causes all of the redundant channels to generate the same incorrect command under some particular set of conditions. Although this is usually thought of as a software problem since it is common for redundant channels to share common software, it is really a problem of design. Redundant channels usually share a common design for the hardware components, the processor is usually the same, and the general structure is usually the same. Thus a generic error could occur not only through an error in that part of the design represented by software, but also through an error in the processor microcode, or in the layout of a particular integrated circuit. A traditional failure modes and effects analysis (FMEA) which examines the system level effect of failing each component in the system, was intended to address this area. FMEAs were not applicable for software because the failure modes could not be defined (eg. it can always come up with another thing it can do wrong). Similarly, with the advent of complex integrated circuits (for example the 80286 has over 300,000 transistors) the failure mode model of each pin stuck at one or zero no longer applies. Furthermore, MOS devices tend to fail in pattern sensitive ways rather than in the traditional "stuck at" way, [1] so the applicability of FMEAs to even the hardware portion of a processing system is now somewhat dubious.

There are some notable properties of the software portion of a design, however, which can assist the task of evaluating safety. First, of course, it does not fail due to wear or stress like hardware, rather "like a fine wine" it improves with age. Secondly, the software design is precisely represented by the source code (which is, conveniently, machine readable). This is in contrast to hardware which is partially represented by a schematic, partially by the operating parameters and tolerances of the components, and which is influenced by external factors such as temperature and humidity. The combination of an accurate representation of the design, and machine readability not only allow safety analysis techniques to be extended to software, but also hold forth the promise of automated performance of the analysis. However, given the potentially unbounded number of paths through the software, the software designer, like the hardware designer, must keep in mind the need to ultimately analyze the code, and thus create a design in which the number of paths which must be examined is limited (for example, by including a limit to prevent excursions beyond a safe value, or other reasonableness tests).

There are three ways to reduce the number of errors uncovered by the end user of a piece of software:

- Don't put the errors in in the first place
- Detect the errors through analysis or testing
- Provide a fault tolerant design

The first approach involves specification, design, and construction techniques which reduce the incidence of errors. Examples include the use of a strongly typed high order language (HOL), modular construction with single function modules, and structured design techniques. There is a vast body of literature dealing with this area, so it will not be further elaborated in this paper. It is worth noting, however, that the argument that real time software must be coded in assembly language for efficiency reasons is clearly obsolete now that several commercial (and military) programs have been completed without the use of any assembly coding.

The second approach includes walkthroughs, design metrics which indicate areas of undue complexity, software safety analysis, test techniques, and test coverage analysis. Walkthroughs are applicable to flight control, and are effective at uncovering a broad range of errors. Unfortunately, under the pressure of deadlines and budget, walkthroughs may be improperly attended, or may be scheduled to occur for lengthy periods (many weeks, when 2 hours at a time is a realistic maximum for effectiveness). Design metrics are intended to uncover areas in a design which are likely to contain errors based on the number and complexity of the interfaces, the structure or other design features. The current state of such metrics, however, is still rather primitive, with most metrics predicting errors no better than very simple measurements such as module length. In addition, many metrics are designed for transaction driven systems which deal with large amounts of data, and are poorly suited for evaluating control systems. Software safety analysis is an extension of the hardware safety analysis techniques which is discussed further in the next section. There is a large body of literature dealing with software testing ([2] or [4] are good introductions to this area). Additionally, standards such as RTCA-DO-178 emphasize exhaustive testing as the main approach to ensuring an absence of errors while providing little guidance as to how much testing is enough. Techniques such as error seeding and reliability models with confidence levels (such as MUSA) show promise in this area, but are too immature for application to safety critical systems yet.

Software fault tolerance falls into two general areas, one related to the fail-passive system design approach, and the other related to fail-operational concepts. The first involves the segregation and isolation of the critical from the noncritical so that a failure of a noncritical function or routine does not compromise critical functions. For example, it is common for control systems to contain a time critical foreground and a non time critical background. The foreground performs the real time control functions, while the background performs noncritical diagnostic and maintenance functions. Providing rigid isolation between background and foreground, supported by a very simple error handler, would allow foreground operations to continue despite the presence of a software error which fails background. Applying this concept to a control system requires only that the hardware preclude write operations to foreground memory locations while background is executing.

The second area of software fault tolerance basically involves providing redundant versions of the software for a critical routine so that an error in one version will not cause loss of the function. This concept is an extension of the hardware redundancy philosophy, and like the hardware approach requires that the independent versions be actually independent. This independence is not easy to show, since the versions at some level must share a common specification of the required function. Raising the level of the common specification requires that not only must multiple coding and testing efforts occur, but also multiple specifications be written. At this time it is a matter of opinion as to whether the additional cost of coding multiple versions of a routine might not be more fruitfully spent in more throughly testing and evaluating a single version of the program. Backup software is sensible if it is either trivial, or where there is a mathematical risk in the primary design. An example of the trivial case is providing a direct stick to actuator link on a fly by wire system to provide some degree of control if the main program which provides good flying qualities should fail. An example of a design with a mathematical risk might be an optimal control algorithm which is required to invert a matrix which cannot be proven to be nonsingular in all cases. Such a design might include a "satisfactory" but suboptimal control algorithm for use when the matrix is found to be singular.

Software Safety Analysis

There are two types of analysis specifically aimed at ensuring the safety of the software design. One, called software common mode analysis, ensures that hardware failures do not propagate across channel boundaries through software paths. This type of analysis is directed at multi-channel redundant systems in which the channels exchange data. A failure in the hardware could cause the data being exchanged to assume any arbitrary value (unless proven otherwise). The analyst identifies each exchanged data item which is writable by software (pure hardware exchanges are covered under the hardware analysis). Each item is then traced to the receiving channel or channels, and each module within that channel which accesses or uses the data item is identified. Each identified module is then examined to determine if any arbitrary value of that data item could cause the module to create an output which is incorrect. This type of analysis has been performed on safety critical software, and it is capable of locating errors not found by other means. Although not currently implemented, a certain degree of automated assistance in the searching and identification area is not difficult.

The second type of software safety analysis examines the source code for any paths which can lead to a specified output. This analysis is called software fault tree analysis, and is analogous to, and supportive of, the hardware fault tree analysis. The analyst begins at the hardware/ software boundary with the way or ways that software can cause the hazard at the top of the particular fault tree (see figure 1). Each of the modules which can reference the relevant output variables is identified and the module analyzed for the logic and data states which must be present for the condition to occur.

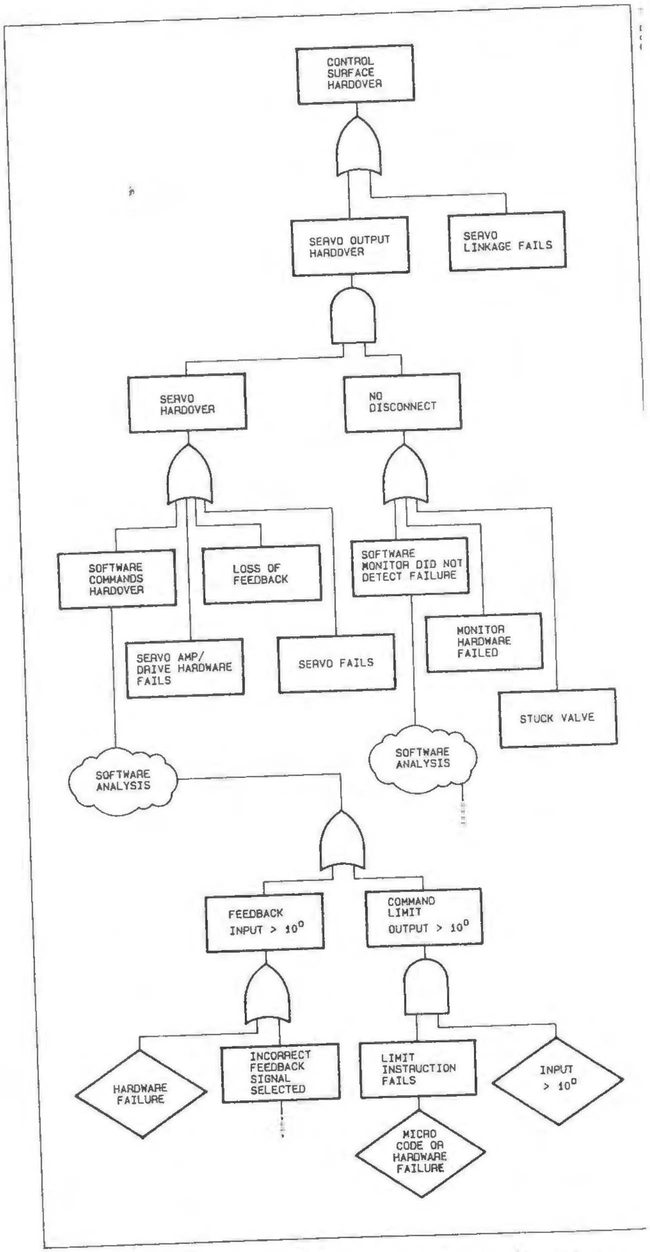


Figure 1: Sample Fault Tree Showing Hardware and Software Elements

This decomposition continues into successively deeper layers of the code until either all possible paths are shown to be impossible (through contradictions such as A and not A or 1=2), or the specific paths which will lead to the hazard are identified [3]. Although a good walkthrough performs a similar function, since all participants try to "break" the code, this analytical method is more rote, and requires a much lower skill level on the part of the analyst.

Summary

The concepts of safety and reliability are distinct and separable. Reliability is achieved if a system performs the intended functions when it is supposed to. Safety is achieved if a system does not perform certain specified erroneous actions (even in the presence of a failure). The two concepts are related, because the greater the reliability of a system, the less opportunity there will be for erroneous actions, however, safety encompasses errors of specification and conceptualization which can lead to a system which is operating in the manner it was designed, but which causes a unsafe condition through that operation (an example is a flight management system which conserves fuel successfully, but which retards the throttles on descent beyond the point at which efficient engine deicing occurs, thus leading to an in-flight shut down of one or all engines). This paper has suggested that safety be addressed as a separately identified item in the system level specification, and that compliance with that item be shown analytically. Two analytic techniques, fault trees, and common mode analysis, have been presented, and their application to software systems explained. While the application of these techniques will not guarantee an absolutely safe system, because the safety specification could be incomplete, or there could be anomalies in the hardware, these techniques can certainly improve the safety aspects of the software which drives the system.

References

[1] El-ziq, Y.M.- Classifying, Testing, and Eliminating VLSI MOS Failures, VLSI Design, Sept,1983 pgs 30-35

[2] Meyers, G.J.- The Art of Software Testing, 1979, Wiley Interscience, New York

[3] Leveson, N. and Harvey, P.- Software Fault Tree Analysis, Journal of Systems and Software, Volume 3, Number 2, June 1983

[4] Beizer, B. -Software Testing Techniques, 1983, Van Nostrand Reinhold Company, New York

84-2595

THE USE OF STRUCTURED METHODS IN THE DEVELOPMENT OF LARGE, SOFTWARE-BASED AVIONICS SYSTEMS

Derek J. Hatley*

Lear Siegler Inc. Instrument Division
Grand Rapids, Michigan

Abstract

Structured methods, notably those developed by Yourdon Inc. for representing the requirements and design of software-based systems, have gained wide acceptance and considerable success in business applications. These applications typically do not have critical control or timing requirements, and the methods take advantage of this fact by employing a data-triggered processor model which de-emphasizes these kinds of requirements. By contrast, large software-based avionics systems require multi-mode operation, direct interaction with a rapidly changing physical environment, and fast response times. The basic Yourdon methods have been successfully extended to represent such requirements, while the rigor and simplicity of the methods have been preserved. These extended methods were applied to the development of a commercial Flight Management Computer System and contributed to significant reductions in development time and to improvements in the early achievement of full system performance compared to those for previous, similar systems.

The Need for Structured Methods

As software based systems grew in size and complexity the problems in getting them to work became increasingly difficult. Overruns in cost and schedule by factors of two, three or more over original estimates have been common. Serious problems have arisen in service which were undetected during development. When changes to the software were required further problems arose, as features unrelated to the ones changed were inadvertently affected. Structured Programming has been in use for some time, and has greatly improved the quality of the implementation, but it became evident that better methods for representing the software design, and the system and software requirements were also needed. As a result, a number of methods were developed for incorporating a formal organization or structure into both the design and the requirements specifications of software-based systems.

Impressive statistics have been gathered on the increasing cost of fixing a problem as the project progresses. It is many times more expénsive to fix after the system is delivered, for example, than at the design stage. Accordingly, the thrust of the structured methods as a whole is to "front end load" the project — putting as much work as possible into getting the requirements established before starting the design, and as much as possible into the design before starting the code — and thus taking advantage of this tremendous cost leverage of finding problems early. The methods make this easier to do by providing a formalized way of representing requirements and design such that it is relatively obvious when something is incorrect or incomplete. This same formalism makes changes much easier later on, with less risk of disrupting the system operation.

Available methods were evaluated at the start of a major development for a commercial Flight Management Computer System (FMCS) with the hope of finding ways to improve the performance of this and other projects, both commercial and military. The Structured Analysis (SA) and Structured Design (SD) methods developed by Yourdon Inc. seemed to have the most promise, but no methods at that time (early 1982) were found to be fully adequate for large, real-time systems such as the FMCS. A decision was therefore made to adopt the Yourdon methods as a basis, but to extend them as needed to meet the needs of such systems.

The Yourdon Methods

All the methods investigated share the goals of demonstrable rigor and completeness and all achieve them to some degree, but with these goals only, they tend to be very similar to high-order programming languages — more suited to machine reading than to human reading. Some methods take account of the fact that specifications are, in fact, read by people and they therefore also include human understandability as a principal goal. The Yourdon methods fall into this latter category and achieve the additional goal in part by using a diagrammatic approach. This has the advantages that the brain is much more powerful at processing pictures than text ("a picture is worth a thousand words"): the mind works best when working with 7±2 items (1), so the number of objects in a diagram is normally kept within this range; we are better at improving on things than creating them, and the methods make it easy to make a quick first cut at an analysis, then concentrate on improving it.

The Yourdon methods have been widely and successfully applied to commercial business applications for some years and this existing experience was another factor making them attractive as a starting point for development of extended methods for real-time systems. This was especially true since the extended methods were to be used immediately on a large, critical project.

The basic Yourdon methods are well documented (2)(3)(4), but a brief summary of those features which are important in understanding the extensions to the methods is included here for reference.

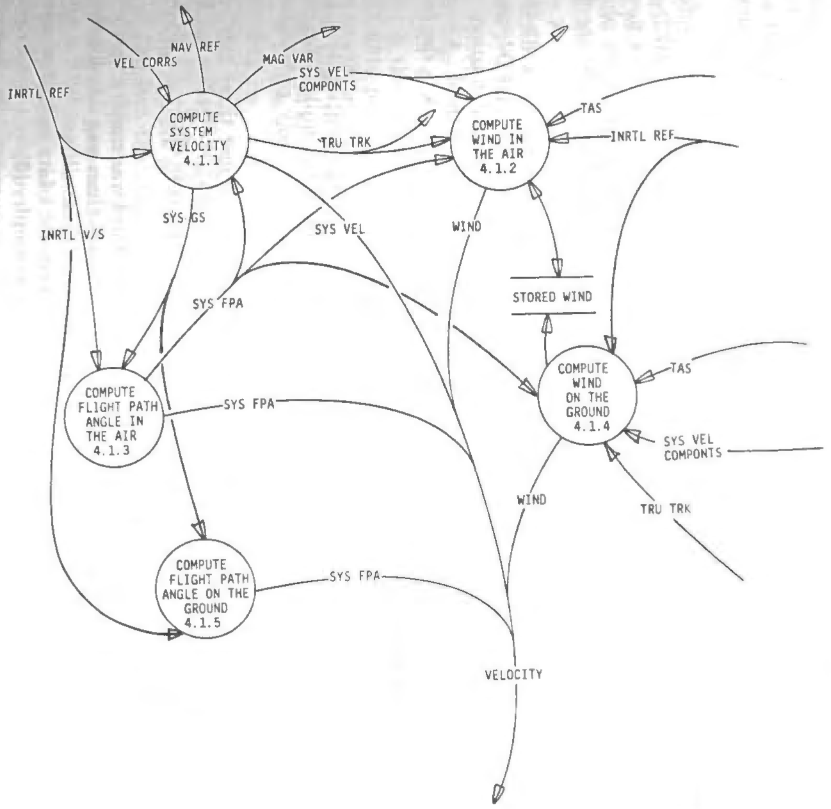
Structured Analysis

SA is a method for specifying the *requirements* of a system, and is notable for its simplicity. It consists of just three entities: *data flow diagrams* (DFDs), *process specs*, and a *requirements dictionary* (this terminology differs slightly from that of Yourdon for reasons discussed later). Of these, DFDs are the main tool, representing a diagrammatic model of the system's functional requirements.

Data Flow Diagrams in turn, consist of three components: *processes*, *data flows*, and *data stores*. Figure 1 is a typical DFD. The processes, represented by circles, each have a name which describes the action that process is to perform to produce its output data flows from its input data flows. (Processes are often referred to as "bubbles" and DFDs as "bubble diagrams" or "bubble charts"). Data flows, represented by named vectors, show the flows of data between processes. In general, data flows represent flows of information, signals, or materials, or groups of such items. Data flow names describe the items or groups of items in the flows. Data stores, shown as parallel line pairs with an associated name, represent data which is to be saved for some later processing. Their names represent the data stored in them.

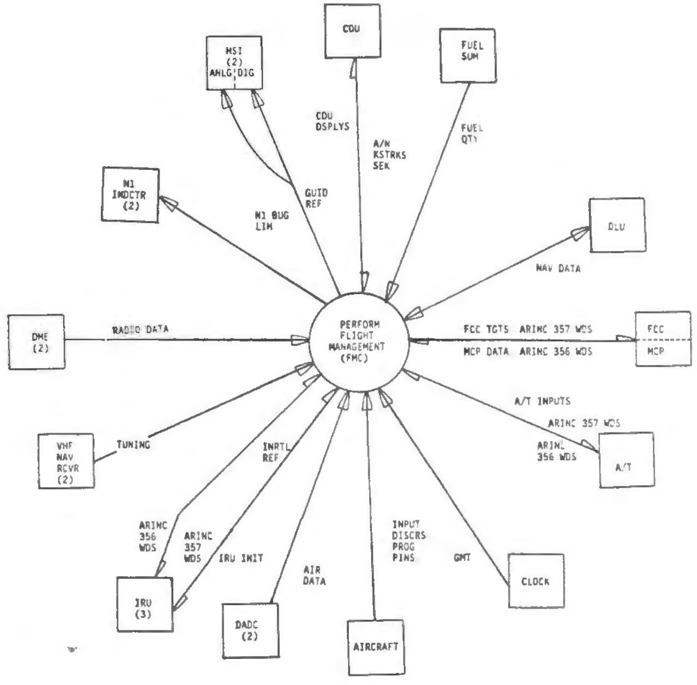
Context Diagram. A system is described by a set of DFDs, starting with a special one called a *context diagram* (Figure 2). This consists of a single process representing the whole system, all the data flows to and from the system, and the external entities with which the system is to communicate. These external entities are called *sources* and *sinks*, or just *terminators*, and are represented by rectangles. The name of the single process in the context diagram is a very general, abstract statement of the total task the system is to perform. Similarly, the data flow names at this level usually represent large generic groups of items.

*Member, IEEE



Process No. 4.1 Compute Velocity

Figure 1 Data Flow Diagram



FMCS Data Context Diagram

Figure 2 Context Diagram

Leveling. The single process in the context diagram is decomposed in the *level 1* DFD into sub-processes, its data flows appearing as inputs and outputs to the level 1 diagram. Each process in the level 1 DFD is broken down into sub-processes in a *level 2* DFD, and so on. As the processes are decomposed so the data flows are progressively divided down from the large groups at the context level until finally they are separated into individual items. This progressive decomposition of the processes and data flows ends whenever a process can be completely and unambiguously described in a few lines of text, equations, or a simple diagram or chart, etc. This description then becomes the process spec for that process, and the process is called a *primitive*. The whole decomposition procedure is called *leveling*. The DFD which decomposes a given process is called the *child* diagram of that *parent* process, and the analogy is extended to *grandparents, grandchildren, ancestors, descendants*, and so on. Since a child diagram is simply a more detailed representation of its parent process, the two are given exactly the same name. Figure 3 shows four levels of DFDs in one diagram, and makes it clear that the set of diagrams *could* be shown as a single network of all the primitives. The higher level diagrams are abstractions of that network, merely collecting the primitive processes together into ever larger groups.

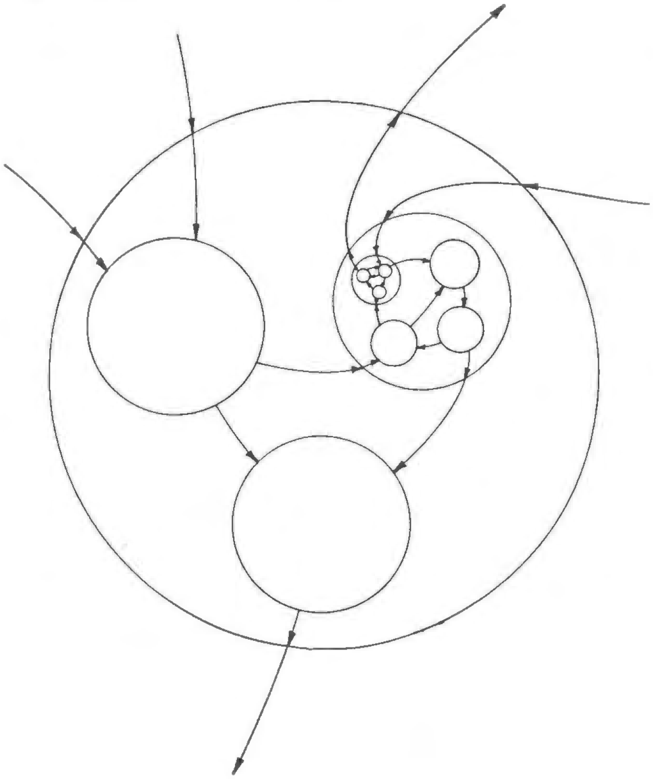


Figure 3 Four Levels of Data Flow Diagrams.

Balancing. It is understood that data flows entering and leaving a DFD are in fact the inputs and outputs of the parent process. Therefore, unlike traditional signal flow or block diagrams, off-page connectors are not needed. Furthermore, verification of this one-to-one correspondence between the inputs and outputs of parents and their children is one of the primary consistency checks of the method and is known as *balancing*.

Numbering System. The method incorporates a numbering system which goes hand-in-hand with the leveling procedure. Just as a parent/child pair are given the same name, they are also given the same number. The single process in the context diagram is Process 0. Processes in the level diagram are numbered with single digit numbers (there as normally no more than 9 processes in a DFD). Processes in level 2 diagrams are numbered with two digit numbers, of which the first is the number of the parent process. In general, processes in a level n diagram are numbered with n digit numbers of which the first $(n - 1)$ are the same as the number of the parent process. This numbering system, combined with an orderly arrangement of the diagrams in the specification, make the method self-indexing. By simply looking at the process numbers in a DFD one knows the level of that DFD and where to find its parent and children.

The DFD Model. The set of DFDs representing the system requirements is an idealized, machine-like model. It is implied that the processes are "data triggered", that is, whenever there is sufficient data on the input data flows for the process to perform its task it will automatically do so. It is also implied that the task is performed instantaneously. Thus, whenever there is sufficient information on the *system* input terminals (context diagram inputs) the corresponding system outputs will appear immediately. This idealized representation allows the requirements statement to be implementation independent. Given the SA specification, the required system response times, and the characteristics of the particular hardware to be used, the designer has all the information needed to proceed, and changes in the two latter items should not affect the former.

DFDs, then, represent the data processing requirements of the system. The philosophy when preparing them is to suppress statements about controlling the system, such as when or under what conditions a process is to operate. In the kinds of system for which the method was developed, these considerations tend to be somewhat implementation dependent, and also fairly straightforward—a process might be activated, for example, whenever a customer comes to a teller's window, or whenever the weekly payroll is due.

Process Specs are the descriptions of how the outputs of the primitive processes in the DFDs are to be generated from their inputs. They are typically just a few lines long, and are written in *structured English* together with equations, diagrams, charts etc. as appropriate.

Requirements Dictionary. This is simply an alphabetical list of all the data flow names in the DFDs, each with a definition in terms of its component data flows, or, in the case of an individual (primitive) signal, its physical properties. A special notation is used to describe the data structure of the flows.

Verification of the presence of a dictionary definition for every data flow name in the DFDs, and for every component data flow within each definition, is another primary consistency check of the method. An important flaw often discovered as part of this check is the presence of "circular" definitions.

Structured Design

SD is a method for representing the *design* of a software system independent of its implementation. For example, it should be possible to use any programming language with a given SD.

Like SA, SD consists of just three entities: *structure charts, module specs*, and a *design dictionary*. Again, the diagrammatic tool—the structure chart—is the main tool of the method.

The basic SD method was much closer to meeting the needs of real-time systems than was SA so only minor adaptations were made to it which will not be discussed further here. A very brief description of its main features follows. For more detail see, for example, (3).

Structure Charts (Figure 4) consist of *modules, calling vectors, data couples*, and *control couples*. The method involves

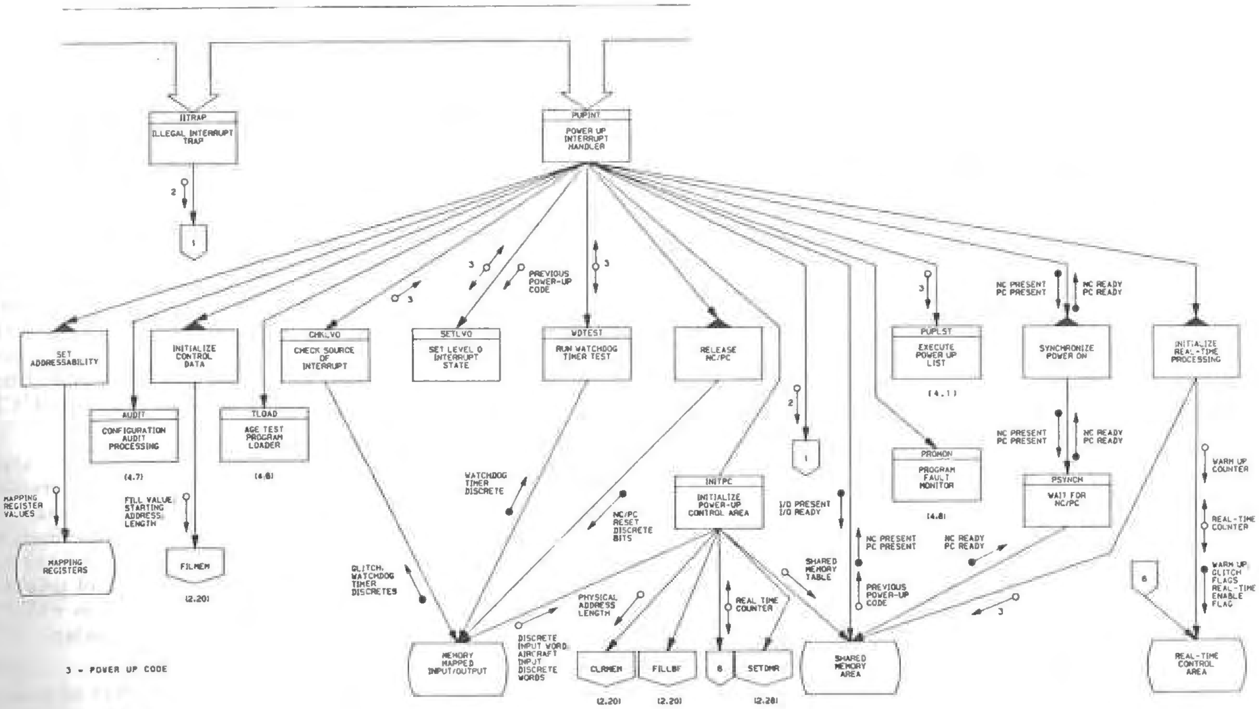


Figure 4 Structure Chart

procedures for maximizing the *cohesion*, or "black boxish" qualities of the modules. and minimizing the *coupling*, or information flow, between them.

Module Specs are the specifications of the modules in the structure charts. describing how their outputs are to be generated from their inputs. and how they are to call their subordinate modules. They are typically written in pseudocode.

Design Dictionary. This is an alphabetical list of all the data and control couples in the structure charts. each with its physical definition and its attributes. such as units. range. precision. etc.

Special Needs of Real-Time Avionics Systems

The basic SA method was developed for non-real-time systems as in banking and other commercial applications. and because of the nature of such systems. the philosophy is to largely defer control and timing considerations to the design phase. In real-time avionics systems. however. control and timing requirements are as complex and important as data processing requirements.

Of the specific characteristics which distinguish real-time (RT) from non-real-time systems. two are particularly important here. First. RT systems contain two distinct types of signals one type is the familiar data signal (data flow) which is used within data processes: the other type has the primary purpose of modifying the response of the system to incoming data rather than to be processed by it. Second. RT systems are required to recognize past events. current status. and expected future events and. again. to modify system response accordingly.

A further consideration in the development of commercial avionics systems is the fact that they must be subjected to the very exacting requirements of FAA certification. such as demonstration of the fact that the system will not fail in ways which will impair the continued safe flight of the aircraft. and that it will not present false or misleading information to the crew. The need to demonstrate these characteristics in systems as complex as FMCS makes it almost mandatory to present all the requirements and design data to the FAA in an orderly and structured manner (5).

Extensions to the Structured Analysis Method

Recognizing the effectiveness of the basic Yourdon SA method within its own scope. the goals of that method were adopted for the extended method, namely:

- Rigor
- Completeness
- Understandability
- Changeability and Maintainability

In addition. it was desired to minimize the changes to the basic method. and to adopt as many of its features into the extended method as possible. In this way best advantage would be taken of all the experience invested in the basic method and the transition from one method to the other would be made as easy as possible.

Much of the strength of basic SA lies in its practicality which arises from features such as leveling. balancing. its numbering system. and the diagrammatic representation described earlier. These. in particular then. were the features desired to be included in the extended method. The characteristic *lacking* in basic SA is the ability to represent control requirements. typically involving complex combinational and sequential logic. In this area it was noted that a large body of knowledge and experience existed based on Finite State machine theory.

The two distinguishing properties of RT systems mentioned earlier give rise. directly. to the two principle new features of the real-time SA method. First. signals are divided into two types data signals (as in basic SA) and *control signals*--while flow diagrams are similarly divided into data flow diagrams and *control flow diagrams* (CFDs) (note that. as will be discussed later. the latter are *not* state transition diagrams). Second. a new type of spec is introduced the *control spec*—which represents the finite state (FS) machine characteristics of the system (and which *may contain* state transition diagrams). To distinguish them from control specs. mini-specs are renamed *process specs*. In fact, several minor changes in terminology have been adopted and are listed below.

Having decided to separate signals into two types it becomes necessary to define how to make that separation in practice.

Changes in Terminology

Real-Time Method	Basic Method
Data context diagram	Context diagram
Control context diagram	None
Data flow diagram	Data flow diagram
Control flow diagram	None
Process spec	Mini-spec
Control spec	None
Timing spec	None
Requirements dictionary	Data dictionary

As is common with many features of structured methods. there are no absolute rules. but some guidelines were established. Any signal representing a continuous physical quantity must be categorized as data. but discrete-valued signals are not always so easily dealt with. It was found that a good approach is to refer back to the original principle: if a signal is used within a process as part of a calculation. categorize it as data. if it is used to modify the response of the system to other signals. categorize it as contro:. It sometimes happens that a signal is used for both purposes. in which case it is categorized as both. and appears both in the DFDs and CFDs.

The primary purpose of the FS machine attributes of the system is to modify the response of the system according to past. current. and expected future conditions. It does this by controlling processes (that is. activating and deactivating them) and can conveniently be thought of in the same way as a feedback control loop in control system theory. Figure 5 illustrates this concept. A second purpose of the FS machine is to signal the status of the system to other systems. and this is done through the control outputs shown.

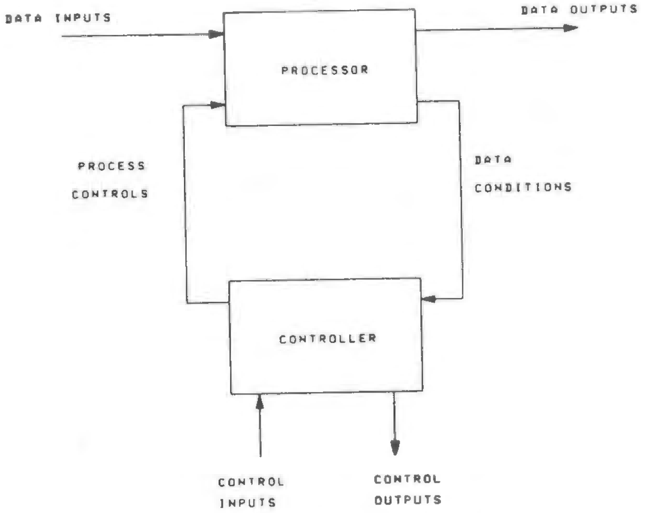


Figure 5 Feedback Control Model

Two additional terms are introduced in ngure 5: *process controls* and *data conditions*. Process controls are the signals which activate and de-activate processes in the data processor. and data conditions are control signals derived through tests on data: for example:

If ALTITUDE > 18000ft.

set HIALT = TRUE

in which HIALT is a data condition.

Integration with the Basic SA Method

It is characteristic of control system design that the entity to be controlled is defined first. since only then can the controlling mechanism be defined. For example. in designing a feedback power amplifier. the output stage must first be designed to drive the required load. then the number of stages and the loop transfer function can be calculated to suit the requirements of the output stage.

This principle was used in structuring the real-time SA method. Since the main purpose of the FS machine is to control the data processor. its structure is slaved to that of the data flow structure. Specifically. control signals are constrained to flow only along the same routes as data signals. and each control spec is associated with one and only one DFD the one whose processes it controls. Thus. each CFD must correspond with a particular DFD and must have the same name and number as that DFD. and each process on that CFD must have the same name and number as a process on the corresponding DFD. Also. a control spec must have the same name and number as its corresponding DFD/CFD pair.

This "slaving" of the control structure to the data structure gives rise to very tightly coupled groups of diagrams: a DFD. a CFD. and a control spec. all with the same name and number. All the inputs to the control spec come from the corresponding CFD and the two must balance. All the outputs from the control spec are either activators of processes on the corresponding DFD. or new control signals which go directly to the corresponding CFD and must balance with it.

This structure also has the very desirable effect of concentrating control requirements close to where they are used. yet there is no loss of flexibility. for the control signals used in the control requirements may flow within the structure in just the same way as the data signals do. The control requirements simply get partitioned in the same way as the processing requirements.

Figure 5 may be thought of as being repeated at each level in the structure with the "controller" block divided into CFDs and control specs. Figure 6 illustrates one level of this configuration. Figure 7 illustrates the composite structure of the extended method by showing one "string" from the "pyramid" of leveled DFDs. CFDs. and control specs.

The following paragraphs describe each of the components of this structure.

Data Flow Diagrams

DFDs are essentially identical to those in basic SA. The one exception is the appearance of data conditions (described earlier) flowing out of the primitive processes in which they are generated. They are shown there to complete the picture of the process, and are also shown flowing out of the same process on the corresponding CFD. Any further flow. to higher or lower levels. is shown only on the CFDs. as with all the other control signals. Figure 8 is a typical DFD with data conditions.

Process activators are not shown at all on the DFDs. only in the control specs. Since. in the completed requirements document. a control spec is located close to its DFD. it is easy to refer to it to find which processes are activated. Processes which do not have activators operate in the same way as in basic SA they are data triggered.

Control Flow Diagrams

The term "control flow diagram" is sometimes used synonymously with "state transition diagram". but this is *not* its meaning in this method. Here, the term is used because the diagram it describes is very similar to a DFD. so it is appropriate for them to have correspondingly similar names. State transition diagrams are referred to exclusively as such in this method. and appear only in control specs.

Figure 9 shows a typical CFD. Like DFDs. CFDs contain processes. signal flows. and stores. and must balance with their parent and child diagrams. The differences between the two are important. however. and are as follows:

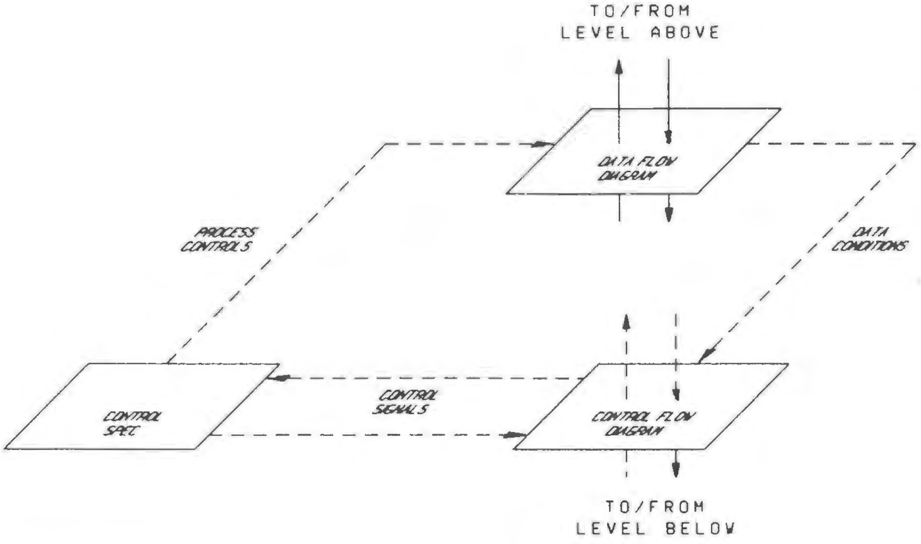


Figure 6 One Level of DFD. CFD and Control Spec

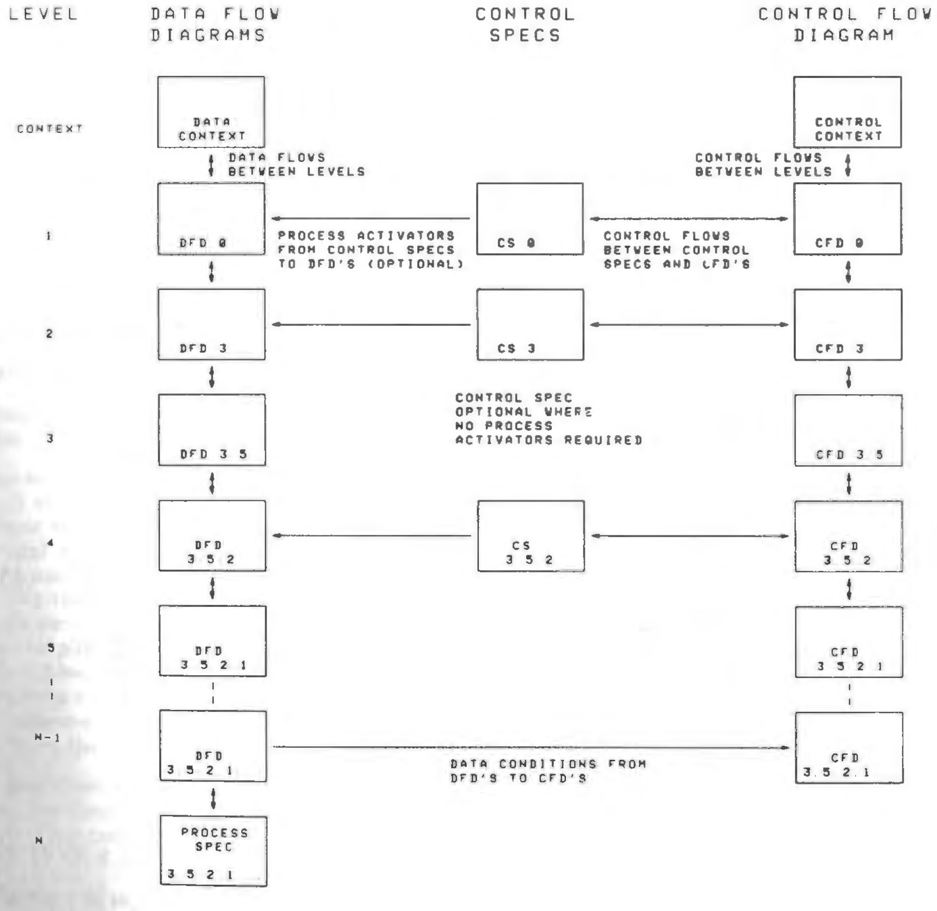
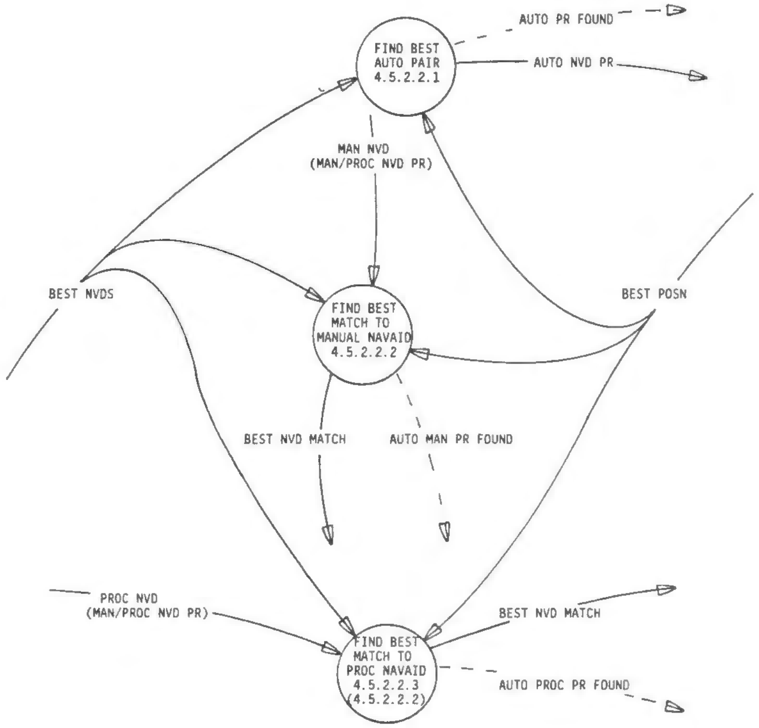
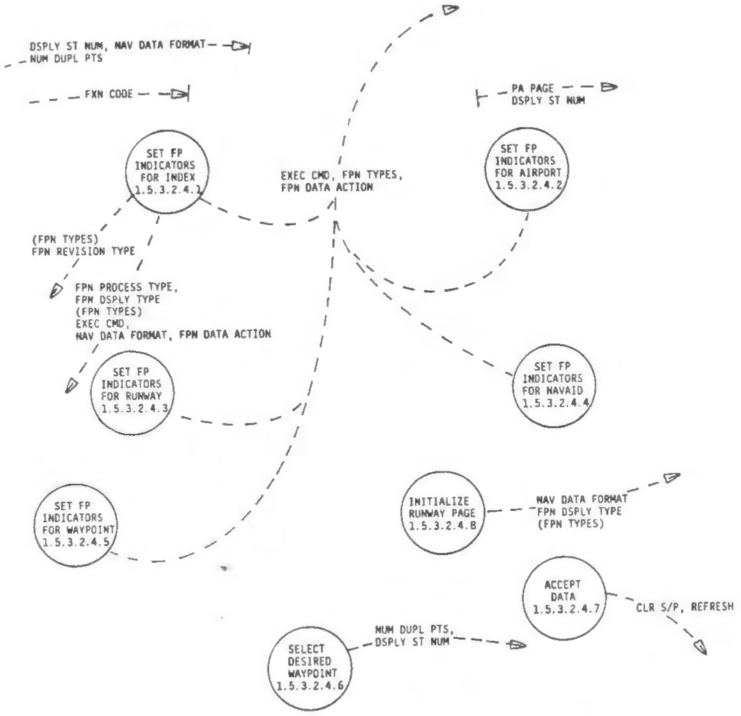

LEVEL
DATA FLOW DIAGRAMS
CONTROL SPECS
CONTROL FLOW DIAGRAM
CONTEXT
DATA CONTEXT
CONTROL CONTEXT
DATA FLOWS BETWEEN LEVELS
CONTROL FLOWS BETWEEN LEVELS
1
DFD 0
PROCESS ACTIVATORS FROM CONTROL SPECS TO DFD'S (OPTIONAL)
CS 0
CONTROL FLOWS BETWEEN CONTROL SPECS AND CFD'S
CFD 0
2
DFD 3
CS 3
CFD 3
3
DFD 3 5
CONTROL SPEC OPTIONAL WHERE NO PROCESS ACTIVATORS REQUIRED
CFD 3 5
4
DFD 3 5 2
CS 3 5 2
CFD 3 5 2
5
DFD 3 5 2 1
CFD 3 5 2 1
N-1
DFD 3 5 2 1
DATA CONDITIONS FROM DFD'S TO CFD'S
CFD 3.5 2.1
N
PROCESS SPEC 3 5 2 1

Figure 7 Composite Diagram of the Method Structure


AUTO PR FOUND
FIND BEST AUTO PAIR 4.5.2.2.1
AUTO NVD PR
MAN NVD (MAN/PROC NVD PR)
BEST NVDS
FIND BEST MATCH TO MANUAL NAVAID 4.5.2.2.2
BEST POSN
BEST NVD MATCH
AUTO MAN PR FOUND
PROC NVD (MAN/PROC NVD PR)
FIND BEST MATCH TO PROC NAVAID 4.5.2.2.3 (4.5.2.2.2)
BEST NVD MATCH
AUTO PROC PR FOUND

Process No. 4.5.2.2 Find Best DME Pair

Figure 8 Data Flow Diagram with Data Conditions


DSPLY ST NUM, NAV DATA FORMAT
NUM DUPL PTS
FXN CODE
PA PAGE
DSPLY ST NUM
SET FP INDICATORS FOR INDEX 1.5.3.2.4.1
EXEC CMD, FPN TYPES, FPN DATA ACTION
SET FP INDICATORS FOR AIRPORT 1.5.3.2.4.2
(FPN TYPES)
FPN REVISION TYPE
FPN PROCESS TYPE, FPN DSPLY TYPE (FPN TYPES) EXEC CMD, NAV DATA FORMAT, FPN DATA ACTION
SET FP INDICATORS FOR RUNWAY 1.5.3.2.4.3
SET FP INDICATORS FOR NAVAID 1.5.3.2.4.4
SET FP INDICATORS FOR WAYPOINT 1.5.3.2.4.5
INITIALIZE RUNWAY PAGE 1.5.3.2.4.8
NAV DATA FORMAT
FPN DSPLY TYPE
(FPN TYPES)
ACCEPT DATA 1.5.3.2.4.7
CLR S/P, REFRESH
NUM DUPL PTS,
DSPLY ST NUM
SELECT DESIRED WAYPOINT 1.5.3.2.4.6

Control No. 1.5.3.2.4 Set FP Indicators for NAV Data

Figure 9 Control Flow Diagram

- their signal flows are control signal flows, and are shown with broken lines to distinguish them from data signal flows.
- signals flowing to and from the associated control spec are shown with a short bar on the end of the vector. This is the only new symbol introduced into flow diagrams.
- consistent with the principle of "slaving" the control structure to the data structure, the processes on a CFD are duplicates of those on the corresponding DFD. If a particular process on the DFD has no control signal flows associated with it, it may be omitted from the CFD.

It is not required that every DFD should have a CFD and control spec associated with it. If none of the processes in the DFD is controlled, then a control spec is not required, and if none of the children of the DFD has any control signals associated it, then no CFD is required (no signals flowing down to or up from lower levels). However, if a control spec is needed, then so is a CFD (to provide the inputs and receive any outputs). All of this is illustrated in, and is best understood from, Figure 7.

It is important not to misinterpret the control signals flowing into and out of processes on a CFD. They are *not* activators of those processes, but signals flowing between levels, just like data flows in DFDs. The process activators only appear in control specs.

Control Specs

Control specs represent the control requirements of the system, generally in finite state machine form. Their purpose is analogous to that of process specs—to show how their outputs are generated from their inputs—but they do this using decision tables and state transition diagrams instead of structured English, equations, etc.

FS machines (6) may be categorized into two types: combinational and sequential. In combinational machines, the current outputs and states of the internal elements are determined entirely by the current inputs, that is, they contain no memory. They are represented as a "3-tuple", $\{I, Z, \omega\}$, where:

I is a finite set of input symbols,

Z is a finite set of output symbols,

ω is a mapping of I onto Z—the output or transfer function.

Combinational machines are usually represented by decision tables in which all combinations of the input signal values (i.e. all the input symbols) are listed with their corresponding output signal values (output symbols). In practice, it is usual for the machine to be "incompletely specified", by virtue of the fact that many of the input symbols are of no interest ("don't care" condition). In these cases the table can be greatly simplified. Figure 10 is a typical control spec using decision tables, including generation both of control signals and process controls. The non-zero numbers in the body of the "process activated" table represent activation of the processes in that numerical sequence.

In sequential machines, current outputs and states of the internal elements are determined by current and past values of inputs—that is, they contain memory. They are represented as a "5-tuple", $\{I, Q, Z, \delta, \omega\}$, where:

I is a finite set of input symbols,

Q is a finite set of states,

Z is a finite set of output symbols,

δ is a mapping of $I \times Q$ onto Q—the next state function,

ω is a mapping of $I \times Q$ onto Z—the output function.

Sequential machines may be modelled in a number of ways (6), including the Moore model, in which the output function depends only on the current state, not on the inputs, and the Mealy model, in which the output function depends on both the current state and the inputs. It can be shown that any representation with one of these models has an equivalent representation with the other, but the Moore representation will usually require more states. Because of its greater flexibility, the Mealy model was chosen for this method.

Although the types of system we are dealing with are invariably sequential machines overall, when the control requirements are partitioned as described earlier, it is usually found that the sequential requirements can be concentrated into a few localized areas; that the rest of the control requirements can be represented in combinational machine form (simple decision tables); and that large parts of the system can be represented in basic SA form, with no control structure at all, using the "data triggering" concept.

Sequential machines are represented using state transition diagrams, or various equivalent tabular forms, any of which may be used in a control spec. Figure 11 shows a typical example using a matrix form. In practice, a given control spec containing a sequential machine model will also require some combinational logic applied to the input and output signals.

A useful feature of basic SA is that any one of its component entities is contained on a single sheet of paper—one DFD to a sheet for example. However, since the control requirements associated with a given DFD may be arbitrarily complex, there can be no restriction on the size of a control spec, and they are frequently several pages long. It is important that their input and output signals should be grouped together and clearly identified near the front so that they can be easily balanced with their DFD and CFD. On FMCS, very long control specs were prefaced by a "users' guide".

Requirements Dictionary

The requirements dictionary (RD), is essentially the same as the SA data dictionary, but it contains the definitions of both the data and control signals. The notation is the same for both - control signals are grouped in just the same way as data signals. The RD was automated on the FMCS program (7) using a commercial data management system, and is divided into fields: "name", "composed of", "used in", and "member of". The last two list, respectively, the flow diagrams in which the signal is used, and other signal groups in which it is included.

Timing Requirements

From the requirements point of view, timing falls into just two categories:

- required rates of receiving inputs and generating outputs
- response times from system input events to resulting system output events.

Input and output rates are stated in the requirements dictionary as attributes of the individual primitive signals. Response times are listed in simple tabular form showing the input signal(s), the event associated with those signals, the output signals, and the resulting event associated with those output signals.

Such considerations as timing budgets for software functions or module calling rates have no place in a requirements spec.

Using the Real-Time SA Method

The guidelines for preparing basic SA specs generally apply to real-time SA specs. In addition, some further guidelines have been found useful, as follows:

- The customer spec usually has data and control requirements totally intermixed, so it is necessary to start separating them before starting the analysis. This can be

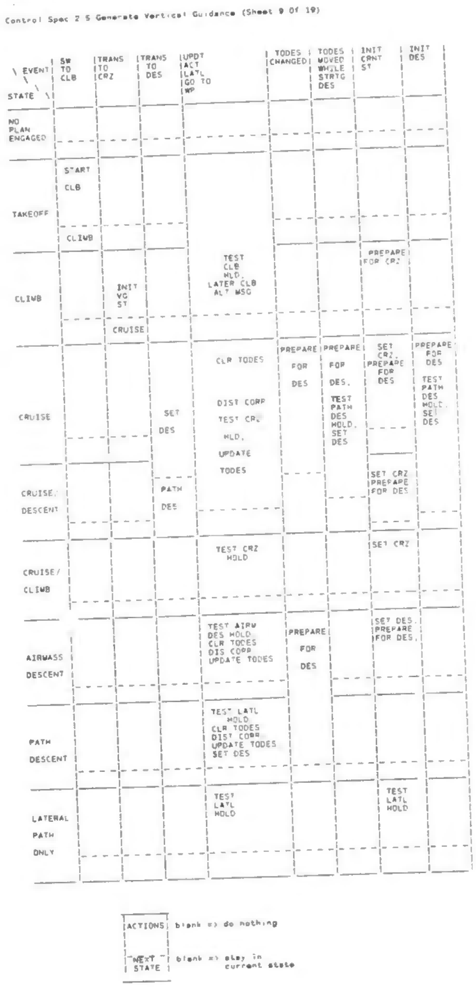
Control 1.5.3.2 Manage Reference Group SEKS (3 of 3)

POS INIT/POS REF Select Enter Keys

CONTROL INPUT				CONTROL OUTPUT				PROCESS ACTIVATED 1.5.3.2.N			
DSPLY ST NUM	FXN CODE	MOD RTE INDIC	ACT RTE INDIC	PA PAGE	DSPLY ST NUM	DSPLY RTE STATUS	DSPLY PG NUM	2	3	5	6
3, 4	6L			None	INIT/REF INDEX			0	0	0	0
	6R	True	False	None	ROUTE	Modified	1				
		False	True			Active					
		Otherwise				Inactive		1	0	2	
3	2L							0	1	0	2
	5L,4R, 5R										

PROCESS NO.	PROCESS NAME
1.5.3.2.2	Enter Reference Airport
1.5.3.2.3	Select POS INIT Variable
1.5.3.2.5	Clear Scratchpad
1.5.3.2.6	Check for Valid IO Data

Figure 10 Decision Table



Control Spec 2.5 Generate Vertical Guidance (Sheet 9 Of 19)

\ EVENT \ STATE \	SW TO CLB	TRANS TO CRZ	TRANS TO DES	UPDT ACT LATL GO TO WP	TODES CHANGED	TODES MOVED WHILE STRTG DES	INIT CRNT ST	INIT DES
NO PLAN ENGAGED								
TAKEOFF	START CLB CLIMB							
CLIMB		INIT VG ST CRUISE		TEST CLB HLD, LATER CLB ALT MSG			PREPARE FOR CRZ	
CRUISE			SET DES PATH DES	CLR TODES DIST CORR TEST CRZ HLD, UPDATE TODES	PREPARE FOR DES	PREPARE FOR DES, TEST PATH DES HOLD, SET DES	SET CRZ, PREPARE FOR DES	PREPARE FOR DES TEST PATH DES HOLD, SET DES
CRUISE/ DESCENT							SET CRZ PREPARE FOR DES	
CRUISE/ CLIMB				TEST CRZ HOLD			SET CRZ	
AIRMASS DESCENT				TEST AIRM DES HOLD CLR TODES DIS CORR UPDATE TODES	PREPARE FOR DES		SET DES, PREPARE FOR DES,	
PATH DESCENT				TEST LATL HOLD CLR TODES DIST CORR UPDATE TODES SET DES				
LATERAL PATH ONLY				TEST LATL HOLD			TEST LATL HOLD	

ACTIONS	blank => do nothing
NEXT STATE	blank => stay in current state

Figure 11 State Transition Matrix

a long and tedious task and, in fact, tends to continue throughout the analysis.

- Work on the data flow structure first, or at least start it first. This follows the principle of defining what has to be controlled before deciding how to control it.
- Try to minimize the amount of control specified. In other words, maximize the amount of basic SA in the spec. Control tends to be design and implementation dependent and the maximum possible freedom should be given to the designer.
- Try to keep as much of the control requirements as possible in combinational machine form (decision tables). This is the simplest and therefore the most desirable form.
- Concentrate the requirements which must be in sequential machine form into localized areas.
- Try to put the control requirements at as high a level as possible in the structure. Control specs typically transform into "boss" modules in the Structured Design, and these decision-making processes should be towards the top of the hierarchy.

Experiences with the Methods

It is usually recommended that new users of structured methods introduce them on a small, low-key project. The project on which this method was introduced was very larg (the SA spec is 15 volumes long) and very critical, and more over, the real-time SA method itself was new and untried To counter these obstacles, all the training was done in house, and a "methods team" was formed at the start of th project, with representatives from engineering managemen systems engineering, software design, software testing, an support software. This team acted as full time consultant to the project staff, providing advice, assistance, and prac tical problem solving on demand.

The level of acceptance varied widely from individual individual, and there were some difficulties in getting co sistent standards from all of the 20 to 25 engineers workin on the requirements definition, but the majority were ov whelmingly positive towards it, and the results have ge erally been excellent. FMCS has performed significant better throughout its development than previous, simil systems.

Customer acceptance, too, has been excellent, and they are using these methods as a model for their own work and as a standard for their other vendors.

As expected, the front end effort to prepare the requirements spec was considerably more than on previous projects, in fact, considerably more than was originally estimated for this project. Nevertheless, the project overall was on schedule, and the results of the additional effort in terms of performance to date, and improved communication with the customer and with the design group, justify this expense.

The most serious shortcoming has to do with the size of the system rather than the method itself: manual implementation is impractical on systems of this size, and full automation is essential in the long term. A number of organizations are now working in this area, and fully automated structured methods tools are expected to be available very soon. Work is also underway to include the extensions to the methods described here into one or more of these tools.

References

(1) G. A. Miller, "The Magical Number, Seven Plus or Minus Two: Some Limits on our Capacity for Processing Information", *The Psychological Review*, Vol. 63, No. 2, Mar. 1956.

(2) D. T. Ross, "Structured Analysis (SA): A Language for Communicating Ideas", *IEEE Transactions on Software Engineering*, Vol. SE-3, No. 1, Jan. 1977.

(3) T. DeMarco, *Structured Analysis and System Specification*, Yourdon Press, 1978.

(4) M. Page-Jones, *The Practical Guide to Structured Systems Design*, Yourdon Press, 1980.

(5) *Software Considerations in Airborne Systems and Equipment Certification Document No. RTCA/DO-178*, Radio Technical Commission for Aeronautics, Nov 1981.

(6) T. L. Booth *Sequential Machines and Automata Theory*, John Wiley and Sons, Inc., 1967.

(7) K. M. Hornbach, "Development Tools Case Study for Large Systems", *Sixth Digital Avionics Systems Conference*, Dec. 1984.

84-2600

A CONCEPTUAL DESIGN METHODODLGY USED TO DESIGN A SUPPORT COMPUTER PROGRAM

Authors: D. J. Berg
J. A. Mozier
R. L. Young

Department of Digital Computers and Software Engineering
McDonnell Aircraft Company
McDonnell Douglas Corporation
St. Louis, Missouri

Abstract

A variation/compilation of some existing methodologies is being used to design an electromagnetic environmental simulation - support computer program. This methodology incorporates models, procedures, and tools that conform to the basic software-life-cycle process, and utilizes cyclic processing and conceptual design decomposition. In this methodology, distinct model types are used at different levels of the decomposition. Each model type can be of either textual or graphical form, or both. Using requirements specifications, user needs are transformed into a Requirements Model. The Requirements Model is transformed into a Data/Function Model using system specifications. The Data/Function Model is transformed, using program design techniques, into a Program Design Model. Finally, using structured programming techniques, the Program Design Model is transformed into the program source code itself. The ultimate goal is to implement a design methodology that consists of a series of steps aimed at changing formal requirements specifications into a design so finely detailed that it can be directly implemented by engineers not familar with the overall design.

I. Background

It is recognized that the software development cycle has at least the following five stages - requirements specification, design, implementation, testing, and maintenance, however this article will only cover the requirements and design techniques used on a specific project. Current literature and department standards cover the topics of structured programming techniques, implementation, testing and maintenance.

The basic methods described here came from the sources referenced at the end of the article. They were modified somewhat, based on the specific experience of the authors and the input of the project and department management.

The specific application on which these methods are implemented is to design a software program called an Electromagnetic Environmental Simulation (EES). The Support Computer Program's (SCP's) prime directive is to simulate an aircraft (A/C) and its mission hardware in an electromagnetic environment. This user controlled simulation is accomplished by emulating the data transfer that occurs between the mission hardware and the Operational Flight Program (OFP) resident in the A/C's mission computer. At McDonnell Aircraft Company this simulation is implemented via a 'host' computer in a Software Test Facility. (see Figure 1)

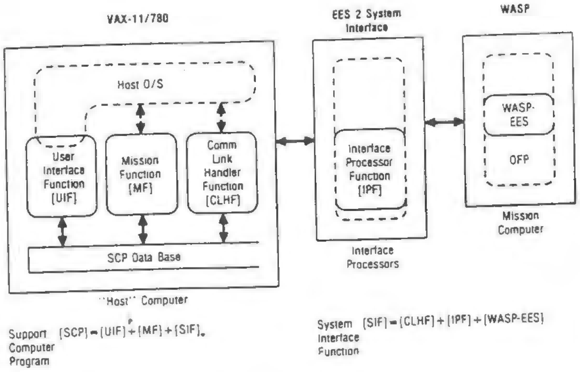

VAX-11/780
Host O/S
User Interface Function [UIF]
Mission Function [MF]
Comm Link Handler Function [CLHF]
SCP Data Base
"Host" Computer
Support Computer Program [SCP] = [UIF] + [MF] + [SIF].
EES 2 System Interface
Interface Processor Function [IPF]
Interface Processors
WASP
WASP-EES
OFP
Mission Computer
System Interface Function [SIF] = [CLHF] + [IPF] + [WASP-EES]

Figure 1 - Software Test Facility

To best design a program under the constraints of the given environment, a variation/compilation of some existing methodologies was necessary.

Jensen and Tonies, in their text Software Engineering, (1979), Chapter 3 - Software Design, present the concepts whose implementation have been described as follows. Requirements Definitions are transformed into the System Design which is then transformed into Program Design.

As early as 1976, Structured Analysis and Design Techniques (SADT), a methodology developed by D. T. Ross, utilized the same concepts shown in a graphical form. His SADT combined a blue-print like graphic language with the nouns and verbs of any other language to provide a top-down structured design model.

Although other design methodologies were considered by the authors, a combination/variation of the two methodologies described above seemed most appropriate to the given task. In order to implement a graphical model, and at the same time meet the stringent requirements of documentation required by a military specification, some variation of the graphical design technique was necessary. A compromise solution was found which takes the specific list of requirements and decompose the requirements categories of a military specification in a one-to-one correspondence with the graphical design decomposition. This not only forms the basis for our requirements model, it also supplies part of the documentation for our Part 1 specification documents. These documents include the functional description of the system.

It was later discovered that Yoshihiro Matsumoto of the Toshiba Corporation has outlined a similar concept in a 1984 IEEE article entitled, "Management of Industrial Software Production".

Because the main points of this article tie so many of these concepts together, excerpts from it will be used as a comparison to the different models used.

II. Introduction

The software design phase, of the EES2-SCP, is most easily described utilizing the following software Design Refinement Level Diagram. (see Figure 2)

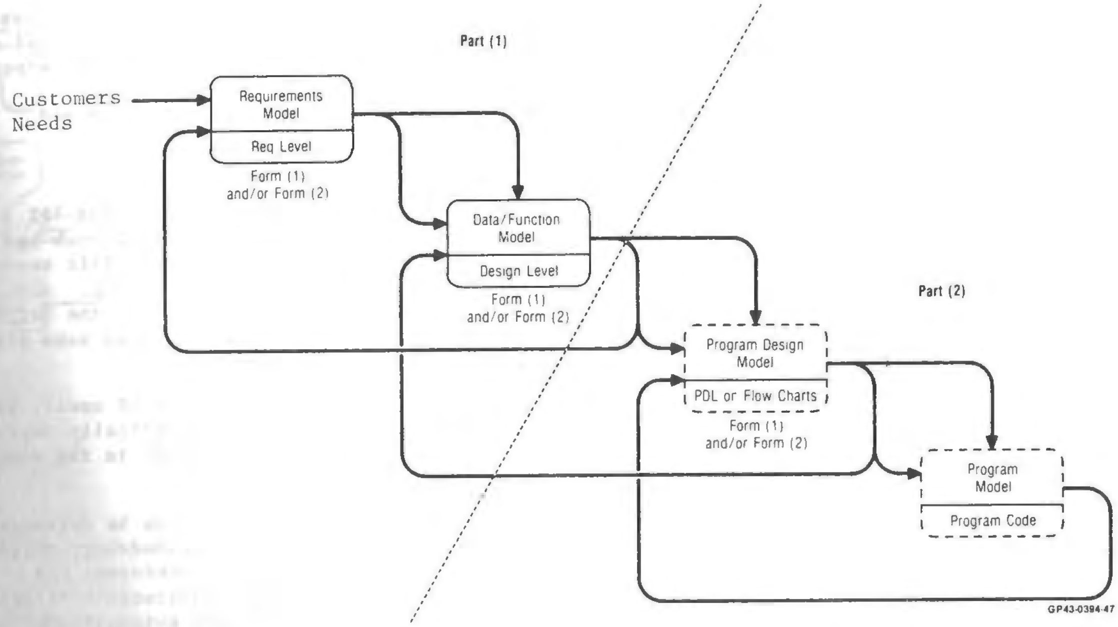


Figure 2 - Design Refinement Level Diagram

Each level of refinement has a specific model type. These models usually come in pairs at each level of abstraction. The model can be one or both of the following forms: Textual Form (Form 1) and/or Diagram Form (Form 2).

1. Classical types of requirements, including user and customer needs, are incorporated into our Software Requirements Document (SRD). This document alone is the textual form of our Requirements Model.

2. The Requirements Model is transformed using system, interface, and functional specifications into the Data/Function Model.

These two models constitute what is considered a Part (1) performance and design requirement for MIL-STD-483.

3. The Data/Function Model is transformed to the Program Design Model using some program design techniques of both SADT and Specification Decomposition.

4. The Program Design Model is transformed using structured programming techniques into the Program Model (program source code).

The last two models are considered the Part (2) product configuration and technical description of the same MIL-STD.

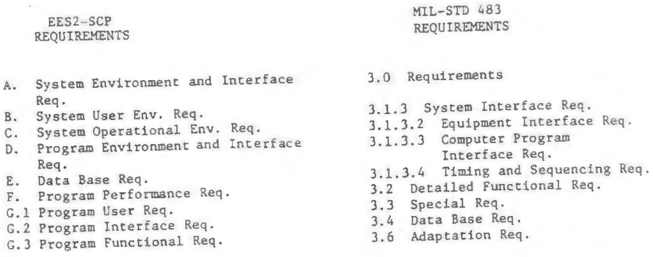
III. Requirements Model

The Requirements Model for the EES2-SCP is in the textual form of a support Software Requirement Document (SRD). Here requirements are typed according to the following general categories, (see Figure 3) which are 1) System Environment and Interface Requirements (both hardware and software), 2) Support Computer Program (SCP) Environment and Interface Requirements, and 3) Functional Requirements. These requirements are mapped into the specific categories given in MIL-STD-483 which includes: 1) System Interface Requirements, (both hardware and software), 2) Computer Program Interface Requirements, and 3) Detailed Functional Requirements This mapping serves two purposes. First, it puts the SRD in the general form of the requirements document for a Part (1) specification. Second, it forms the basis for the Detailed Functional Requirements which can be used as the textual form of a Data/Function Model.

The engineer continually refines, through structured decomposition methods, the Requirements Specification of MIL-STD-483, Appendix VI.

IV. Data/Function Model

The Data/Function Model is that model where the System, Interface, and Top Level Functional Design are described both textually and graphically. It is at this level that data structures, functions,



EES2-SCP
REQUIREMENTS

A. System Environment and Interface Req.
B. System User Env. Req.
C. System Operational Env. Req.
D. Program Environment and Interface Req.
E. Data Base Req.
F. Program Performance Req.
G.1 Program User Req.
G.2 Program Interface Req.
G.3 Program Functional Req.

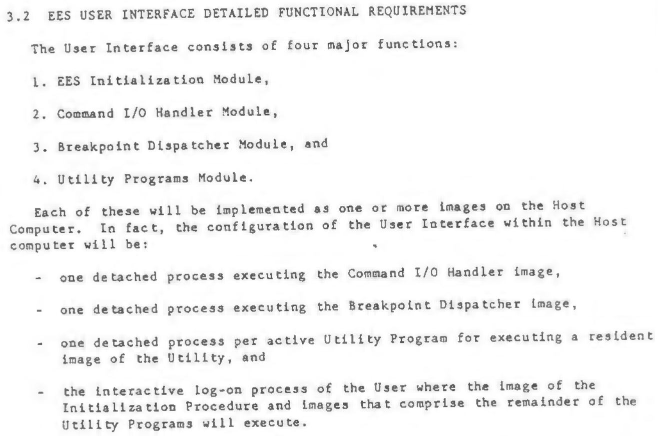
MIL-STD 483
REQUIREMENTS

3.0 Requirements

3.1.3 System Interface Req.
3.1.3.2 Equipment Interface Req.
3.1.3.3 Computer Program Interface Req.
3.1.3.4 Timing and Sequencing Req.
3.2 Detailed Functional Req.
3.3 Special Req.
3.4 Data Base Req.
3.6 Adaptation Req.

Figure 3 - Requirement Categories

data flows, and control flows appear. The textual form of this model is the structurally decomposed (more detailed) Requirement Specification, now called a Data/Function Specification. (see Figure 4).



3.2 EES USER INTERFACE DETAILED FUNCTIONAL REQUIREMENTS

The User Interface consists of four major functions:

1. EES Initialization Module,

2. Command I/O Handler Module,

3. Breakpoint Dispatcher Module, and

4. Utility Programs Module.

Each of these will be implemented as one or more images on the Host Computer. In fact, the configuration of the User Interface within the Host computer will be:

- one detached process executing the Command I/O Handler image,
- one detached process executing the Breakpoint Dispatcher image,
- one detached process per active Utility Program for executing a resident image of the Utility, and
- the interactive log-on process of the User where the image of the Initialization Procedure and images that comprise the remainder of the Utility Programs will execute.

Figure 4 - Detailed Function Requirement

The graphical form of this model is two-fold. (1) a modified Data Flow Diagram (DFD) of the SADT form, and (2) a Functional Flow Diagram (FFD). (see Figures 5 and 6).

The DFD's are used to show the functions performed to accomplish a specific state change. The DFD's describe the dynamic relationship between memory, I/O Devices, and the processes themselves via the flow of data from one to another. Some control ability is added to the DFD's as a way to show the interconnection of processes that was previously not available on other types DFD's. This limited control is also used to show the interdependence of the DFD's and the FFD's. The FFD's along with the DFD's, instead of the traditional structure charts, provide the link between the abstract processes of the DFD's and the physical modules, subroutines, and programs to be developed. The FFD's also serve another purpose. They are the required form for demonstrating the functional task description of the specific computer program in present Preliminary Design Reviews (PDR's).

V. Program Design Model

The Program Design Model is where the program configuration, the data structures, the package interfaces, etc., are put together. Again, decomposition of the Data/Function specification forms the textual part of the model. DFD's are used sparsely, and the main emphasis is on Math Flows. (see Figure 7)

Subroutine logic is described at this level, this is the design level from which code is derived. Software Engineering Department standards, (not described here), for documentati of the Math Flows are adhered to.

VI. Program Model

The Program Model corresponds to the actual code. The particular application under discussi implemented Fortran 77 due to the mathematical a I/O capabilities of the language. As this aspec of the design is the most documented and understood, this model will not be addressed.

VII. Conclusions

Although the EES2-SCP project is about 75% complete at the writing of this paper, much has been learned by the authors about specific needs the design phase of Software Engineering. Simul neously with the ongoing project work, the Softw Engineering Department is studying this same are and has made similar conclusions.

The authors present below a list of needs, ba on experience, to enhance and dramatically impro productivity as related specifically to the desi phase of the software-life-cycle.

1. A Design Methodology needs to be develope and standardized. This methodology shou incorporate the tools, procedures, and techniques to support requirements defin tion, program design, and automation of many of the tasks associated with the software development process as current technology can provide (e.g. documentati reusable software, variable model forms, code generation, etc.).

2. Methods, procedures, and models need to include structures to handle large real-time, multi-processing, multi-taski simulations and programs. Specific mean are needed for showing Operating System Program Communication, timing and sequen of program events on the same model structure, interrupt and time - queue handling, etc.

3. Methods are needed to resolve the differences in what the programmer/desig with an object oriented programming background 'sees' as task definition in software design. Object oriented programming replaces the OPERATOR/OPERAN and INPUT/PROCESS/OUTPUT concepts with MESSAGE/OBJECT and REQUEST TO PERFORM/ OPERATIONS ON/DATA. This is stated to l the advantage in that the decision of ho command is implemented is made by the Ol that performs the command, not by the environment of the operation. This difference in task definition requires a methodology that can lend itself to the ideas of object oriented design, but be flexible enough to handle the Process (Functional) decomposition of many pres techniques.

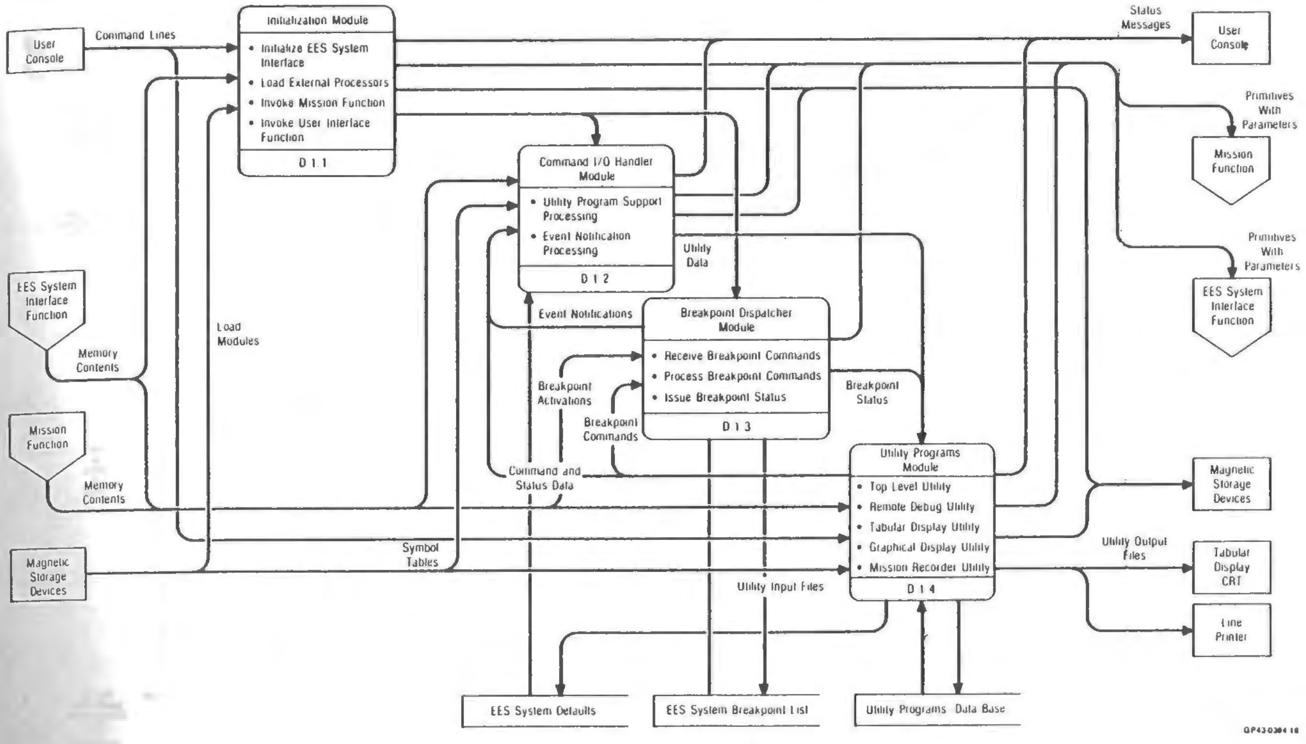

User Console
Command Lines
Initialization Module
• Initialize EES System Interface
• Load External Processors
• Invoke Mission Function
• Invoke User Interface Function
D 1.1
Status Messages
User Console
Primitives With Parameters
Mission Function
Command I/O Handler Module
• Utility Program Support Processing
• Event Notification Processing
D 1.2
Utility Data
Primitives With Parameters
EES System Interface Function
EES System Interface Function
Memory Contents
Load Modules
Event Notifications
Breakpoint Dispatcher Module
• Receive Breakpoint Commands
• Process Breakpoint Commands
• Issue Breakpoint Status
D 1.3
Breakpoint Activations
Breakpoint Commands
Breakpoint Status
Mission Function
Memory Contents
Command and Status Data
Utility Programs Module
• Top Level Utility
• Remote Debug Utility
• Tabular Display Utility
• Graphical Display Utility
• Mission Recorder Utility
D 1.4
Magnetic Storage Devices
Symbol Tables
Magnetic Storage Devices
Utility Output Files
Tabular Display CRT
Line Printer
Utility Input Files
EES System Defaults
EES System Breakpoint List
Utility Programs Data Base
GP43-0384-18

Figure 5 - User Interface Functions DFD

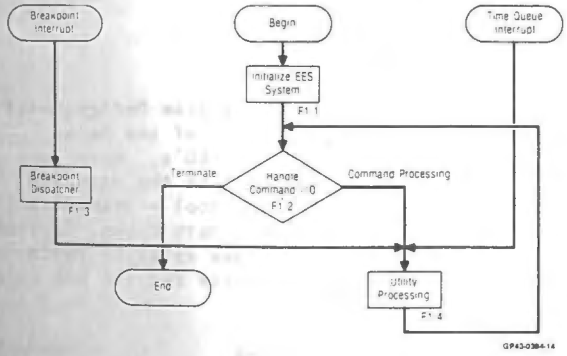


Figure 6 - User Interface Function FFD

VIII. Recommendations

The recommendations described in this section are based on the experiences of the authors, which includes a considerable amount of time spent in hand generated textual and graphical models, and in documentation. Many of these concerns have been addressed by a methodology group of a software technology committee, whose recommendations are based on Dr. Matsumotos article. The recommendations are as follows: A graphical model, using structured decomposition would be most useful for 'seeing' large scale problem definition, but the state of computer generation of these models is not immediately ready for implementation at the design level. It was concluded that software development in the form of textual model packages would be easier to implement immediately. Therefore, a PDL should be encouraged at all levels of design. Then, an automated tool set should be developed to generate the graphical model from the textual model. (see Figure 8)

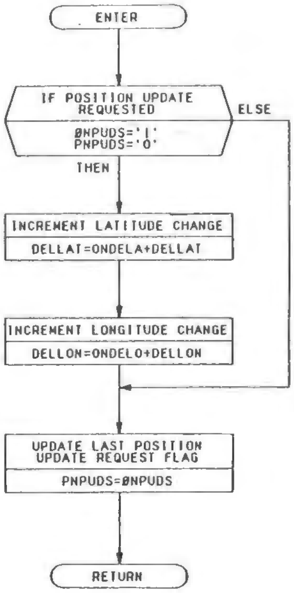

ENTER
IF POSITION UPDATE REQUESTED
ØNPUDS='1'
PNPUDS='0'
ELSE
THEN
INCREMENT LATITUDE CHANGE
DELLAT=ONDELA+DELLAT
INCREMENT LONGITUDE CHANGE
DELLON=ONDELO+DELLON
UPDATE LAST POSITION UPDATE REQUEST FLAG
PNPUDS=ØNPUDS
RETURN

Figure 7 - INS - Lat/Lon Update Subroutine Math Flow

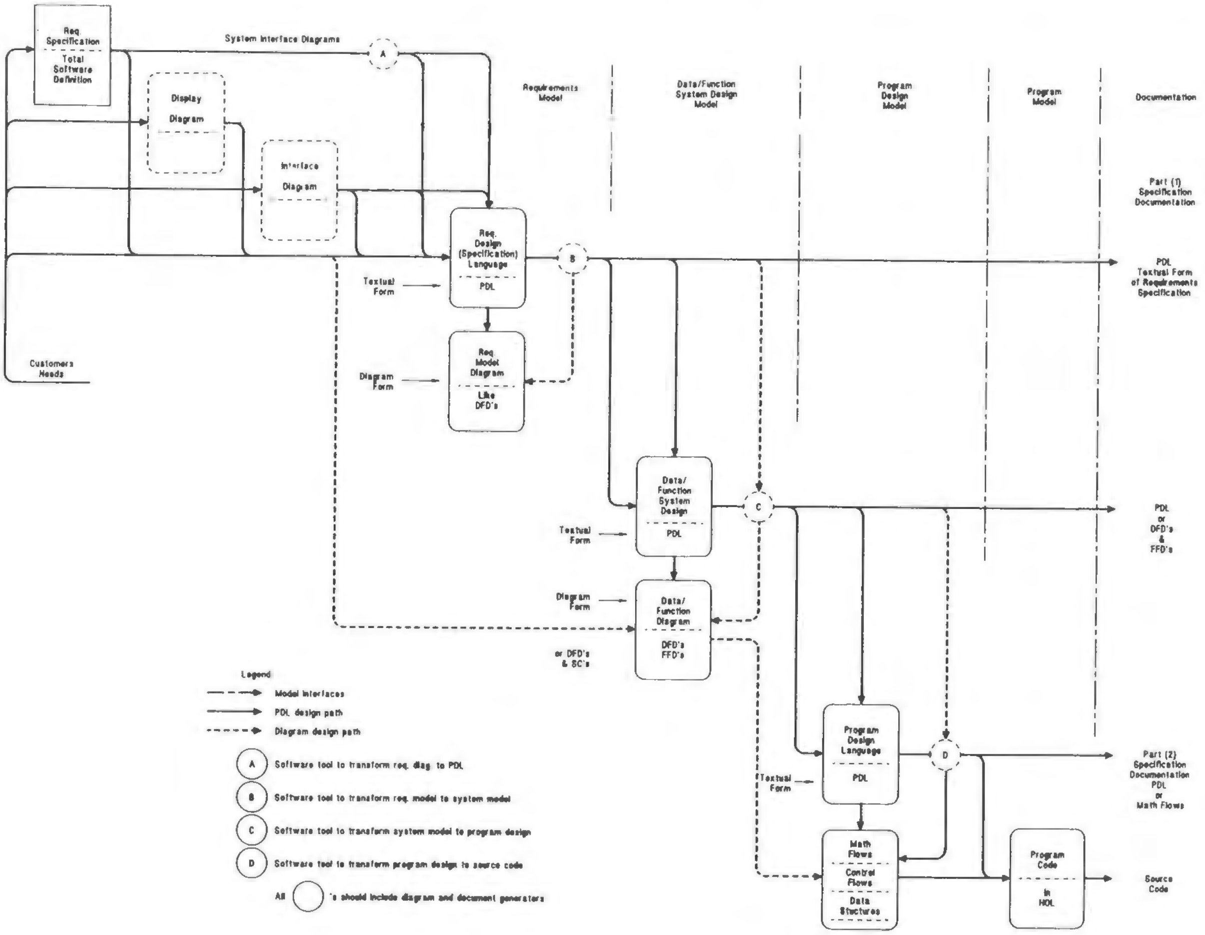

Req. Specification
Total Software Definition
System Interface Diagrams
A
Display Diagram
Interface Diagram
Requirements Model
Data/Function System Design Model
Program Design Model
Program Model
Documentation
Part (1) Specification Documentation
Req. Design (Specification) Language
PDL
B
PDL Textual Form of Requirements Specification
Textual Form
Diagram Form
Req. Model Diagram
Like DFD's
Customers Needs
Data/ Function System Design
PDL
C
PDL or DFD's & FFD's
Textual Form
Diagram Form
Data/ Function Diagram
DFD's FFD's
or DFD's & SC's
Legend
Model interfaces
PDL design path
Diagram design path
A Software tool to transform req. diag. to PDL
B Software tool to transform req. model to system model
C Software tool to transform system model to program design
D Software tool to transform program design to source code
All ○ 's should include diagram and document generators
Program Design Language
PDL
Textual Form
D
Part (2) Specification Documentation PDL or Math Flows
Math Flows
Control Flows
Data Stuctures
Program Code
In HOL
Source Code

Figure 8 - Revised Design Refinement Level Diagram

Reference Section III. The EES2-SCP Requirement Model was written in the form of a Part (1) Specificatation according to a military standard into specific requirement categories. An improvement on this would be to have the designer input the user needs, using an automated software tool - transform (A), into a PDL with special syntax for this level. This model, in PDL form, should have the accepted main entities (categories) of object oriented design requirements. Another software tool - transform (B), could then generate Requirement Model Diagrams, and implement the transformation to the Data/Function model by setting up a template for the PDL at the next level of refinement.

Reference Section IV. Requirement (now Data/Function) Specifications, and DFD's were used as the model for the SCP. Here again, a Data/Function system design PDL is recommended. It should have the form and expanded syntax to be able to describe the program at this level. A software tool - transform (C), should generate the diagram form of this model. A Data/Function diagram, as described by Dr. Matsumoto is also recommended. It has the advantage of providing the designer with a means of viewing functional distribution, control flows, and data flows all on the same diagram. This tool should then set up a template for the PDL at the next level of refinement.

Reference Section V. The Program Design Model was implemented by decomposition of the Data/Function specification and the DFD's. Here, the textual form of the model could be the standard form of any PDL. An automated tool - transform (D) should be used to generate any Math Flows, Control Flows, or Data Structures. Some existing tools now use a PDL and set up the template for the HOL code, as is true for ADL and ADA.

References

1. Yoshihiro Matsumoto, "Management of Industrial Software Production", Computer, pp. 59-71, February 1984.

2. R. W. Jensen and C. C. Tonies, Software Engineering, Prentice - Hall Inc, 1979, pp. 64-221.

3. D. T. Ross, "Structured Analysis (SA): A Language for Communication Ideas", IEEE Transactions on Software Engineering, January 1977.

4. J. Baechle, D. Berg, M. A. Cowan, F. Houska, D. Merrick, C. Post, K. S. Tracy, P. J. Wilson, "Software Methodology Recommendation Report", Memo 312-199, Department of Digital Computer and Software Engineering, McDonnell Aircraft Company, July 1984.

SESSION 2 COMMERCIAL TRANSPORT AVIONICS

Chairmen:

Richard L. Heimbolt
Lockheed Corp.

James H. Shannon
Douglas Aircraft Co.

This session deals with the application of avionics to commercial airlines with emphasis on new avionics and on operator experience with new technologies.

DIGITAL FUEL QUANTITY INDICATING SYSTEM FOR AIRCRAFT

84-2602

Ronald L. Newport
Principal System Design Engineer

Donald J. Nelson
Senior System Design Engineer

Mark T. Manfred
Principal System Design Engineer

Commercial Aviation Operations
Honeywell Inc.
Minneapolis, Minnesota

Abstract

The digital fuel quantity indicating system is a state-of-the-art, high-accuracy system which has been developed for the new generation of commercial transport aircraft. Improved aircraft operating efficiency and simplified maintenance procedures are benefits provided by this system. The paper presents a description of the system and system output functions, the architecture of the system, and the operation of the system, including fuel measurement, fueling control and built-in test functions. Several new system features are discussed, such as linear uncharacterized fuel sensors and the densitometer, which contribute to the benefits provided by this system.

Introduction

Honeywell has developed a state-of-the-art digital fuel quantity indicating system (FQIS) for use on the new Boeing 767 and 757 commercial transport aircraft. New characteristics embodied in this system enhance operational performance in several ways:

a. Accuracy. The FQIS has demonstrated ±1% of full-scale accuracy (for each tank) on the ground and ±2% in the air compared with ±3% of full-scale accuracy on the ground and ±4% in the air, typical of previous systems. The incorporation of density measurement techniques allows the system to be tolerant of changes in fuel composition without accuracy degradation. This system also has the ability to maintain system accuracy in the presence of substantially high levels of fuel contamination.

b. Elimination of On-Board Calibration. The digital nature of the equipment and tolerances maintained during manufacture of the equipment result in components which are interchangeable and replaceable without calibration or adjustment on the aircraft.

c. Caution Alert Signals. Caution alert signals are provided by the FQIS to the flight crew to identify potential fuel configuration problems, such as low fuel levels in the main tanks, a lateral fuel load imbalance between the two main tanks or a measurable fuel quantity remaining in the center auxiliary tank with the associated fuel pumps turned off.

d. Self-Monitoring. Built-In-Test-Equipment (BITE) features provide fault detection and isolation to individual system component level, including wiring to individual capacitive sensors, thus simplifying on-aircraft maintenance actions. Fault isolation to the individual circuit card of the processor unit is also provided to simplify in-shop maintenance actions. In addition, BITE provides continuous monitoring and annunciation of the system operational status to the flight crew.

e. Fault Tolerance. The dual-redundant hardware configuration of the processor unit and innovative sensor substitution software algorithms, in many cases, permit uninterrupted operation in spite of the presence of a system fault.

Background

Fuel quantity measurement in the fuel tanks of commercial transport aircraft, as well as all other high-performance aircraft, has been based on a capacitance technique employing vertically oriented, cylindrical sensors. These sensors, consisting of inner and outer coaxial electrodes, are mounted in the tanks so that the fuel level in the sensor corresponds to that of the tank itself. Since fuel has a dielectric constant approximately twice that of air, the capacitance of the sensors will roughly double in value when fully immersed in fuel, compared with its value in air. This change in capacitance provides a means of determining the quantity of fuel in the tank.

Since aircraft fuel tanks have an irregular shape, and the airplane operates over a wide range of attitudes, each tank requires a number of sensors in order to measure fuel quantity accurately. To compensate for the tank shape and for the range of attitudes, the sensors in previous systems are electrically wired in parallel and physically profiled, typically by varying the diameter of the inner electrode, so that the total capacitance change per unit of fuel volume is ideally a constant. Determining the optimum profiling for each sensor in order to minimize errors due to tank orientation is a complex task requiring techniques such as linear programming. The result is a compromise for all factors of orientation, tank shape, and fuel volumes, since only one profile can be built into a sensor, thereby limiting the achievable overall system accuracy. Improvement in accuracy can only be obtained with this type of system by adding more sensors.

Fuel quantity information is generally displayed in terms of weight or mass, since the energy content of the fuel load is more a function of mass than of volume. The capacitance type of measurement system is adapted for mass measurement because there is a predictable relationship between density and dielectric constant. However, since aircraft fuels are mixtures of various hydrocarbon compounds, errors of up to 2% can be encountered with the conventional system due only to variations in fuel composition.

DIGITAL FUEL QUANTITY INDICATING SYSTEM FOR AIRCRAFT

84-2602

Ronald L. Newport
Principal System Design Engineer

Donald J. Nelson
Senior System Design Engineer

Mark T. Manfred
Principal System Design Engineer

Commercial Aviation Operations
Honeywell Inc.
Minneapolis, Minnesota

Abstract

The digital fuel quantity indicating system is a state-of-the-art, high-accuracy system which has been developed for the new generation of commercial transport aircraft. Improved aircraft operating efficiency and simplified maintenance procedures are benefits provided by this system. The paper presents a description of the system and system output functions, the architecture of the system, and the operation of the system, including fuel measurement, fueling control and built-in test functions. Several new system features are discussed, such as linear uncharacterized fuel sensors and the densitometer, which contribute to the benefits provided by this system.

Introduction

Honeywell has developed a state-of-the-art digital fuel quantity indicating system (FQIS) for use on the new Boeing 767 and 757 commercial transport aircraft. New characteristics embodied in this system enhance operational performance in several ways:

a. Accuracy. The FQIS has demonstrated ±1% of full-scale accuracy (for each tank) on the ground and ±2% in the air compared with ±3% of full-scale accuracy on the ground and ±4% in the air, typical of previous systems. The incorporation of density measurement techniques allows the system to be tolerant of changes in fuel composition without accuracy degradation. This system also has the ability to maintain system accuracy in the presence of substantially high levels of fuel contamination.

b. Elimination of On-Board Calibration. The digital nature of the equipment and tolerances maintained during manufacture of the equipment result in components which are interchangeable and replaceable without calibration or adjustment on the aircraft.

c. Caution Alert Signals. Caution alert signals are provided by the FQIS to the flight crew to identify potential fuel configuration problems, such as low fuel levels in the main tanks, a lateral fuel load imbalance between the two main tanks or a measurable fuel quantity remaining in the center auxiliary tank with the associated fuel pumps turned off.

d. Self-Monitoring. Built-In-Test-Equipment (BITE) features provide fault detection and isolation to individual system component level, including wiring to individual capacitive sensors, thus simplifying on-aircraft maintenance actions. Fault isolation to the individual circuit card of the processor unit is also provided to simplify in-shop maintenance actions. In addition, BITE provides continuous monitoring and annunciation of the system operational status to the flight crew.

e. Fault Tolerance. The dual-redundant hardware configuration of the processor unit and innovative sensor substitution software algorithms, in many cases, permit uninterrupted operation in spite of the presence of a system fault.

Background

Fuel quantity measurement in the fuel tanks of commercial transport aircraft, as well as all other high-performance aircraft, has been based on a capacitance technique employing vertically oriented, cylindrical sensors. These sensors, consisting of inner and outer coaxial electrodes, are mounted in the tanks so that the fuel level in the sensor corresponds to that of the tank itself. Since fuel has a dielectric constant approximately twice that of air, the capacitance of the sensors will roughly double in value when fully immersed in fuel, compared with its value in air. This change in capacitance provides a means of determining the quantity of fuel in the tank.

Since aircraft fuel tanks have an irregular shape, and the airplane operates over a wide range of attitudes, each tank requires a number of sensors in order to measure fuel quantity accurately. To compensate for the tank shape and for the range of attitudes, the sensors in previous systems are electrically wired in parallel and physically profiled, typically by varying the diameter of the inner electrode, so that the total capacitance change per unit of fuel volume is ideally a constant. Determining the optimum profiling for each sensor in order to minimize errors due to tank orientation is a complex task requiring techniques such as linear programming. The result is a compromise for all factors of orientation, tank shape, and fuel volumes, since only one profile can be built into a sensor, thereby limiting the achievable overall system accuracy. Improvement in accuracy can only be obtained with this type of system by adding more sensors.

Fuel quantity information is generally displayed in terms of weight or mass, since the energy content of the fuel load is more a function of mass than of volume. The capacitance type of measurement system is adapted for mass measurement because there is a predictable relationship between density and dielectric constant. However, since aircraft fuels are mixtures of various hydrocarbon compounds, errors of up to 2% can be encountered with the conventional system due only to variations in fuel composition.

System Description

A block diagram of the complete FQIS, including aircraft interfaces, is shown in Figure 1. The system consists of the following components, as shown in Figure 2:

- A set of tank units in each tank to sense the fuel height at selected locations within each tank.
- One compensator unit in each tank to sense the dielectric constant of the fuel.
- One densitometer in each tank to sense the density of the fuel.
- Flight deck fuel quantity displays showing individual tank fuel quantities and total fuel quantity remaining.
- Three load-select indicators located at the fueling station, each having a dual display showing the fuel quantity and the fuel load preselect command for a tank.
- One fueling station load-select control to preselect the desired fuel load for each tank.
- One processor unit to provide all measurement, data processing, data transmission, BITE, and control functions.
- Associated system wiring and connectors both inside the fuel tanks and external to the tanks.

System design is based on digital data processing techniques. It incorporates dual channels of digital data processing in a central processor unit, digital display of fuel quantity at the flight deck and at the fueling station, continuous measurement of the fuel density in each tank, in flight as well as on the ground, and fuel volume measurement through the use of linear capacitance-type sensors in each fuel tank. The following output functions are provided:

- Digital display of fuel quantity in each tank.
- Digital display of total fuel quantity remaining.
- Digital control and display of preselected fuel load in each tank.
- Fueling valve control to automatically terminate fueling of each tank at either the preselected fuel load or at a predetermined "full" volume (a software constant defining the maximum amount of fuel volume allowed for a tank).
- Individual tank and total fuel quantity data outputs to an ARINC 429 Digital Information Transfer System (DITS) bus.
- Individual tank sensor capacitance and density data outputs to the ARINC 429 DITS bus to aid in troubleshooting problems with in-tank sensors and associated wiring.
- System operational status information to an ARINC 429 DITS bus.
- Fuel configuration caution alert information.
- System "Maintenance Required" status information.
- Individual system component fault status information.
- Individual processor unit circuit card fault status information.
- Automatic blanking of fuel quantity displays when system faults are detected.

System Architecture

The FQIS is centered around the processor unit which contains dual processing channels, dual excitation circuits, and dual power supplies. A single set of sensors in each tank provides inputs to the processor unit; the processor unit provides a single set of outputs to the fuel quantity displays located on the flight deck and fueling station and to the ARINC 429 DITS bus. The processor unit also provides outputs that control the fueling valves.

Inputs from the sensors are available to both data processing channels of the processor unit, and both channels are continuously on line, simultaneously performing fuel quantity computations. Only one channel is selected to transmit fuel quantity information to the displays and to the DITS bus. This channel is designated the output-enabled channel and is randomly selected at power-up. Both channels have the ability to control the fueling valves. Where dual valves are used to fuel a particular tank, each channel controls one valve; where a single valve is used, the valve control signals are wired in parallel. Each channel is powered by its own power supply, operating off an independent 28-Vdc airplane power bus.

Each data processing channel of the processor unit has associated with it an excitation circuit, one of which is randomly selected at power-up. This circuit provides an 18.75-kHz sine wave excitation to the tank units and compensators and a 5-Vdc excitation to the load-select control at the fueling station.

Digital System Operation

A. Fuel Quantity Measurement

The processor unit measures the individual capacitance of each tank unit and compensator. These sensors are individually and sequentially excited once per second by the excitation circuit through a multiplexer under processor unit control. A null balance circuit (discussed later) then measures the capacitance of each tank unit. The tank unit capacitance is a minimum when the unit is dry and is increased by immersion in fuel. Since the tank unit is a linear device in this system, the capacitance added by fuel is a linear function of the immersed length. It is also a function of the dielectric constant of the fuel. Thus, the compensator is needed to measure the dielectric constant of the fuel in order to obtain the immersed length for each tank unit, also known as wetted sensing length (WSL). Wetted sensing length is computed using the relation:

$$\mathrm{WSL} = \frac{(\Delta C)(L)}{(K-1)\,Ca}$$

where ΔC is the capacitance increase due to fuel, L is the total length of the tank unit, K is the dielectric constant for the fuel, and Ca is the capacitance of the tank unit in air.

The dielectric constant of the fuel is computed from the measured capacitance of the compensator unit when it is fully immersed in the fuel, using the relation:

$$K = \frac{Cf}{Ca}$$

where K is the dielectric constant of the fuel, Cf is the capacitance of the compensator unit in fuel, and Ca is the capacitance of the compensator unit in air, which is a known constant.

Measurement of both the tank unit and the compensator unit capacitance values is done using the same null balance circuit, which is shown in Figure 3. The circuit is controlled

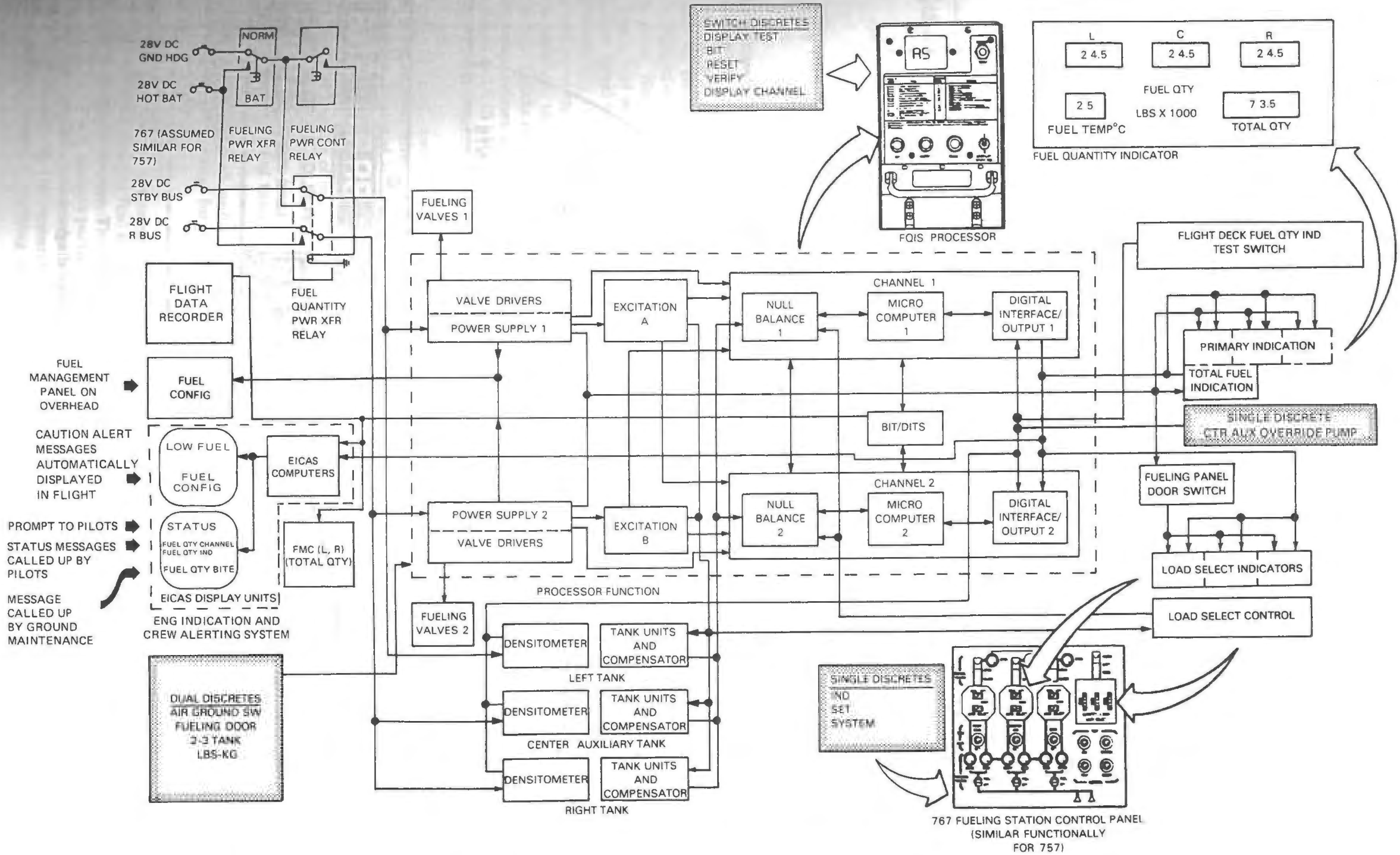


Fig. 1 Fuel quantity indicating system functional block diagram.

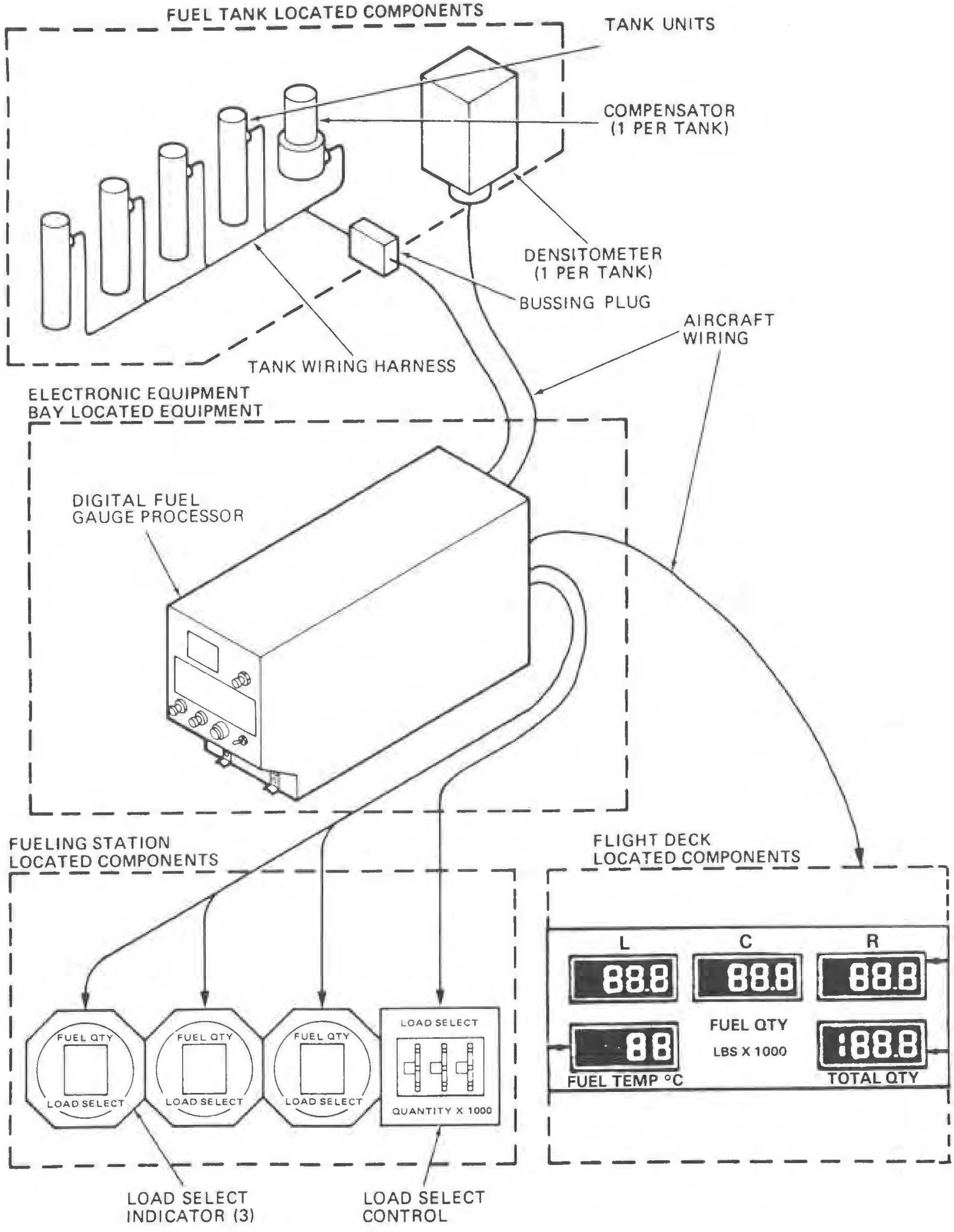


Fig. 2 Fuel quantity indicating system component block diagram.

by the microcomputer which uses a 12-step successive approximation subroutine to determine the logic states of each of the inputs of the 12-bit multiplying digital-to-analog converter (MDAC). The MDAC supplies the excitation of the reference capacitor of the null balance circuit, using the same voltage supplied to the tank units and compensator. The phase of the signal to the reference capacitor is inverted, so that, effectively, the current through the reference capacitor tends to cancel the current through the sensor and, at balance, the input signal to the null balance amplifier is zero.

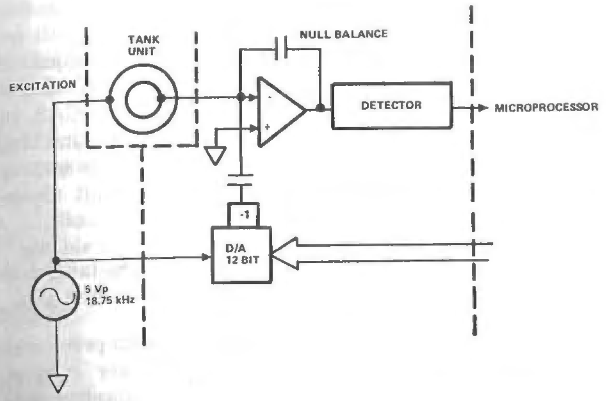


Fig. 3 Capacitive sensor measurement circuit.

The fuel measurement techniques employed by this system, as well as the physical spacing between the tank unit electrodes, provide a high level of immunity against fuel contaminants. Fuel contaminants provide a resistance in parallel with a capacitive sensor and can cause capacitance measurement error. Since the desired signal (capacitance) returned from a fuel sensor is proportional to frequency and the undesired signal (resistance) is independent of frequency, the high frequency (18,750 Hz vs. 400 Hz in conventional systems) enhances the ratio of desired signal to undesired signal. In addition, this system uses a phase-sensitive demodulator that rejects resistive currents returned from tank units and compensators and also acts as a narrow bandpass filter centered at 18.75 kHz, thus allowing for more accurate capacitance measurements.

The capacitive fuel measurements are used by each microcomputer to determine a WSL for each tank unit. The WSL values are converted to measures of partial tank fuel volume using tables stored in the microcomputer memory. These tables replace the physical profiling of tank units used in earlier systems. The use of multiple profile tables for each tank unit provides increased accuracy with fewer tank units. The selection of the table to be used in the partial volume computations is a function of tank volume and whether the aircraft is in flight or on the ground. The partial tank fuel volume computed for all tank units are summed in the microcomputer to obtain a value for the total fuel volume of each tank.

Fuel quantity in weight is computed from the fuel volume measured in each tank and the fuel density measured in the tank by means of a densitometer. The densitometer uses the principle of attenuation of gamma particles emitted by a low-strength Americium 241 source, the attenuation being a function of the density of the material in the path. The number of gamma particles reaching each of two detector tubes in the densitometer is counted in the processor unit. The path of the gamma particles is through a different distance in fuel for each detector tube, thus allowing the density to be computed from the ratio of the number of particles reaching each tube. This ratiometric approach using dual detectors simplifies the densitometer design and enhances accuracy of the density measurement by eliminating factors such as: (1) initial source strength; (2) source decay; (3) variations in detector tube excitation voltage; and (4) temperature effects on detector tubes and electronics. Figure 4 shows the operating concept of the densitometer.

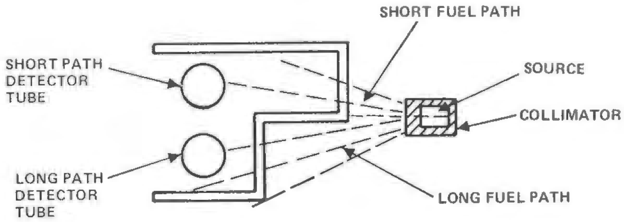


Fig. 4 Densitometer operating concept.

B. Fueling Operations

Two modes of automatic fueling operation are provided by the system: (1) fueling of each tank to a preselected fuel weight, or (2) fueling of each tank to a predetermined "full" volume. In either mode, when the computed fuel load, either weight or volume, for a tank reaches the preset shutoff point, a signal is transmitted to close the appropriate valve (or valves) for that tank.

For volumetric fueling, each tank's "full" volume is a constant that is stored in a program memory. For fueling to a preselected fuel weight, a load-select command is entered manually at the fueling station by means of the load-select control. The load-select control setting is "read" by the processor unit on command of a discrete signal from a switch, one for each tank, located at the fueling station. The reading of the load-select control setting is stored in the microcomputer's nonvolatile memory and is also transmitted to the appropriate load-select indicator, thus providing feedback to the fueling operator to verify the setting.

During fueling, the actual fuel quantity for each tank is continuously compared with the stored values of load-select weight and full-tank volume. If a load-select value has not been preselected via the load-select control, fueling will continue until the full-tank volume is reached. If a value has been preselected which would cause the volume to exceed the full-tank volume, fueling will stop when the full volume for the particular tank is reached.

Delays in the system, including valve closure time and data processing delays, are accounted for by an "anticipation" constant in the software. This constant causes the valve closure signal to be transmitted before the actual quantity reaches the shutoff value.

C. Caution Alert Signals

The FQIS provides two caution alert signals to inform the flight crew of potential fuel configuration problems. This information is displayed to the flight crew via the EICAS (Engine Indicating and Crew Alerting System) displays.

The first of these caution alert signals, labeled 'Low Main Fuel,' is transmitted by the processor unit whenever the fuel quantity measurement in either main tank is less than a fixed minimum level.

The second caution alert signal, labeled 'Fuel Configuration,' is transmitted by the processor unit if any of three conditions exist. The first is the low main fuel condition defined above. Secondly, the Fuel Configuration caution alert will be transmitted whenever the fuel quantities calculated for the left and right main tanks differ by more than a variable limit (i.e., a lateral imbalance exists). This limit varies as a function of total fuel load. The third condition for which the Fuel Configuration caution alert is transmitted is when the fuel quantity measured for the center auxiliary tank exceeds a set limit and both of the center auxiliary tank fuel pumps are off.

D. Built-In-Test Equipment (BITE)

The FQIS provides comprehensive and automatic monitoring, accurate analysis and display of fault information, and a high degree of fault tolerance.

1. Fault Detection. Faults are detected by the results of various tests. These tests are performed in three distinct modes: power-up BITE, continuous BITE, and manually initiated BITE.

Power-up BITE consists of tests which interfere with the normal operation of the fuel quantity system and cannot be performed on a continuous basis during normal system operation.

The tests which are performed on a continuous basis during normal system operation after power-up compose continuous BITE. Most of these tests are periodic, usually performed once per second. These tests include sensor data tests for reasonableness and contamination, and tests of individual processor functions which do not interrupt normal system operations.

Manually initiated BITE consists of those tests which are performed only as a result of a switch action. Switch actions may occur at the flight deck, at the fueling station, or on the front of the processor unit. These tests typically check the operating status of an individual component of the fuel system, such as the flight deck display or the fueling valves.

The test results are stored both in random-access memory (RAM) for current fault status information, and in nonvolatile memory (NVM) for cumulative fault status information.

2. Fault Isolation. There are two types of fault isolation provided by the FQIS: demand and continuous. Demand fault isolation provides a cumulative analysis of the system faults from the last reset of the NVM to aid on-aircraft maintenance. This function examines all fault data stored in the NVM to ascertain which system component is most likely to contain the fault that has been detected. For further detailed shop maintenance activities, processor unit faults are isolated to the individual circuit card contained within the unit.

Continuous fault isolation provides a current analysis of the system's operational status based on the contents of the fault information stored in the RAM. The results of the analysis of this fault information are displayed to the flight crew via the EICAS displays.

3. Fault Annunciation. The FQIS provides two types of fault annunciation. First, the current fault status information from continuous fault isolation is transmitted to the EICAS via the ARINC 429 DITS bus. This information is used by the EICAS to display a status message to the flight crew when an FQIS fault is detected. When the accuracy of the fuel quantity measurement for a tank becomes suspect due to a detected system fault, the fuel quantity displays for that tank and for the total fuel quantity are blanked to provide the flight crew with an indication of the problem.

The second type of annunciation provided is to the maintenance crew. This type of annunciation is provided in two areas. First, if a fault has been detected and that fault information is stored in the NVM, a 'maintenance required' signal is tranmitted by the processor unit to the EICAS and a maintenance message will be displayed on the EICAS. In addition to the EICAS display, the failure information contained in the NVM is analyzed upon demand and presented to the maintenance crew by means of a two-digit alphanumeric display on the front panel of the processor unit.

When all fault information has been retrieved, the fault data can be reset with a switch on the front of the processor unit.

4. Fault Recovery. The redundancy of dual data-processing channels, sensor excitation circuitry and power supplies enables the processor unit to maintain all fuel quantity measurement, calculation, data output, and fueling functions after a single BITE-detected processor unit failure or loss of one aircraft 28-Vdc power bus. The system also has fault recovery features associated with a densitometer fault, a contaminated compensator, and one contaminated tank unit per tank.

On power-up, the selections of the output-enabled processor channel and the active excitation circuit are random. The determination of the output-enabled channel and the active excitation cirucit subsequent to power-up is controlled by the results of BITE. Whenever a channel or excitation is determined to be faulty, that channel or excitation is then deselected and the other channel or excitation is selected. BITE tests are also performed to provide for the selection of the channel and excitation circuit for which the lower fuel quantity is calculated.

Software redundancy provides the capability of system operation after detection of a failed densitometer or a contaminated compensator with minimal accuracy degradation. When a failed densitometer or contaminated compensator is detected, software substitutes data from a "good" sensor in another tank for data from the failed device. The effect of this substitution on system accuracy is determined by the difference in fuel properties between the tank in which the failed device is located and that in which the substituted device is located. Typically, the fuel in the mirror-image main tanks will be virtually identical, resulting in very little error. A larger variance can exist between a main tank and the center auxiliary tank because of differences in fuel consumption from the two tanks, resulting in different fuel mixtures when the tanks are fueled.

Software provides another recovery feature when a single tank unit in a tank is contaminted. Data from the contaminated sensor is replaced by data from either one or two adjacent good sensors, which are analytically selected to minimize system accuracy degradation. This capability for recovering from one contaminated tank unit per tank is possible with the FQIS because the tank units in a tank are excited individually, thus allowing the contaminated unit to be identified.

Digital FQIS Patents

Many of the features described in this paper are included in one or more of the following Honeywell FQIS patents:

1. U.S. patent no. 4,337,638, "Liquid Gaging System Self-Test Circuitry"
2. U.S. patent no. 4,350,039, "Liquid Gaging System Null Balance Circuitry"
3. U.S. patent no. 4,352,159, "Liquid Gaging System Lost Sensor Recovery"
4. U.S. patent no. 4,355,363, "Digital Characterization of Liquid Gaging System Sensors"
5. U.S. patent no. 4,363,239, "Liquid Gaging System Contamination Monitor"
6. U.S. patent no. 4,373,390, "Liquid Gaging System Compatible with Multiple Characterization of Each Sensor"
7. U.S. patent no. 4,388,828, "Liquid Gaging System Calibration"
8. U.S. patent no. 4,451,894, "Liquid Gaging System Multiplexing."

Conclusions

The digital fuel quantity indicating system combines the well-established capacitance fuel measurement technology and state-of-the-art digital data processing and nuclear density measurement techniques, with resulting improvements in system accuracy, fault tolerance, and maintainability. The capability of software to handle tasks previously done by hardware—in this system, specifically the tank unit profiling—is a significant advancement. Software also provides the extensive self-test, caution alert signal, fault isolation and fault recovery capabilities of this system, features not previously provided by more conventional systems.

84-2603

ARCHITECTURAL SOLUTIONS TO SAFETY PROBLEMS OF DIGITAL FLIGHT–CRITICAL SYSTEMS FOR COMMERCIAL TRANSPORTS

Larry James Yount
Staff Engineer
Sperry Corporation, Flight Systems
Phoenix, Arizona 85027
(602) 869-1804

Abstract

This paper discusses the special advantages and problems encountered when critical functions of modern commercial transport aircraft are implemented with software rather than discrete hardware. The dissimilar processing techniques in the Sperry dual-dissimilar SP-300 flight control system of the Boeing 737-300 aircraft serve as a reference point for discussing the various advantages and disadvantages of dissimilar redundancy. The tendency in the industry to include fallback provisions in the form of discrete analog circuitry for critical functions is evidence of the problems and uncertainties that remain regarding critical software implementations. Techniques to overcome generic processor hardware and software faults are discussed. Also discussed are new concepts in processor independent hardware monitors to ensure proper program execution in critical software systems. The emphasis in this paper is on architectural solutions to critical safety validation problems rather than on an exhaustive analysis. Particular architectures designed for fail-op and fail-op-squared, flight-critical systems that are exclusively dependent on software implementations are analyzed.

Introduction

The use of digital processing in implementations of flight-critical system functions has brought software and digital processor hardware under close scrutiny. Both are difficult to analyze due to the almost limitless failure modes and indeterminable effects.

The initial choice of system architecture and the corresponding system fault-analysis technique are crucial to the cost, schedule effectiveness, and safety of a flight control computer (FCC). Architectural short-cuts that compromise computing independence can lead to schedule uncertainties, cost excesses, and other penalties. An architecturally dependent rather than an analysis-dependent system design often presents the least long-term (life-cycle) risk. Such an approach has proved successful with the SP-300 flight control system for the Boeing 737-300 aircraft. The SP-300 employs dissimilar processing. Independence of all redundant elements, both hardware and software, are maximized.

Recent advanced flight-controls development efforts at Sperry concentrated on the theme of architectural solutions in flight-critical applications. Clearly, as digital processor-based systems become increasingly complex, critical exposure times lengthen, and criticality requirements become more stringent, new solutions to the problem of safety verification must be found. As we look ahead, safety-critical, highly complex, full-time systems utilizing artificial intelligence techniques appear inevitable. The threat that heavy reliance on a single type of complex computer might lead to hazard is the stuff science fiction is made of; yet for critical systems this has already become a real concern. Dissimilar processing techniques are both a first step and a common-sense approach to ensure the safety of real-time digital processing systems.

Digital processors are not generally dedicated to a single function. Through multitasking, one computer can appear as a multiplicity of processing entities. If task criticalities include critical as well as noncritical functions, then the effective criticality of all tasks tends to default toward the most critical. Similarly, when multiple aircraft-control functions of different criticality are integrated, the partitioning of hardware fault effects can be both a safety and aircraft function availability issue.

The major subject of this paper is the development of system architectures that are tolerant of generic processing faults. Other subjects covered deal with techniques for both software fault-effects paritioning with a single central processing unit (CPU) and hardware fault-effects partitioning within a processing system.

Dissimilar Processing

The emergence of dissimilar processing was in response to the unique problems introduced by the application of highly complex processor hardware and software to critical-function electronic systems. It is generally agreed that digital processing allows increased sophistication, enhanced flexibility, greater reliability, and improved maintainability. Not surprisingly, however, these advantages have been accompanied by the introduction of new and difficult analysis and verification problems.

The term "dissimilar processing" can be interpreted to include a wide range of redundancy techniques. For this paper, dissimilar processing is defined as follows:

- Software Design Dissimilarity
- Software "Coding" Dissimilarity
- Processor Dissimilarity

Software design dissimilarity includes the software operating system, the software executive, procedures for handling interrupts, techniques for passing data between modules, and software task/module partitioning. The software code implementation is based upon different specifications, different programming teams, and different languages. Processor dissimilarity protects against hardware generic faults and ensures code dissimilarity.

The basic certification requirements for aircraft systems are established by Section 25.1309 (as interpreted by AC 25-1309-1) of the Federal Air Regulations.[1] In earlier critical control systems, designs were verified through accepted techniques of rigorous analysis. When these analysis techniques were applied to digital systems, with their complex software and processor hardware, a few unfortunate factors became apparent. Although the analysis task may have appeared straightforward, the engineers responsible were faced with an almost limitless quantity of unique failure modes. It is necessary to prove that for a one-hour exposure time the probability of a hazardous event resulting from a generic processing fault (software or hardware) is less than 10^{-9}. The implications of this dilemma are obvious. Even when utilizing the most advanced computer-based analysis tools available,[2] the adequacy of the analysis is questionable when confidence levels approaching 100 percent are required, as in critical aircraft function implementations.

To overcome the verification problems introduced by software and processor hardware, as well as for purposes of certification schedule risk reduction, dissimilar processing was adopted for and is fundamental to the SP-300 digital autopilot flight director category IIIa system.[3] Approach, landing, and go-around software was developed for a dissimilar secondary processor as well as for the primary processor. The secondary processor software package remained small, comprising only 10 to 15 percent of the combined primary/secondary processor software package.

Software for the SP-300 primary and secondary processors was developed by two independent software teams. Separation begins with the software specification. Module specifications, software code development, and module test development are independent. The system specification, which includes aircraft control laws and mode transition criteria, was common to both software development efforts. Software development hierarchy and isolation boundaries are shown in Figure 1.

It was found that the secondary processor software development costs were 40 percent less per word of memory than that of the primary processor. This was attributed to the inevitable higher efficiency of the much smaller programming team. The SP-300 development cycle was smoother and more predictable than had been experienced on other design development efforts where similar processing was utilized.

The use of 32-bit floating-point arithmetic in the primary processor and 32-bit fixed-point arithmetic in the secondary processor effectively eliminated computational inaccuracies/differences as a significant factor in determining processor-to-processor comparison monitor threshold tightness limits. The more familiar factor of "sensor skew" proved to be the dominant factor relative to processor computational differences, as expected in any unsynchronized similar processing application.

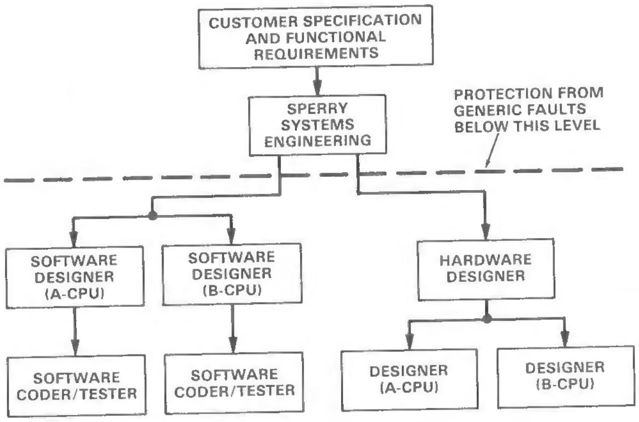

CUSTOMER SPECIFICATION AND FUNCTIONAL REQUIREMENTS
SPERRY SYSTEMS ENGINEERING
PROTECTION FROM GENERIC FAULTS BELOW THIS LEVEL
SOFTWARE DESIGNER (A-CPU)
SOFTWARE DESIGNER (B-CPU)
HARDWARE DESIGNER
SOFTWARE CODER/TESTER
SOFTWARE CODER/TESTER
DESIGNER (A-CPU)
DESIGNER (B-CPU)

Fig. 1 General fault protection resulting from dissimilar software/hardware implementation

Another SP-300 anticipated benefit from dissimilar processing relates to post-certification changes to software. For the SP-300, the possibility that such a change to software will result in a significant undetected coding error is essentially eliminated. This is especially significant because the personnel responsible for such a change may be far less familiar with the software than the personnel associated with the original certification.

A hardware post-certification concern also exists relative to supplier-introduced changes to complex integrated circuits such as microprocessors. Such changes may have been introduced either intentionally or accidentally. The extreme complexity of these devices makes the detection of subtle changes (with a degree of confidence approaching 100 percent) a major new area of concern.

During 737-300 Certification Plan negotiations, agreements were reached with the FAA that give tangible credit for the use of dissimilar processing. Examples of this credit include the following:

- No special analysis is required to assure the absence of support software design errors.
- Coding errors (at the detail level) in CPU-1 and CPU-2 (primary and secondary processors, respectively) causing the same effect do not have to be considered.
- The criticality of all CPU-1 software is classified essential.
- The criticality of CPU-2 software retains the "critical" designation; however, module test coverage analysis is not required as part of the overall test coverage analysis.
- Processor failure modes and effects analysis (FMEA) are not required.

The FAA agreed that if this approach proves successful, more credit could be given in the future. The FAA uses the standards contained in the Radio Technical Commission for Aeronautics document DO-178[4] to evaluate software.

When software development team isolation is mandated, specification of dissimilar microprocessors does not carry a significant software development burden. Rather, dissimilar processor selection is a key element in ensuring the necessary isolation. Dissimilar processor selection significantly reduces the criticality of the associated support software.

Before leaving the subject of dissimilar processing, the unique characteristics of a generic fault in software should be addressed. Software-based redundant systems have the unique characteristic capacity to be precisely identical. Accordingly, the possibility exists that a generic fault in detail program code or processor hardware could cause a unique set of otherwise benign time-dependent events to precipitate precisely the same hazardous response in all redundant systems at precisely the same time. Therefore, it is argued that the unique capacity of software systems to be precisely identical increases the significance of the generic fault problem in such systems.

Generic Fault Tolerant Architecture

An architecture utilizing dissimilar processing techniques has been developed that yields a system capable of tolerating generic digital processing faults as well as random faults. The architecture is a derivative of the dual-dual type. A dual-dual system consists of two independent FCCs, each containing two independent signal processing lanes. Before describing the generic fault-tolerant architecture, the background of the architecture will be developed and the terminology clarified.

A "lane" (Figure 2) includes input signal conversion electronics, one or more digital processors, and output signal conversion electronics. An I/O memory-and-control element serves as the data crossroads for input conversion, processing, and output conversion of information.

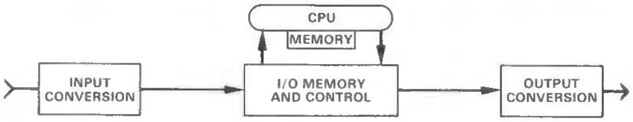


Fig. 2 One processing lane (with one CPU)

Combining two lanes with fault detection via cross-lane comparison of computed outputs (Figure 3) forms the most basic fail-passive architecture.

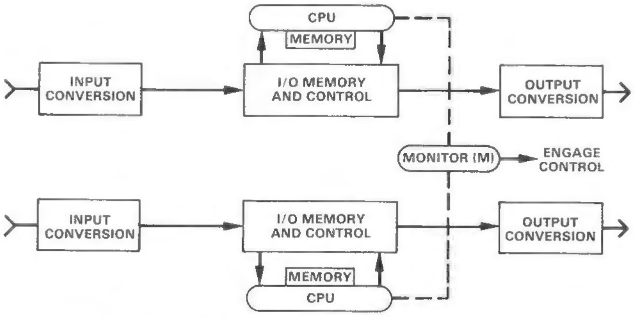


Fig. 3 Dual lane FCC (fail-passive)

Combining two fail-passive systems yields a fail-operational system. Such a system (Figure 4) is composed of four processing lanes grouped in pairs of two (i.e., dual-dual). Each pair is independently fail-passive, and two such pairs are therefore fail-operational.

The fail-passive system of Figure 3 and the fail-operational system of Figure 4 are not protected against generic processing faults. The basic building block of the subject generic fault-tolerant architectures is shown in Figure 5. A single flight control computer (FCC) consists of two signal processing lanes. As shown, the upper lane has CPU-2 as its dedicated processor. In the lower lane CPU-3 is the dedicated processor. CPU-1 has access to data in both lanes. CPU-1 is the primary processor. The computational requirements of CPU-2 and CPU-3 are held to a minimum, and consist only of critical function software.

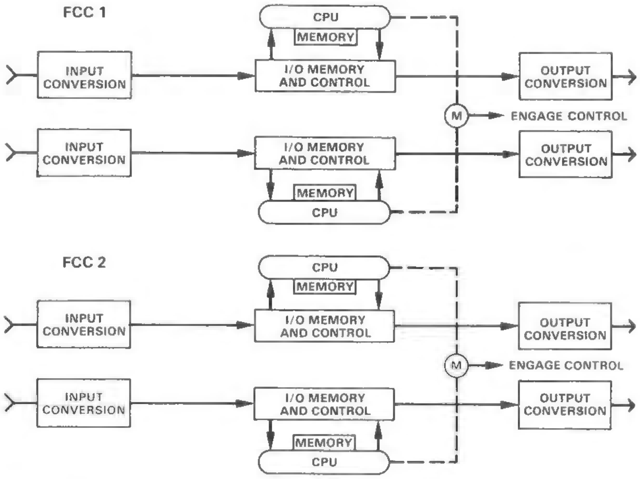


Fig. 4 Dual-dual configuration (fail-operational)

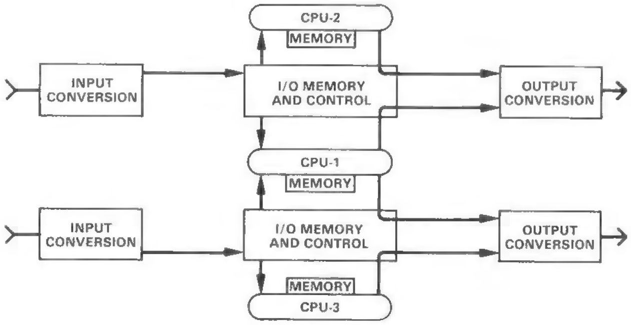


Fig. 5 Single FCC engaged configuration

The fail-operational configuration is shown in Figure 6. In the first FCC, CPU-1 access to the lower lane has been eliminated. In the second FCC, CPU-1 access to the upper lane has been eliminated. In this hardware interlocked configuration, the two lanes of an FCC are fully isolated. The M1 and M2 comparison monitors are monitors between processors associated with opposite lanes. In the first FCC, the M1 monitor determines the validity of CPU-1 outputs and is implemented so that it directly

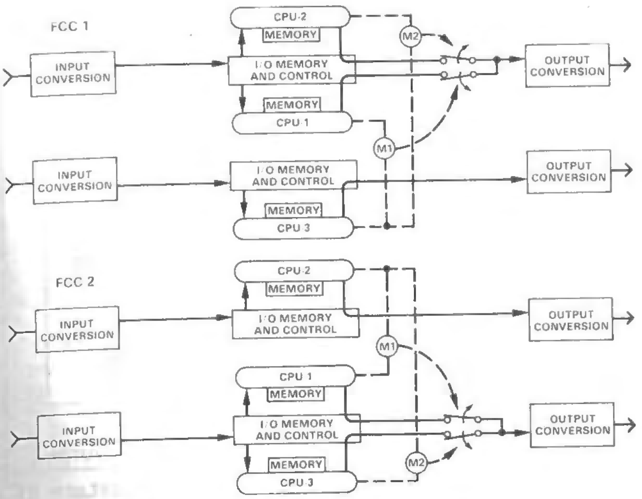


Fig. 6 Dual FCC engagement configuration

controls the ability of CPU-1 to affect external control. Similarly M2 determines the ability of CPU-2 to affect external control. If both M1 and M2 indicate a fault, the entire FCC is judged faulted and is shutdown. The system fail-operational response to either a CPU-1, CPU-2, or CPU-3 generic processing fault is shown in Figures 7, 8, and 9, respectively.

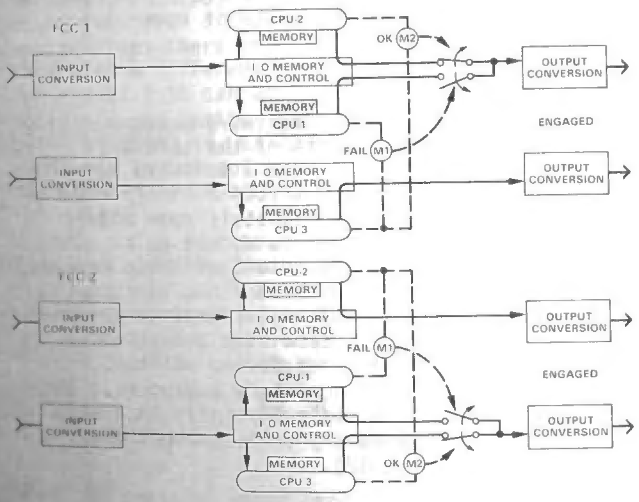


Fig. 7 Response to CPU-1 generic fault

The detection of a CPU-1 generic fault in both FCCs precludes CPU-1 of both FCCs from affecting external control. Note that for this fault case, both FCCs are left fully operational for critical functions despite the fault. A CPU-2 or CPU-3 generic fault results in the fail-passive shutdown of either the first or second FCC, respectively.

Critical function software is implemented in each of the three dissimilar processors. This triples the chance of an error in that portion of software. Loss of fail-operational performance would result in errors of sufficient severity in two of the three processors such that the cross-lane monitor thresholds were exceeded in both FCCs. For an exposure time of 36 seconds* the software error rate for each of the three processors must be held to less than approximately 10^{-3} per hour to ensure that loss of function is extremely improbable. Alternatively, for a similar processing system, the probability that a software error would result in exceedance of aircraft performance safety margins must be less than 10^{-7} per hour for the same exposure time. The 10,000:1 relationship for the two cases is highly significant.

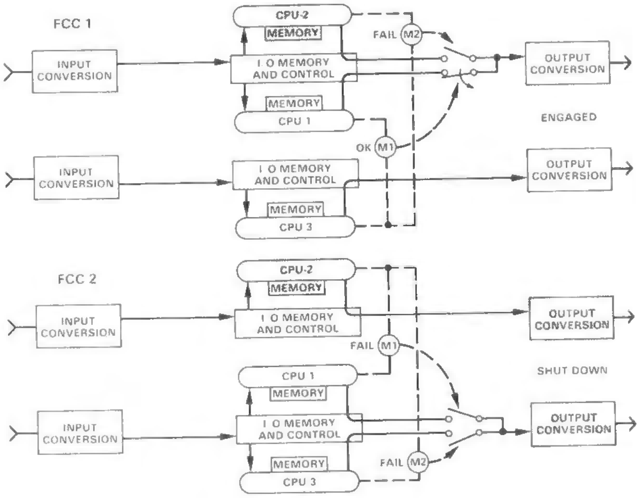


Fig. 8 Response to CPU-2 generic fault

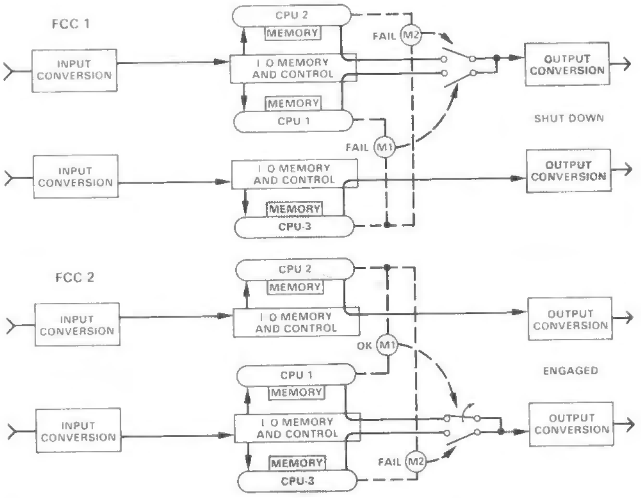


Fig. 9 Response to CPU-3 generic fault

*A typical automatic landing exposure time (1/100 hour).

It is difficult to quantify software error rates in critical systems. It has been suggested in the literature that: "experience tends to show that a reasonable expectation of bugs in a large software developed with maximum care is in the order of 10^{-5} per operating hour."[5] A more complete industry study of attainable software error rates is needed.

Two-Fail-Operational Performance

The first application of the generic fault-tolerant architectural concepts discussed is targeted toward a fail-operational system. However, the fail-op system is actually a subset of a more general two-fail-operational architecture (Figure 10). For this architecture a generic fault in any CPU would result in fail-passive shutdown of one FCC. After shutdown of an FCC, the system degrades to fail-operational.

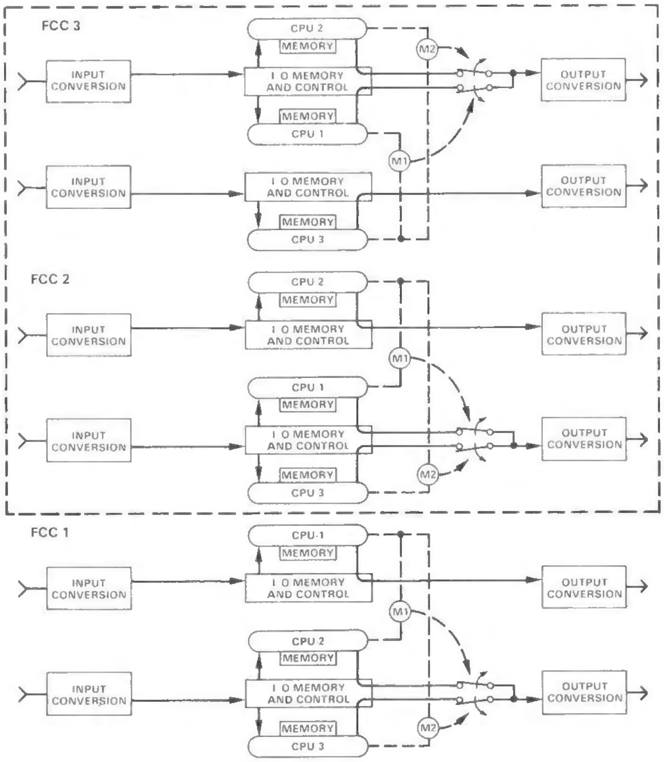


Fig. 10 Two-fail-operational

Software Fault Effects Partitioning

There is a definite reluctance in the industry to accept implementations of critical functions in software. An example of such a critical function is a comparison monitor between two redundant channels of a critical aircraft landing system. Proponents of the software implementation approach argue that distrust of the software approach is unfounded. While it is true that much of the distrust of critical software implementations stems from the much larger experience base with hardware implementations, it must be acknowledged that software techniques introduce a unique set of subtle problems. Among these are:

- Execution - A noncritical software module could erroneously disturb program flow and affect the proper calling sequence or the execution rate of critical modules.
- Data - A noncritical software module could erroneously alter the scratchpad memory locations of a critical module.

These problems result from the sequential nature of software. In the past, techniques intended to protect against them were often under the direct control of the same processor they were designed to protect. This is a questionable practice for critical software functions.

To solve these problems a processor-independent technique has been developed that monitors software flow while dynamically controlling the write-protect configuration of scratchpad memory, as a function of position on the software flow map. Improper program flow is immediately detected. Accordingly the software-monitor device functions as a sophisticated heart-beat monitor. Critical software scratchpad memory areas are protected from erroneous alteration by software of lower criticality.

As shown in Figure 11, the software-monitor circuitry basically consists of the software monitor integrated circuit and a software identity ROM. Each software module writes a binary key-code to the software monitor immediately upon module entry and immediately prior to module exit. The software monitor is the recipient of these key-codes, comparing them against defined legitimate sequences (including branching) stored in the software identity PROM. Each time a legitimate key-code is written to the software monitor, the software monitor transitions to a new state. The monitor is therefore a state machine. Each state uniquely defines the memory write-protection configuration (Figure 12).

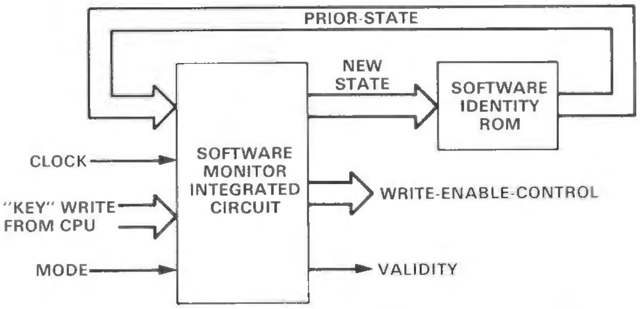


Fig. 11 Software monitor implementation

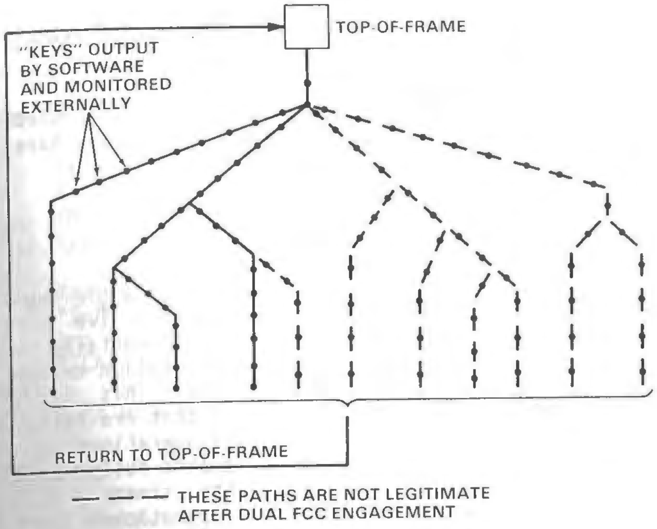

TOP-OF-FRAME
"KEYS" OUTPUT BY SOFTWARE AND MONITORED EXTERNALLY
RETURN TO TOP-OF-FRAME
— — — THESE PATHS ARE NOT LEGITIMATE AFTER DUAL FCC ENGAGEMENT

Fig. 12 Software flow monitoring (single task shown)

As previously mentioned, legitimate sequences of key-codes can include branching. Therefore, multiple legitimate sequences of key-codes could be defined. To manage this situation the software monitor integrated circuit has a mode input. When the mode input is active, a predetermined group of previously legitimate key-code sequences are no longer legitimate. During landing, for example, cruise sequences are excluded from the legitimate set.

So far, the discussion of the software monitor has dealt only with a single software task. In most applications there are several independent tasks, each with a different repetition rate. An individual task can at any time be temporarily suspended and another task may resume at the exact point at which it was last suspended. Each task is monitored by the software monitor for completion at its defined repetition rate.

One remaining execution-related problem that requires additional consideration is a software-induced processor "crash." There is no way to absolutely protect against this. Programming ground rules and sophisticated support/development systems cannot completely solve this problem. There is a means, however, for critical software, when coupled with the software monitor, to tolerate a crash caused by lower criticality software.

When the crash or other program flow disruption occurs, the software monitor will detect a problem almost immediately. All scratchpad memory (state variables) of the critical software remain protected, as previously discussed. The software monitor then forces the system into a recovery sequence (reset). The system is designed so that after first recovery, the system resumes execution of critical software while excluding all lower level software. Worst-case recovery time corresponds to the repetition rate of the task responsible for critical function software.

Hardware Fault Effects Partitioning

Perceived industry trends include the integration of previously federated avionics systems. For example, the Sperry-built digital flight guidance system (DFGS) for the Douglas MD-80 aircraft merges autopilot, flight director, altitude warning, yaw damper, autothrottle, thrust rating, and Mach trim. Problems arising from merging many functions into the same digital processing system include the following:

- Single faults can result in the loss of multiple aircraft systems.
- Where the avionics systems to be merged are of different criticality levels, the analysis requirements for the composite system tend to be elevated to that of the highest criticality individual system. (The difference in failure probability requirements between a "critical" system and an "essential" system is typically four orders of magnitude or 10,000:1.)

An approach taken by Sperry, which has been applied in the SP-300 flight control system for the Boeing 737-300 aircraft, relies on techniques to partition the classes of faults that could affect more than one I/O conversion device or more than one processing element (Figure 13). To standardize the interface of all I/O conversion devices, an interface control (IC) integrated circuit was developed. The interface of any I/O conversion device to the central I/O controller is entirely contained on the IC associated with that I/O conversion device. All control and communication with the central I/O controller is through the IC. The interface is defined so that no faults or design errors of an I/O device can affect another I/O device. Accordingly, the failure modes of a low-criticality I/O device cannot affect a high-criticality I/O device.

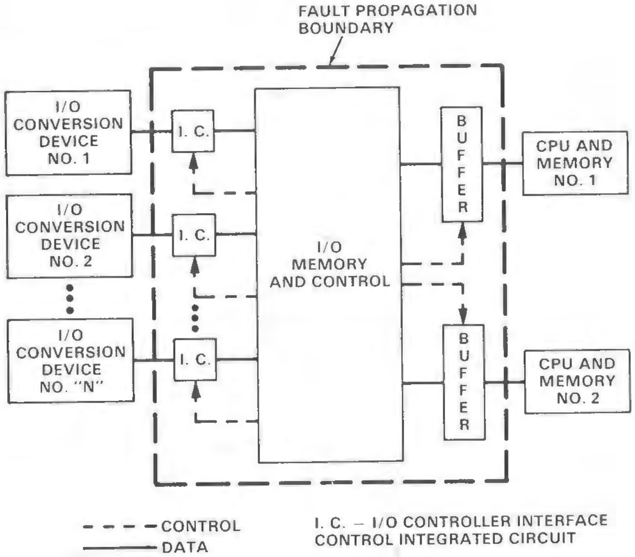

FAULT PROPAGATION BOUNDARY
I/O CONVERSION DEVICE NO. 1
I/O CONVERSION DEVICE NO. 2
I/O CONVERSION DEVICE NO. "N"
I. C.
I. C.
I. C.
I/O MEMORY AND CONTROL
BUFFER
BUFFER
CPU AND MEMORY NO. 1
CPU AND MEMORY NO. 2
- - - - CONTROL
——— DATA
I. C. – I/O CONTROLLER INTERFACE CONTROL INTEGRATED CIRCUIT

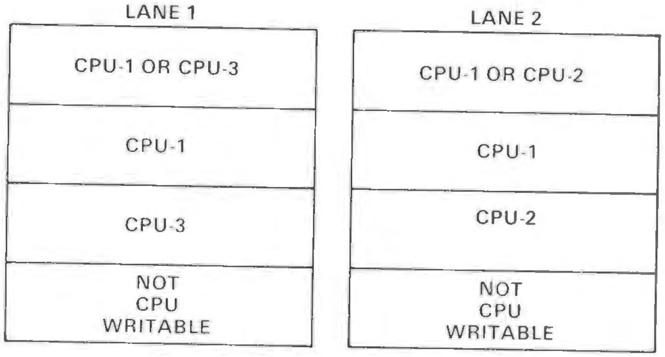
Fig. 13 Hardware fault effects partitioning

The access of each processor to I/O data via transfers to or from the central I/O controller is maximized, yet each processor is protected from contamination by the other. The following techniques have been employed:

- The interface (buffer) between a processor and the central I/O controller is under the control of the central I/O controller. A processor must request and be granted access.

- The time interval allotted to each processor for access to I/O memory is bounded by a dedicated hardware limiter. If the allotted time interval is exceeded, the processor is declared faulted and prohibited further data access to the central I/O controller.

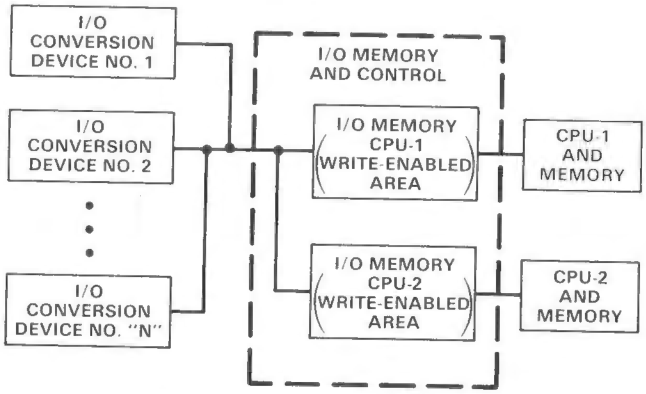
- Any processor can read data from anywhere in I/O memory. However, the ability to write to areas of memory is rigidly partitioned so that one processor cannot alter the data of another (Figure 14).



LANE 1	LANE 2
CPU-1 OR CPU-3	CPU-1 OR CPU-2
CPU-1	CPU-1
CPU-3	CPU-2
NOT CPU WRITABLE	NOT CPU WRITABLE

Fig. 14 I/O memory write-enable partitioning

- Input data is simultaneously written into multiple memory areas by the I/O controller. Each is dedicated to a different processor (Figure 15).



I/O CONVERSION DEVICE NO. 1
I/O CONVERSION DEVICE NO. 2
I/O CONVERSION DEVICE NO. "N"
I/O MEMORY AND CONTROL
I/O MEMORY CPU-1 (WRITE-ENABLED AREA)
I/O MEMORY CPU-2 (WRITE-ENABLED AREA)
CPU-1 AND MEMORY
CPU-2 AND MEMORY

Fig. 15 I/O memory write-enable partitioning

- The net amount of central I/O controller data access time cumulatively available to all processors is governed. Between the beginning and end of each complete I/O controller scan of all I/O device requests, the amount of time allowed for processor data access is bounded by a hardware governor. This guarantees the minimum data transfer rate necessary for all I/O devices, regardless of the number of processors, their short-term I/O data demands, or their fault status. Exceedance of this limit is not judged as a fault; rather, further processor I/O data access is suspended until all I/O device data access requests have been serviced.

Conclusion

Dissimilar software implementation of a single set of aircraft-control algorithms is an effective technique for protecting against the potentially hazardous effects of software implementation errors and generic processor hardware faults. This technique can be applied to systems that are fail-passive, fail-operational, two-fail-operational, and beyond. The fail-op system remains fully operational, even with a loss of the primary processor in both FCCs. The fail-operational architecture is actually a subset of a two-fail-operational general architecture.

The potential for an undetected generic fault in a digital processing system is of elevated significance because such systems have the unique capacity to be precisely identical.

The need for software fault-effects partitioning within a single CPU has resulted in the development of a new isolation and monitoring technique. It can be assured that critical software (e.g. augmentation) can maintain function in the presence of a fault in lower criticality software of an arbitrary type.

Techniques for containing the effects of hardware faults were developed and applied on the SP-300. These techniques preclude a fault in one I/O conversion device from contaminating another I/O conversion device. Similarly, techniques were developed that allow multiple processors to access data from the same I/O conversion devices (via an I/O controller) without the possibility of a random fault or generic hardware/software fault in any processor contaminating another processor.

Finally, the development efforts described in this paper have demonstrated the ability to rely far more heavily on top-down architectural solutions to safety problems. The inevitable trend is toward more complex safety problems as technological advancements precipitate a proliferation of "critical" digital processing implementations for aviation as well as other industries.

References

1. Federal Aviation Administration, "System Design Analysis," Advisory Circular No. 25.1309-1, September 7, 1982.

2. L. J. Yount, Sperry Corporation, Flight Systems, Phoenix, Arizona and G. W. Winkler, Sperry Univac Defense Systems Division, St. Paul, Minnesota, "Digital Computer Self-Test

Confidence Level Validation Methodology," IEEE Catalog No. 79CH1518-0, Library of Congress Catalog Card No. 79-91191.

3. John F. Williams, Larry J. Yount, and James B. Flannigan, "Advanced Autopilot-Flight Director System Computer Architecture for Boeing 737-300 Aircraft," Sperry Corporation, Flight Systems, Phoenix, Arizona. Fifth Digital Avionics Systems Conference, November 1983.

4. Radio Technical Commission for Aeronautics, "Software Considerations in Airborne Systems and Equipment Certification," Document No. RTCA/DO-178, November 1982.

5. RTCA Paper No. 226-83/SC152-13, Paragraph 7.2.

About the Author

LARRY JAMES YOUNT is a Staff Engineer with Sperry Corporation, Flight Systems in Phoenix, Arizona, where he is employed in the Commercial Flight Controls Department. During his employment at Sperry, he has been active in digital flight control computer design. He had systems redundancy management design responsibility for the Sperry SP-300 dual-dissimilar flight control system developed for the Boeing 737-300 aircraft. At present, he is involved in the development of advanced redundancy management concepts for critical aircraft systems.

He earned B.S. and M.S. degrees in Electrical Engineering at the University of Louisville in 1973 and 1974, respectively.

Before joining Sperry in 1976, he was employed at Martin Marietta Corporation in Orlando, Florida.

84-2604

APPLICATIONS OF AIR-GROUND-AIR DIGITAL COMMUNICATION SYSTEM ACARS/AIRCOM

C. Anthony Bennett

Senior Member Technical Staff
Teledyne Controls
Los Angeles

Abstract

In the United States there is currently operational an air-to-ground, VHF, digital data link that is providing commercial airlines with a means of efficient and reliable communications. Though the original development of the system was directed at providing a low volume of short messages (aircraft Out-Off-On-In times) provisions were made for expanded message quantity and length. This paper addresses these new applications of the originally developed air-ground data link.

History

After more than 25 years of study and specification writing by the Airline Electronics Engineering Committee (AEEC) a characteristic was developed in 1975 which resulted in production hardware being initially delivered in 1976. There are over 1500 aircraft now equipped, communicating with nearly 200 ground sites, with virtually all of the operation now concentrated in the United States. However, this geographic coverage is rapidly changing, with four stations now operational in Australia. By the end of 1985 there will be six stations in Europe and 23 in Australia, Indonesia and Southern Asia. Refer to the figure illustrating RF coverage.

Operational Characteristics

The system operation is best described as a two-way, simplex, digital data link operating over the standard VHF frequency band assigned to commercial airlines for company communications. Some of the system characteristics are summarized below:

- Listen to RF path for an empty channel before transmitting.
- A message consists of up to is 256 eight bit Bytes which include bit and character sync, identification of sender, provisions to identify address of recipient, other header information, and a computed error check polynomial.
- Receiving unit recomputes error check polynomial to verify integrity of message and sends an acknowledgement to sender
- There are provisions for multi-block transmission for messages longer than 210-220 ASCII characters (or 8-bit Bytes)
- The current data transfer rate is 2400 bits per second.

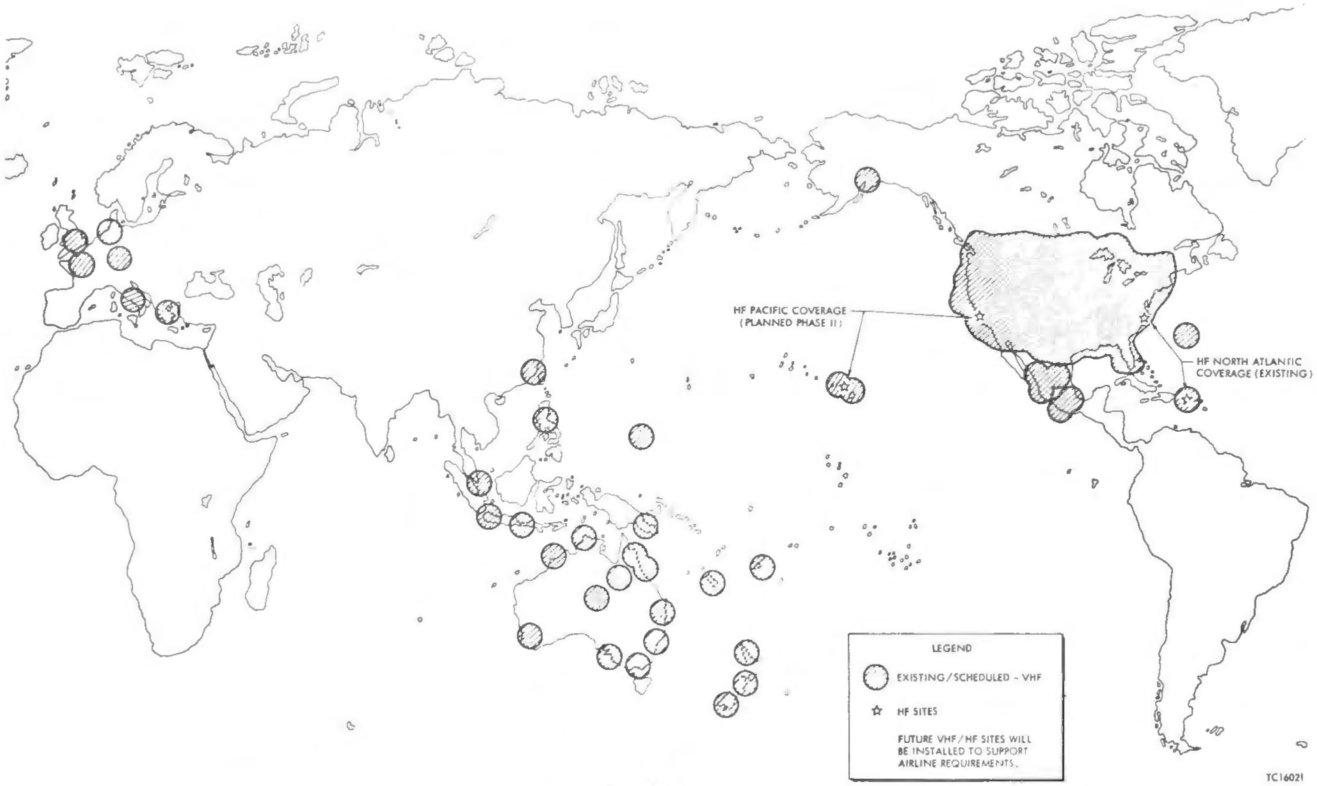

HF PACIFIC COVERAGE
(PLANNED PHASE II)
HF NORTH ATLANTIC
COVERAGE (EXISTING)
LEGEND
○ EXISTING/SCHEDULED - VHF
☆ HF SITES
FUTURE VHF/HF SITES WILL BE INSTALLED TO SUPPORT AIRLINE REQUIREMENTS.
TC16021

Figure 1. R F Coverage

Equipment Required

Airborne

The basic complement of equipment consists of a Control Unit mounted on the flight deck and a Management Unit located in the electronics bay.

The Management Unit interfaces with a standard radio transciever which is either dedicated for data link operation or shared with normal company voice communications. Other optional interfaces of the Management Unit, depending upon the application, include the following:

- Flight Data Acquisition Unit
- Navigation System (INS or Omega)
- Radar Display
- Flight Management Computer

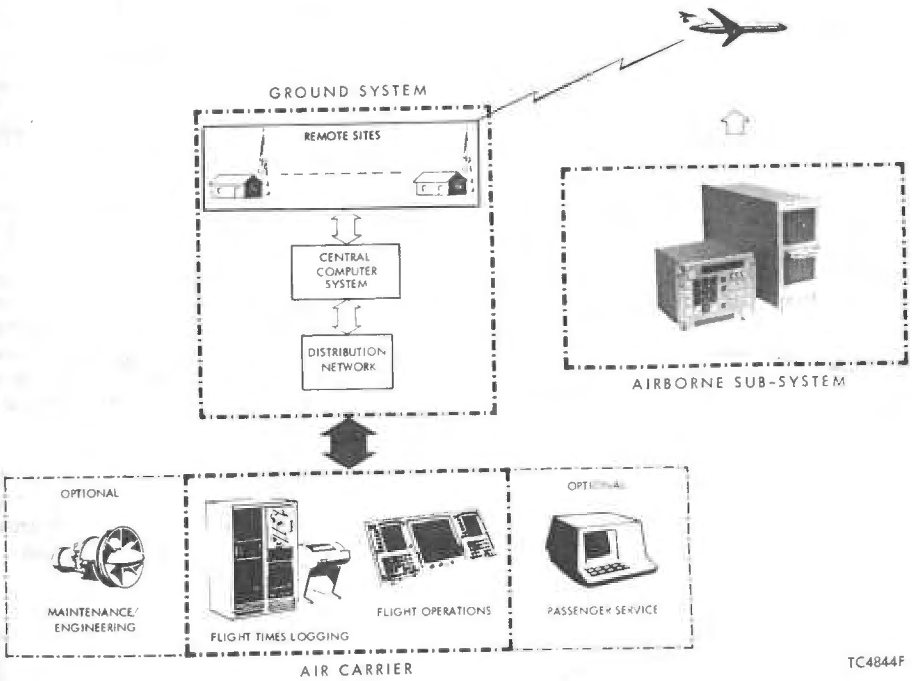
In addition to these interfaces to existing subsystems, the Management Unit includes interface circuitry to permit addition of the following optional subsystems:

- Cockpit Printer
- Passenger service terminal which may include a full keyboard, CRT display, printer, with an interface to the existing passenger entertainment system

Ground

A ground network has been developed by Aeronautical Radio, Inc. (ARINC) in the United States which consists of nearly 200 remote sites under control of Chicago central computer. Messages received at a remote site are forwarded to Chicago for processing and formatting and then distributed to the appropriate airline terminal via another ARINC message switching system. ARINC has identified this data link system as ACARS (ARINC Communications and Reporting System).

A similar and compatible ground network has also been developed by the Sociéte Internationale de Télécommunications Aéronautiques (SITA) for operation outside of the United States. SITA has identified their services as AIRCOM (SITA's digital air-ground communications service). The four remote sites in Australia and one in Amsterdam are part of the SITA network and 25 more are scheduled for operation in 1985.



MESSAGE TYPES

- ETA/DELAY INFORMATION
- REQUEST FOR VOICE CONTACT (UP OR DOWN LINK)
- ENGINE DATA AT TAKE-OFF/STABILIZED CRUISE
- AIRCRAFT LRU STATUS
- AUTOMATIC OUT-OFF-ON-IN TIMES
- WEATHER/POSITION REPORTS FROM AIRCRAFT
- WEATHER UPDATES TO AIRCRAFT
- FLIGHT MANAGEMENT COMPUTER UP-DATE
- REQUEST FOR MAINTENANCE AT DESTINATION
- PASSENGER SERVICE DATE

Figure 2. ACARS/AIRCOM Overview

Current Applications

This bi-directional digital communication link between the aircraft and the ground is being utilized for data exchange of various types including such items as:

- Documentary data manually entered during initialization routine such as; gross weight, passengers on board, destination/departure station, take-off thrust, crew pay no., flt no.
- Automatic collection of Out-Off-On-In times
- Manual entry of ETA/Delay information
- Downlink request for a voice contact followed by an automatic switching to the voice frequency when the ground station has the party on the line
- Uplink request for a voice contact on a designated/displayed frequency and automatic switching to the voice channel
- Manual entry of engine data during stablized cruise (some aircraft configurations will permit this to be automatic)
- In certain aircraft configurations engine data (exceedances, EGT divergence, take-off data) can be automatically collected by the aircraft subsystems or interrogated by ground personnel
- With an on-board printer implemented, flight operations data (Load Information, Flt Plan Update, Gate Assignment) can be sent uplink to the flight deck
- With an appropriately configured Flight Management Computer (FMC) and interfaces, Flight Plan updates can be automatically sent to the FMC. Also, other changes to the FMC data base which occur on a 28-day cycle can be uplinked as required
- Requests for maintenance actions at the destination station can be manually entered and sent prior to arrival
- In certain aircraft configurations the aircraft's LRU fault status can be downlinked
- Both en route and destination station weather information can be sent to the flight deck printer
- A passenger service terminal for airborne reservations (airline, hotel or car) or other passenger needs could be installed in the cabin area
- Uplink of passenger complement, seat assignment, and special considerations, can be provided

As of mid-1984 the operational aircraft now utilizing these message types account for a present day message volume of 1,600,000 per month in the ARINC network and this is estimated to be increasing at a rate of about 50,000 per month.

Future Applications/Advances

The utilization of this data link system is continuing to expand with the specific applications only limited by the ingenuity of the various users. The new avionics being supplied provide the user with a considerable amount of computer memory (128k Bytes), all under the software control of a powerful microprocessor, such as a Motorola MC 68000. With the basic operational program using less than 24k Bytes the remaining 100k Bytes could be utilized for protected message storage in EEPROM until an aircraft is able to offload the messages to the ground stations. This permits aircraft operation outside of the VHF ground network without loss of collected data.

Some of the new and planned applications are summarized below:

- Collection of weather data both during Take-Off/Climb and Cruise provides vertical profile and en route weather data automatically for use in ground computers to optimize weather forecasts. Three airlines have implemented this capability.
- The addition of a complementary HF link, which is currently in-flight test would permit communciations in the areas where VHF ground stations are not available. It may be possible in the future to provide automatic position reports for flight following and thus justify a more optimum metering and spacing of aircraft on high density routes such as the North Atlantic.
- A satellite link is also being studied. The trade-offs in this study address the frequencies available which permit simplified antenna design but have restrictions on message type vs the available low cost frequencies that require complicated (steerable in some cases) antenna design. The power requirements and timing criteria are also being reviewed.
- In the area of passenger service terminals there are numerous possible terminal types that are available which will provide services in the following areas:
  - Crew payroll numbers
  - Stores used
    - Beverages
    - Meals
    - Headsets
  - Prompting Routines
    - Aircraft characteristics
    - Preflight Announcements
    - Different languages
  - Passenger Advisories
    - Connecting flights
    - Arrival facilities
    - Customs requirements
    - Currency
  - Crew Information
    - Flight time
    - Next assignment
    - Accommodations
  - Airline Report
    - Stores required
    - Maintenance required

- Airline owned and operated ground networks which are integrated with their present communciations network for ticketing and flight operations are being seriously studied by some airlines.

- A transmitter/receiver which is optimized for digital data transfer instead of voice communications can be foreseen.

- The use of bit oriented data transfer instead of ASCII can improve message efficiency and most likely will occur when the ground distribution networks can support this capability.

- The possibility of supporting inquiry/response type traffic on this RF link is being evaluated as to need and impact on network.

- A distributed ground computer system with smarter remote sites is being developed to replace the current central processor system. This will reduce ground communications costs.

Summary

It has been proven that this system reduces flight deck work load while providing an even greater amount of data to the ground. Additional future applications are only limited by the ingenuity of the operating airline.

84-2605

AN OVERVIEW OF THE DIGITAL AVIONICS ASSESSMENT ACTIVITIES BEING CONDUCTED BY THE FEDERAL AVIATION ADMINISTRATION AT NASA-AMES RESEARCH CENTER

William Larsen
FAA/AMES Development and Logistics Field Office
AMES Research Center
Moffett Field, California 94035

Donald Eldredge
Ellis Hitt
Battelle Columbus Laboratories
Columbus, Ohio 43201-2639

Dennis Mulcare
William Ness
Lockheed-Georgia Company
Marietta, Georgia 30063

Abstract

The Federal Aviation Administration is currently conducting a research program which is intended to provide data and information which will aid in the functional assessment, by certification engineers, of new and retrofit aircraft which use digital systems for avionics and flight control functions. The FAA's ability to provide the needed data and technical information from the ongoing research program, in a timely manner, is critical to the total certification process. This requirement has been the basis for the development of a program plan designed to meet the research objectives which will support the certification and operational needs for current and future aircraft which incorporate advanced integrated flight control and avionics systems.

Two documents have been pivotal in establishing airworthiness compliance for these digital systems: the FAA Advisory Circular AC25.1309-1 and AC20-115. The former addresses analytical assessment of systems according to functional criticality; the latter focuses on the orderly development (traceability), management, and certification of flight software. However, since both of these documents are subject to interpretation during their application, it has been important to establish the minimum contribution to the certification process required for their practical application. To assist in this task, the Digital Systems Technology Program has been developed and is providing specific data and information for the FAA in support of the long-term needs in the development of airworthiness and operational criteria. These information and data, along with other certification related data packages, will be documented as Data Bases and "Lessons-Learned" in the FAA's Proposed Handbook Volume II - Validation of Digital Systems in Avionics and Flight Control Applications.

INTRODUCTION

Aircraft development in the past has been influenced by the desire for improved performance usually through more efficient aerodynamic designs or propulsion systems. However, one of the most noteworthy advances has been the development of the electronic systems and devices which are utilized in the application of active controls and other advanced flight control and avionic systems and concepts. Many of the advanced concepts offer the potential of improved aircraft performance through increased energy efficiency. Furthermore, it appears that active controls and other advanced flight control and avionics systems, implemented digitally, will significantly influence aircraft technology, and therefore, the Federal Aviation Administration (FAA) must examine the impact of these advances on airworthiness criteria and certification procedures. The FAA-Digital Systems Technology Program was established in order to support these responsibilities and aid in implementing activities in the research area and to provide for initiation of new (or continuation of ongoing) FAA projects related to software based digital systems, validation and verification, failure modes and effects reliability, active controls technology and aircrew/aircraft interface issues. Emphasis is directed toward those activities that have a flight safety impact and the potential of aiding and supporting the certification process through the dissemination of data and information and acquisition of pertinent criteria and procedures.

BACKGROUND

During the past several years the aviation community has witnessed an ever increasing pace to introduce advanced technologies, new aircraft design concepts, and sophisticated high integrity integrated electronic systems. These advanced concepts have impacted many technology areas that are pertinent to the digital systems program; namely, "flight-critical"and "flight-essential" electronic systems which include the following types of software-based digital systems: stability and control augmentation systems, active control systems, advanced displays, the aircrew/aircraft interface, and aircraft handling qualities and flight characteristics. The FAA is confronted with the task of reviewing, revising, and updating its airworthiness assessment criteria, certification procedures, maintenance-inspection requirements, operational considerations, etc., in order to assure a minimum level of safety for aircraft utilizing new design concepts and advanced systems technology; and as such must revise and prepare advanced/updated/new regulatory and guidance material which reflects the impact of these technologies. This Digital Systems Technology Program is the continuation of efforts initiated in 1975 to obtain data and information to aid and support the airworthiness processes cited above.

OBJECTIVES

The overall objectives of the aircraft digital Systems Assurance Assessment technology program are:

(a) To conduct studies, investigations and analyses that will provide data and information to support certification and regulatory activities pertaining to implementation of software-based digital systems.
(b) To coordinate and disseminate materials and findings within the FAA and industry as appropriate.
(c) To be responsive to specific aircraft safety needs.
(d) To establish and maintain expertise in aircraft digital systems assurance assessment technology.

Therefore, the general focus of the Aircraft Digital System Assurance Assessment Technology program is the impact of advanced and new technologies on the aircrew and aircraft. Therefore, the program has been established to focus on efforts that will provide data and information to aid and support aircraft airworthiness assurance techniques, certification procedures, and other pertinent regulatory and safety issues applicable to the operation of an aircraft and its systems by the crew in the National Airspace System (NAS).

Historical Overview

Historically, civil aircraft have incorporated independent hardware (black boxes) for each system implemented and installed in the aircraft; and the FAA's present standards address certification procedures and criteria from the concept of separate engineering disciplines. It is clear that derivative and new generation aircraft which incorporate integrated flight control and avionics systems are dependent, in a complex manner, on the aircrew/aircraft interaction as a total integrated system. A concentrated effort is needed to research this technology area, with emphasis on the total integrated system, in order to acquire and disseminate important data and information within the FAA to assure that certification criteria and procedures are valid and current with advanced technology. Many of the new integrated systems have already been implemented on new derivative aircraft such as the DC-9-80 and the L-1011-500, are utilized on the current generation of new transport aircraft (B767/757) and will be used to an even greater extent on newer aircraft (A310/A320) to improve efficiency and performance. These advanced systems also are being used, to an increasing degree, on other classes of new aircraft such as commuters, general aviation and rotocraft.

The introduction of software based digital flight control and avionics technology in the current generation of aircraft has presented a problem for traditional certification techniques, especially with the increasingly wide-spread implementation of integrated systems which use bus architectures for intersystems and intrasystems communication of data and information. Within this advancing technology, many of the current certification methods may still be applicable, however, new test techniques and assurance assessment methodologies must be used to evaluate and analyze the operation and reliability issues associated with the implementation of software programs which include executive, operational, and interface routines.

The extreme flexibility afforded by these software based digital information transfer systems presents significant problems in assuring that the software structure including compilers, higher-order languages, and architectural design is not adversely affected by changes in the "firmware" or software during the implementation or maintenance life of the systems. It is possible, with these new systems, to make changes to the systems structure such that the basic certification criteria are altered and the flight control system and aircraft safety may be subject to errors or initiation of unintended functions which result in failures. Therefore, it is necessary to establish testing, assurance assessment, and configuration management technology practices to insure that the systems and integration of systems are not vulnerable to errors in design or implementation, by providing regulation and guidance material to design and maintenance engineers which will insure that systems retain the degree of reliability and operational integrity established during the initial certification activity.

In order to insure this integrity, it is necessary (a) to maintain and update the proper data bases and information, related to this emerging technology and (b) provide these data and information in a timely manner to the airworthiness engineers and certification specialists in order to insure that the current and next generation aircraft are the safest ever.

Crucial Technology/Major Concerns for the Digital System Technical Program

The three driving issues that are of particular concern in the establishment of an assurance assessment methodology/technology are:

o Design Verification
o System Performance/Robustness
o Validation Technology.

These areas of concern are important in that they are the major issues faced by the Systems evaluator in his task of establishing the airworthiness of "flight essential/flight critical" software-based digital flight control and avionics systems - especially as the industry starts to implement "full-time/full-authority" systems which are critical for the continued safety-of-flight.

In order to properly assure that these "new" systems are indeed airworthy, the system designer, the system integrator and the airworthiness specialist must have sufficient knowledge of the system as it develops, and the System Design Verification must be planned such that verification is performed during the early stages of development and that the verification process lays the groundwork for validation of the system as well. In addition, explicit checks for robust system performance are needed, during the early stages of verification/validation, in order to insure that marginal or slightly out-of-bounds operational conditions will not overtax system capability. Furthermore, often advanced validation methodologies/technologies need to be perfected and calibrated, and the results of their applications recorded in data bases and other information sources in order for the systems evaluator to be able to judge their impact in establishing the airworthiness of the system designed and evaluated using these new tools and techniques.

In order to develop a proper Design Verification methodology the system designer/integrator needs to assure that the design model is fully analyzed for consistency, completeness and reachability in order to establish the level of compliance with the stated design and performance requirements. Similarly, the airworthiness specialist must have the tools, data bases and information necessary to evaluate the effectiveness of the design verification effort and its contribution to the airworthiness of the system.

System Performance/Robustness must be similarly understood by the system designer, the system integrator and the airworthiness specialist. The testing program established to demonstrate compliance with the system specification must demonstrate that the evolving system has the capacity to tolerate operating conditions somewhat outside the design specification (e.g., the capacity of an autoland (digital flight control system) to perform an automatic landing despite a severe wind shear). Testing for robustness involves implementing a program which contains combinations of operational environment extremes which stress system performance beyond that strictly required. The rationale for this type of testing being that unforeseen and potentially hazardous circumstances may occur in the life cycle of the system and that these circumstances can be duplicated during validation/verification testing. Once again, the airworthiness specialists must have access to the necessary data bases and information that will allow them to assess the completeness of the System Performance Testing/Evaluation Program.

The establishment and understanding of the tools, techniques and methodologies which constitute the selected Validation Technology for a given system requires that the contribution of each and its impact on the conclusiveness of system validation be well documented in data bases and other information sources; and that these data bases and information sources are available to the airworthiness specialists. The implementation of a validation technology is dependent on the proper levels of testing to demonstrate compliance with the design specifications and the intended use of the system. However, since it is not possible to exhaustively test all cases/conditions the system designer/integrator must evolve a testing strategy based on reduced complexity and fewer test cases which requires that multilevel testing using real-time test execution monitors and hardware/software instrumentation at the lower levels be formulated and proved to establish the required level of assurance assessment.

In order to properly assess the applied validation methodology, the airworthiness specialist must: (a) understand the complexities of test case design and instrumentation, and must be able to interpret the observable data for the specific system under test; and (b) must have the proper data bases and information available to assist in the understanding and interpretation of the data submitted for certification.

In summary, in order to properly evaluate the emerging "flight essential/flight critical" digital system the airworthiness specialists need access to the proper data bases and information to be able to assess the airworthiness of these new and innovative software-based digital systems that are now entering the field of commercial aviation. To this end, the existing FAA Digital Systems Technology Program has been providing studies, assessment workshops and other activities since its inception in 1975; all of which have been directed toward being responsive to the needs of airworthiness specialists who have the responsibility of evaluating/approving the in-service implementation of digital systems technology in commercial aviation.

The Digital Systems Technology Program, which started in 1975, has been a joint FAA/NASA program since its inception. Figure 1 presents an overview of the program activities that have been completed and the projected activities through 1986 and beyond. From this figure, it can be seen that significant contributions have been made to the establishment of the data and information necessary to understand the role of Validation Technology and Testing in the validation/verification of digital systems.

Figure 2 shows the major products/outputs of the ongoing program. From this figure, it can be seen that this program has benefited from cooperative funding/direction. Without this cooperative effort the program could not have succeeded and would not be able to provide the necessary data and information necessary for the establishment of required/data bases and information sources.

Figure 3 presents several examples of the proposed work under the Digital Systems Technology Program along with the associated payoffs and problems that the work addresses with respect to emerging airworthiness issues. It is these payoffs and problems that can be expected to increase the complexity of assessing the airworthiness of the next generation of aircraft. System designers/integrators can be expected to take advantage of these advances in digital technology and apply these advances in new, unique and innovative ways for the development and implementation of "full-time/full authority" flight critical systems in order to increase the capabilities and decrease the cost of the next generation commercial transport aircraft.

Through the Digital Systems Technology Program, these new tools, techniques and methodologies can provide the necessary data bases and information to the airworthiness specialists to assist in keeping pace with the evolving technology. The primary output of all these programs will be reports, data bases, and other information sources which will be available to the airworthiness specialists to assist them in performing their role of assessing the airworthiness of the new systems.

CONCLUSIONS

The establishment and continuation of the Digital Systems Technology Program along with the Government/Industry Committees (RTCA/SAE) activities, promises to deliver an assurance assessment/guidance data base which will be of great value in assessing the airworthiness of current and next generation civil aricraft through the integrated application of state-of-the-art tools, techniques and methodologies to evaluate the integrity, reliability and capability of the on-board software based digital systems. The application of these data bases and "Lessons-Learned" along with the utilization of advisory materials will provide a basis for the airworthiness assessment of new and innovative designs. Furthermore, the application of these techniques and the understanding of the results by airworthiness personnel should result in the in-service implementation of high technology Digital Systems which are safe and comply with the reliability requirements of existing advisory documents for the current generation and perhaps the next generation of advanced software based digital flight control and avionic systems.

The results obtained from the ongoing laboratory work, the ongoing certification of current and next generation systems, and the technology advances in hardware/software integration may dictate that the existing tools and methodologies have to be augmented by new techniques such as finite state machines, artificial intelligence, robotic testing and advanced computer based reliability estimation techniques in order to meet the high reliability and availability requirements.

SUMMARY

The results of the above ongoing work are to be published in "Handbook-Volume II, Validation of Digital Systems in Avionics and Flight Control Applications" as data bases and "Lessons-Learned", which will be available in late 1986. (See Figure 4 for the proposed Outline/Table of Contents). In addition, the FAA will sponsor a formal and extensive review of the program and its near-term results in late 1985/early 1986 at a Government/ Industry workshop to be held at NASA-Ames. The results of the ongoing NASA (Ames and Langley) programs and their relationship to the FAA/Industry program will be presented and discussed; and that the ideas, methodologies and technology identified in the earlier 1976, 1979 and 1982 Government/ Industry Workshops will be re-evaluated in light of the current and projected state-of-the-art in Digital Systems Technology; and that emerging certification issues will be identified and discussed.

Figure 1. FAA/NASA/Industry Research Plan - Digital Avionics Systems Airworthiness Issues

Figure 3. Example of Proposed Program Activities

Task Activity	Inputs	Outputs	Payoff	Problems	Certification Issues
I. Systematic FIIS Evaluation	Definition Study-Phase I Implementation of low-level Automated Testing-Phase II Device Description/Complexity (VLSI/VHSIC)	% Monitor Coverage (Software Monitors/ Comparators) Latency-time to recognize effect/complexity Calibration of CPU Self-Test Identification of Instrumentation Requirements/ Costs/Technology/ Complexities Statistical Data for Analytical (Reliability Prediction/Models and Distribution Characterization Identification of Recovery Mechanisms	Increased Accuracy Improved Monitoring Increased Reliability Increased Fail (Operational Capability Increased Modularity Common Processing More Functions Per Device Reduced Space, Weight, and Power Increased Speed Redundancy Management in Hardware Design Reduced Maintenance Less Software Required	System Validation More Complex Software/Device Complexity Real Time Executive Software More Complex Increased Risk of Environmental Test Case Design Very Complex Reliability Issues Emerging	Published Evaluation Criteria/credit not yet available
II. Software Fault Tolerance and Complexity	Software Error Detective and Correction Software Reliability Assessment Comparative Analysis of Fault Tolerant Design Techniques Lockheed IRAD: (a) Software Fault Containment (b) Software Metrics Existing RDFCS Flight Software (Autoland Control Laws) Simulator Investigation Rev	Generation and Demonstration of Practical Complexity Measures for Software Design Quantitative Evidence of Reduced Complexity Identification/Demonstration of Fault Tolerant Software Techniques Validation Testing Results Comparison of Robustness vs Fault Tolerance (Voters/Comparators/Triggering of Recovery Logic)	More Reliable Software Improved Software Design/Practices Solve Generic Software Problems Standard Software Core Modules Standard Algorithms Increased Fault Coverage	Initial Software Development May be More Expensive Quantitative Evaluation More Difficult Requires New Software Metric Tools to be Developed System Validation More Complex May Require New Hardware Developments/ New Architecture/ Designs	Published Evaluation Criteria/credit not yet available
III. Electrical Systems/Power Control/Electro-Mechanical Actuators Certification Issues Studies	USAF/AFFDL C-141 Electro-Mechanical Actuator Study Lockheed IRAD: (a) Smart Servo Actuator Study (b) All Electric Airplane Study NASA-Langley IDEA Program Studies Full Authority Digital Engine Controller Studies/ Certification Issues (a) Hamilton Standard/ Pratt & Whitney (2037 FADEC) NASA-Langley Power Control/Distribution Studies - AIRLAB	Identification of Certification Issues Integrated Systems Architecture Assessment Propulsion/Power Control System Analysis Enhanced Software Fault Tolerance Enhanced Hardware EMI/EMC Tolerance/Susceptibility Identification of Required Data Bases and Information Sources for Advisory and Regulatory Material	Ability to be Commanded Using Digital Buses Allow Distributed Architectures Improved Feedback/ Monitoring Lighter Weight/Less Power Higher Bandwidth Improved Architecture with Decentralized Processing Lightning/EMI/EMP Immunity	System Validation More Complex Reliability Issues Emerging High Risk Technology Severe Environment --High Temperature --High Stress Initial Cost of Interface Units Expensive Reliability of Transmit/ Receive Devices not Yet Established Standards Not Yet Developed	Published Evaluation Criteria Emerging (Not Complete)
IV. Test Case Definition/ Strategies	Bendix Latent Fault Modeling and Evaluation Work Phase I, II and III Fault Insertion and Instrumentation System Advanced Fault Insertion and Simulation Methods	Statistical Data Base for Input to Sensitivity (Reliability Prediction) Models Identification of Sensitivity Analysis Methodologies Test Case Strategies (Fault Tolerant Systems) Design Assessment Methods for Susceptible Areas Generic Methodology for New Microprocessor Designs/ Variations Estimate of Conditional Probability of System Unreliability	Increased System Reliability Reduced Design Costs Standard/Automated Tools Reduced Design Errors Decreased Life Cycle Costs	Standardized Methodology Not Developed/Approved Increased Development Costs Reliability Models Not Fully Developed for Hardware/Software Interfaces Requires Highly Reliable/Validated Languages/Compilers Cost of Implementation Completeness	Evaluation Criteria Emerging (not yet available)

Figure 2. Digital Systems Technology Program Outputs (Does not include Papers Presented at Meetings and Symposia)

Year	Contract	Performing Organization	Report No.	Report Title
1976	-- [7]	FAA HDQTRS/NASA-AMES	NASA TM X-73-73174	Government/Industry Workshop on Methods for the Certification of Digital Flight Controls and Avionics
1978	NAS4-2571[3]	NASA - Dryden Flight Research Center - CSDL	NASA TM-72860	Digital Fly-by-Wire Flight Control Validation Experience
1979 (a)	NAS2-9784[2]	Lockheed-Georgia Co. Battelle Columbus Labs Rockwell International/Collins	NASA CR-152234	Industry Perspectives on Simulation Methods and Research for Validation and Failure Effects Analysis of Advanced Digital Control/Avionics
(b)	NAS2-9783[2]	Bendix Sikorsky	NASA CR-152233	" " "
(c)	NAS2-9782[2]	Douglas	NASA CR-152232	" " "
1979	NAS4-2571[3]	Charles Stark Draper Labs	Contactor Report R-1324	Reliability Analysis of the F-8 Digital Fly-by-Wire System
1981 (a)	NAS2-10270[2]	Lockheed-Georgia Co. Battelle Columbus Labs	NASA CR-166148 DOT/FAA/CT-83-8	Automated Reliability and Failure Effects Methods for Digital Flight Controls and Avionics Systems
(b)	" "	Lockheed-Georgia Co.	NASA CR-166374 DOT/FAA/CT-82/140	Digital Flight Control System Validation - Technology Assessment
(c)	" "	Lockheed-Georgia Co.	NASA CR-166427 DOT/FAA/CT-82/161	Simulator Investigation Plan for Digital Flight Controls Validation Technology
1982 (d)	" "	Lockheed-Georgia Co. Rockwell International/Collins	N/A	Development of the RDFCS Pallet
1982	-- [4]	NASA-AMES Research Center	NASA TM-84276	An Integrated User-Oriented Laboratory for Verification of Digital Flight Control Systems - Features and Capabilities
1983	NAS2-11179[1]	Lockheed-Georgia Co. FAA Technical Center	DOT/FAA/CT-82/154	Integrated Assurance Assessment of a Reconfigurable Digital Flight Control System
1983	DTFA03-81-C-00059	Battelle Columbus Labs FAA Technical Center	DOT/FAA/CT-82/115	Handbook - Volume I. Validation of Digital Systems in Avionics and Flight Control Applications
1983	NAS2-11511[1]	Lockheed-Georgia Co. FAA	DOT/FAA/CT-83-32	Hardware Fault Insertion and Instrumentation System (FIIS) Definition Study
1983	NAS2-10832[4]	Charles Stark Draper Labs	N/A	Acquisition and Installation of Fault Insertion Unit (FIU)
1984	NAS2- [7] NCC 2-303	University of Southern Colorado	Ongoing	Investigation of Advanced Fault Insertion and Simulator Methods
1983	NAS1-17412[5]	Battelle Columbus Labs	--	Software Fault Tolerance for Avionics
1983	NAS2-11162[6]	Chris Kendall Associates	DOT/FAA/CT-83/49	Aircraft Generated Electromagnetic Interference on Future Electronic Systems
1984	NAS 2-246[4]	Stanford University	Ongoing	Executable Assertions and Flight Software. Development of an Aviation Software Testing Methodology
1984	--	FAA Technical Center/ FAA Liason Office-(NASA-AMES)	DOT/FAA/CT-84-9	The Effect of Aircraft Generated Electromagnetic Interference (EMI) on Future Avionic Systems - A Compendium
1984 (a)	NAS1-17529[5]	Lockheed California Co.	(ORAL Presentation)	Systems Study for Integrated Digital/Electric Aircraft (IDEA)
(b)	NAS1-17528[5]	Beoing Commercial Airplane Co.	(ORAL Presentation)	" " "
1984	NAS2-11853[1]	Lockheed-Georgia Co.	Ongoing	Methods for Verification and Validation of Digital Flight Control Systems

NOTES

1. Funded primarily by FAA Technical Center (NASA-AMES providing Pallet Support Funds and Personnel).
2. NASA-AMES/FAA Headquarters (FAA providing $75.0K/year (1977-1981)).
3. NASA-Dryden Funding (F-8 Program).
4. NASA-AMES funding only.
5. NASA-Langley funding only.
6. Funded by FAA Technical Center and U.S. Navy (Washington, D.C.)
7. FAA/NASA-AMES Joint Interest Program - Equal Funding

Figure 4. Proposed Table of Contents for "Handbook - Volume II"

0.0 SUMMARY
    0.1 EXECUTIVE SUMMARY
    0.2 CERTIFICATION TECHNOLOGY RATIONALE

1.0 INTRODUCTION
    1.1 PURPOSE
    1.2 BACKGROUND
    1.3 HANDBOOK SCOPE
    1.4 HANDBOOK ORGANIZATION

2.0 APPLICABLE DOCUMENTS/TECHNOLOGY ASSESSMENT ACTIVITIES

3.0 INTEGRATED ASSESSMENT ASSURANCE METHODOLOGY (DUAL-DUAL ARCHITECTURE)

4.0 QUADRUPLEX DIGITAL FLIGHT CONTROL SYSTEM ASSURANCE ASSESSMENT

5.0 ANALYTICAL SENSOR REDUNDANCY (QUADRUPLEX ARCHITECTURE-PITCH SAS TEST CASE)

6.0 SOFTWARE RELIABILITY ASSESSMENT METHODS

7.0 SOFTWARE FAULT TOLERANCE:
    7.1 SOFTWARE ERROR DETECTION AND CORRECTION TECHNIQUES
    7.2 FAULT TOLERANT DESIGN (FAULT TOLERANT DESIGNS USING COMPLEXITY MEASURES AND OTHER TECHNIQUES)

8.0 DATA BUS INTEGRITY ISSUES

9.0 GUIDELINES TO ASSESS EMC DESIGNS

10.0 FAST RISE TIME EMC TRANSIENT TESTING

11.0 ADVANCED FAULT INSERTION AND SIMULATION METHODS
    11.1 PHASE I - DEFINITION STUDY
    11.2 PHASE II - IMPLEMENTATION OF LOW LEVEL AUTOMATED TESTING
    11.3 INVESTIGATION OF ADVANCED FAULT INSERTION AND SIMULATION METHODS

12.0 LATENT FAULTS
    12.1 PHASE I - LATENT FAULT METHODS STUDY
    12.2 PHASE II - TEST CASE DESIGN

13.0 LIGHTNING STUDIES (IMPACT ON DIGITAL SYSTEMS)

14.0 EMAS (ELECTRICAL SYSTEMS CERTIFICATION ISSUES
    14.1 SMART SERVOS/ACTUATORS
    14.2 PRIMARY/SECONDARY POWER SOURCES
    14.3 DIGITAL ENGINE CONTROLLERS

15.0 HANDBOOK VOLUME II

16.0 ADVANCED VALIDATION ISSUES
    16.1 ARTIFICIAL INTELLIGENCE
    16.2 REMOTE PROCESSING
    16.3 ROBOTICS (TESTING/CONTROL)
    16.4 ADVANCED ARCHITECTURE (DISTRIBUTED FAULT TOLERANT PROCESSING)
    16.5 ADVANCED CONTROLS AND DISPLAYS
    16.6 VOICE COMMAND PROCESSING
    16.7 SYSTEM INTEGRATION TECHNOLOGY

APPENDIX

84-2606

RECENT ADVANCES IN AIRCRAFT ON-BOARD WEIGHT AND BALANCE SYSTEMS

James P. O'Brien

Engineering Section Manager
Sundstrand Data Control
Redmond, Washington

Abstract

Aircraft Weight and Balance Systems (WBS) impose rigorous requirements upon the weight measuring transducers employed. Temperature, shock, humidity, and vibration extremes, along with operational considerations, dictate high reliability, survivability, and maintainability designs.

The Sundstrand WBS transducers use hermetically sealed accelerometer sensing elements as inclinometers to measure aircraft weight. The acceleration sensor employs a quartz flexure suspended proof mass as part of a servo control loop that produces an output current independent of the output load, and allows full regime self test. A temperature sensing element is mounted on the accelerometer magnet structure, to allow temperature error modeling, that results in a set of 4th-order temperature correction equations. These equation coefficients, along with sensor identification data, are stored in a PROM that is assembled with the sensor and control circuitry into the end-item WBS transducer. In operation, stored transducer data are transmitted to the WBS computer where transducer temperature corrections are implemented. The WBS Computer also initiates and evaluates transducer self testing and implements failure identification.

The resultant WBS transducers are of a single part number, are fully interchangeable, and do not require individual calibrations. Both laboratory and on-aircraft testing have shown the transducers to have met the survivability, reliability, and maintainability design goals.

Introduction

A Weight and Balance System (WBS), for on-board aircraft use, requires independence from ground support equipment. A typical WBS consists of an appropriate set of weight measuring transducers, a computer, a weight display, and associated wiring. The weight transducers are usually installed on, or within, the aircraft landing gear structures, such that they respond to variations in applied landing gear force due to aircraft weight. The summation of the transducer outputs is proportional to weight and, by taking into account the aircraft and landing gear geometry, the individual transducer outputs can be used to compute the aircraft center of gravity. Weight and center of gravity information from the computer is then displayed on the flight deck and can be used for load and trim calculations, fuel quantity and distribution, and cargo loading purposes. For a WBS to be effective, it must be a stand-alone system, independent of both ground equipment and other aircraft equipment.

Historically, the most troublesome component of a WBS has been the weight measuring transducer. By virtue of its location at the landing gear, the transducer is subject to severe environmental extremes: wide temperature variations from desert heat to polar cold, humidity from near zero to one hundred percent, take off induced vibration on rough surfaces, landing shocks, and thermal gradients from braking heat. All of these types of factors influence the transducer reliability, and reliability is a paramount consideration in order to maximize system dispatch and minimize spares provisioning and ground support. Equally important, but often overlooked, is system maintainability. To be successful, a WBS must include extensive self test and monitoring capabilities together with a very high probability of detecting, isolating and annunciating a Line Replaceable Unit (LRU) failure, followed by a maintenance repair activity that requires little time and labor, minimal ancillary equipmemt, and does not require system recalibration. In addition, the dictum of minimizing size, weight, and power consumption applies, particularly for larger aircraft where the number of landing gears require more transducers than a smaller aircraft.

This paper addresses the Sundstrand WBS in general, with emphasis on the transducers developed for WBS use.

System Description

The Sundstrand WBS consists of the following components:

Computer

The WBS computer is a 2 MCU digital computer, having four multilayered printed circuit boards and a Built-In-Test (BIT) display driver. A Z8000 microprocessor, CMOS RAM, and 128k-bit EPROM's constitute the digital processor. A 14-bit Analog-to-Digital (A/D) converter is used for analog data input, and LSI devices are used to implement serial digital, ARINC-429 input/output. A high efficiency switching mode power supply provides all required system power from the aircraft 115vac, 400hz bus.

Air Data Module

Attached to the airframe by a lanyard and fastened to the front of the computer with thumbscrews, the Air Data Module (ADM) contains EEPROM and associated circuitry for the WBS non-volatile memory. Data stored within the ADM includes airframe identification, transducer identification and characteristics, specific airframe calibration data, and failure history data. Use of the ADM allows the WBS computer to be non-airframe specific, such that the computer proper is interchangeable within, and between, airframe types. When a computer is removed and replaced, all airframe and WBS specific data remains with the airframe in the ADM.

Display
The WBS outputs weight and center of gravity data in ARINC-429 form that can be input to the airframe flight deck display or, alternately, input to an optional WBS display panel. Multiple displays can be connected to the WBS per ARINC-429.

Transducers
The WBS transducers are used in the Main and Nose landing gears for weight sensing, and in the aircraft attitude module to determine fuselage attitude angles. All transducers are of a single design and part number and are fully interchangeable. The total number of transducers per aircraft installation is dependent upon the airframe type (number of landing gears) and whether the installation is single or dual channel.

Transducer Mounts
Attached to the landing gear structures, the transducer mounts serve as mounting platforms for the transducers. The attitude module has its own transducer mount. The quantity of transducer mounts per aircraft depends only upon the number of landing gears on a specific airframe. In general, only the quantity of transducers and mounts changes with the application. The computer, ADM, display, and attitude module are constant WBS components, regardless of airframe type or system configuration, and all are fully interchangeable across airframes.

Operation
The operating premise of the Sundstrand WBS is to use acceleration sensing elements, as precision inclinometer transducers, to measure the bending moments of the landing gear structures resulting from the applied aircraft weights. Taking the nose landing gear as an example: with an inclinometer installed in each side of the nose gear axle, the measured axle slopes are dependent upon the applied nose gear weight and the effective nose gear axle spring constant which is determined at initial system calibration. By utilizing sum and difference techniques, the absolute slope of the axle itself can be determined and the applied weight can be computed independent of this slope. The transducers utilize high impedance current outputs to minimize EMI effects upon the transducer to computer wiring. At the computer, the transducer outputs are sum and difference filtered, A/D converted, and input to the Z8000 for further processing. Computer software operates in a foreground calculation, background test mode, with multi level self test and fault isolation annunciation of all LRU elements of the WBS. During self test, each transducer is dynamically tested in each of its sensitive axis directions, all computer circuits are tested down to functional component group level, and the ADM contents are verified by comparing them to a copy stored in RAM at system power up. Detected faults are annunciated by setting the system failure output signal high. During ground self test, any detected faults are then displayed on the computer 8-digit alphanumeric display, in clear text English. No failure encoding schemes are employed.

Upon detection of a failure, any of the WBS LRU components can be replaced without regard to system calibration, except the ADM. Transducers and computer are non-airframe specific and can be freely interchanged. The ADM, which is attached to the airframe by a lanyard, contains system and airframe calibration data and is airframe specific. For reliability reasons the ADM has been simplified to an EEPROM and several passive components. Failure of the ADM requires that its replacement be programmed with calibration data for that airframe, which was recorded at the initial airframe calibration.

Transducer Development

As the recognized heart of any WBS, the weight measuring transducers were the subject of an intensive development program at Sundstrand. The design element chosen for the transducer is the Sundstrand QA2000 accelerometer, which is a quartz flexure suspended, force balanced, servo accelerometer as used in the AV8B, 757, and 767 inertial reference systems.

Sensor
Within the sensor, the acceleration proof mass is suspended by a quartz flexure in an air damped mechanism. Hybrid microelectronics servo the proof mass, producing an acceleration output current that is independent of output load. Included within the sensor is a highly stable, linear current output, temperature sensor mounted directly on the internal accelerometer magnet structure, which produces an output current proportional to absolute temperature. It is this temperature sensor that allows temperature error compensation to be achieved in the WBS transducers. The basic sensor is enclosed in an hermetically sealed, laser welded case, to assure performance in severe environments. The sensor element weighs less than 80 grams and is approximately one inch in length and diameter. Each sensor contains its own voltage regulator that will operate with input voltages of ± 13 VDC to ± 28 VDC.

The sensor proof mass is suspended between two permanent magnets with capacitive pickoffs that are connected to an oscillator detector. Microscopic displacment of the proof mass results in an output signal from the oscillator detector that is amplified and input to the servo amplifier, along with its own compensated feedback. The servo amplifier drives torquer coils, positioned within the permanent magnetic field, that re-establish the proof mass null position. The current drive through the torquer coils is then a measure of the acceleration that caused the original proof mass displacement. The proof mass itself is supported and constrained to allow only one degree of freedom about a single well defined axis. The sensor temperature output is independent of applied acceleration and is scaled at 1 micro-amp per degree Kelvin. A partial listing of the sensor performance characteristics is given below in Table 1.

Sensor Temperature Modeling

As indicated in Table 1, the sensor is temperature sensitive. The three basic parameters of Bias, Scale Factor, and Axis Alignment all have temperature coefficients and all exhibit thermal hysteresis. Prior to the development of thermal modeling, the classical sensor technique is to build into the sensor a degree of temperature compensation - usually by way of selecting components to minimize thermal effects at one or two specified temperatures. This technique has many drawbacks, some of which are: compensation is effective only at the specific calibration temperatures; compensation must be applied to an unfinished sensor element with the possibility of further degradation during the final manufacturing processes; unless extremely complex, compensation is static and incapable of reacting to non-linear changes. To avoid excessive temperature degrading of the sensor, Sundstrand has developed a temperature modeling technique that compensates the thermal response of the sensor across its full temperature range. The result of this temperature modeling is a set of polynomial temperature compensation equations that define the Bias, Scale Factor, and two Axis Alignment temperature compensation curves over the full sensor temperature range. Based upon the sensor temperature output signal, the WBS computer can then utilize these compensation equations to account for all temperature related sensor deviations. Thus, modeling allows temperature compensation to be effected in software as opposed to adjustment or calibration in hardware.

Temperature Modeling Facilities.

To implement sensor temperature modeling, an inertial lab test facility was constructed specifically for that purpose. With an area of 2000 square feet, the test lab has test stations mounted on a passively isolated toroidal test pad, which is approximately 40 feet square, 8 feet thick, and has a weight of over 500,000 pounds. The test pad provides a minimum of 60db isolation between the units under test and external vibration noise sources. The sensor test stations, and the test sequences, are automatically controlled by an HP9825C computer and scanner. Separate test sequencing is used to establish the Bias, Scale Factor, and Axis Alignment compensation equations.

Temperature Modeling Sequence

The temperature modeling approach is applicable to any sensor whose repeatability is superior to both its unmodeled accuracy and its sensor-to-sensor environmental variations. The fundamental requirement for software modeling is sensor repeatability and the QA2000 sensor exhibits nearly perfect turn-on to turn-on repeatability, and excellent environmental repeatability. By using modeling, compensation elements such as trim and calibration resistors may be omitted from the design. Modeling allows characterization to be accomplished on sealed, finished sensors, whose behavior is less likely to further change than one that must be sealed following characterization. Modeling implies extensive testing and the formation of a large dependable data base, and these have been accomplished at SDC by automated, computer controlled facilities. Each sensor is modeled over its full operating temperature range using computer controlled test eqiupment that controls temperature, positions the sensor attitude and measures and calculates the appropriate parameters.

Parameter	Max Limits	Units
Bias	4.0	milli g
Bias Temp. Coeffecient	30	micro g/deg C
Bias Thermal Hysteresis	120	micro g peak to peak
Scale Factor	1.2 to 1.46	milli amp/g
Scale Factor Temp Coefficient	120	parts per million/deg C
Scale Factor Thermal Hysteresis	250	parts per million p-p
Axis Alignment	1.0	milli radian
Axis Align. Temp Coefficient	4.0	micro radians/deg C
Axis Align. Thermal Hysteresis	75	micro radians p-p
Frequency Response	300	hertz
Damping Ratio	0.3 to 0.7	(None)
Vibration Rectification	20	micro g/g2 peak
Linearity	25	micro g/g2 peak
Shock	100	g
Threshold and Resolution	1.0	micro g
Temperature Range	-55 to +95	deg C
Temp. Sensor Output at 20 deg C.	290 to 297	micro amps
Temp. Sensor Output	1.0	micro amps/deg C

Table 1. QA2000 Sensor Performance Characteristics

Production sensors to be modeled are subjected to screening tests at both subassembly and top assembly levels. They are then aged by extensive thermal cycling prior to modeling. The modeling sequence is described in the following paragraphs.

Bias and Scale Factor Modeling

Bias and Scale Factor are modeled using a temperature tumble test which measures the bias and scale factor parameters over a temperature range of -55 deg C to +95 deg C, as the test specimens are rotated between a +1g and a -1g orientation. The test begins at room ambient temperature. With the test specimens sealed into the test chamber, the temperature is reduced to -55 deg C and a cold soak period is entered. The temperature is then raised to +95 deg C for a hot soak period. The first test data is taken at +95 deg C. Temperature is lowered in 30 deg C increments, with data taken at each increment, until -55 deg C is reached, then raised back to +95 deg C in the same 30 deg C increments with data repeated at each step. The temperature is then lowered back to +35 deg C where the final data is recorded. During the temperature slewing, the sensors are tumbled from +1g to -1g and back repeatedly. All data are taken only when the unit temperature is within 0.4 deg C of the target temperature and the temperature gradient is less than 0.05 deg C per minute. The bias and scale factor modeling sequence lasts for 17 hours. The recorded data is reduced off line to bias and scale factors using algorithms that cancel effects of temperature drifts during the measuring sequences. Bias and scale factors are then fitted to a 4th-order polynomial temperature model, in which the independent variable is the temperature sensor current of the unit under test. A polynomial curve is least-squares-fitted to both the descending and ascending temperature test points. The final temperature model equation is the mean interpolation between these two curves. The model equation coefficients are submitted, along with the finished sensor, to be incorporated into the WBS transducer. Test data and polynomial coefficient data are archived by sensor, and ultimately transducer, serial numbers.

Axis Alignment Modeling

Axis alignment modeling is performed on both the hinge axis alignment and the pendulous axis alignment. The modeling sequence is similar to that used for bias and scale factor modeling, with the following exception: rather than simply tumbling the test units between +1g and -1g, a computer controlled dividing head, having a 1-arc-second position accuracy, is used to position the test units in a 4-point tumble about their sensitive axes; the 4-point tumble test lasts for 24 hours. The derivation of the correction equations is similar to those previously described.

Transducer

The WBS transducer consists of the previously described sensor, a machined housing, a printed circuit board that acts as the sensor terminator board and contains other circuitry, a hybrid circuit for coefficient storage, and a failure test indicator LED. The various components are assembled and encapsulated in an RTV compound for environmental and handling protection. The acceleration output current from the sensor is passed through a low pass filter network to the airframe wiring. The self test input from the WBS computer goes through an isolation circuit to the sensor self test input and, in parallel, is desensitized and input to the failure test LED driver network and the coefficient storage multiplexer as a control line. The multiplexer, along with an oscillator counter circuit, a PROM storage circuit, and a voltage to current converter, is part of the hybrid circuit module. The coefficients of the sensor temperature compensation equations are stored within the PROM, along with specific transducer part-number identifiers and checksums. The PROM output is one of the multiplexer inputs, the other being the sensor temperature output. When any given transducer in the WBS is in either the no self test or the negative self test condition, the sensor temperature data is output from the multiplexer through the voltage to current converter. When a positive self test signal is input to the transducer, the sensor output slews to respond on the signal output lines, the multiplexer is switched over to output coefficient data on the temperature line, and the failure test LED is illuminated. The coefficient identification data, which is 1K-bit maximum, is clocked asynchronously through to the computer twice for redundancy. Upon detection of a failure in the transducer, the computer pulses the positive self test signal in a way that keeps the failure test LED visually lit. The presence of the lighted LED on the transducer serves to positvely identify the failed device to maintenance personnel.

In operation, two transducers per channel are mounted on each of the aircrafts landing gears, with one transducer per channel mounted within the fuselage on the attitude mount. All transducers are of a single part number and all carry, within their respective memories, all data pertinent to identify the specific transducer and their particular temperature compensation coefficients. When, for whatever reason, a transducer is replaced, the WBS does not have to be recalibrated in any way. After one aircraft flight the new transducer will have undergone Auto Zero and will be fully accepted by the computer.

Transducer/Computer Interface

Each transducer is wired to the WBS computer to provide operating supply voltages, an output signal and return line, a temperature coefficient signal line, and a self test input. All interface connections are made to the computer analog I/O printed circuit board, which contains provisions for eleven transducer inputs. This number is sufficient to service a B747 aircraft. For other, smaller aircraft, extraneous circuits are simply not used.

As part of the initial calibration sequence, the airframe type is stored in the air data module, and software interprets the specific channels that should be active. The eleven transducer signals and the eleven temperature coefficient inputs are buffered, filtered, and current to voltage converted at the I/O inputs. The temperature coefficient filter configuration is switchable as a function of the transducer self test line status, to account for the difference between the analog temperature signal and the serial digital coefficient signal. A three level multiplexer selects one of: the transducer outputs, the temperature coefficient signals, or the miscellaneous inputs such as power supply voltages etcetera. If the selected signal is one of the transducer outputs, one of the temperature inputs, or one of the miscellaneous inputs, it is passed through a 14-bit A/D converter and hence to the Z8000 microprocessor for further action. If the selected signal is one of the transducer coefficient data inputs, as determined by the positive self test status to that transducer, it is sent through a level shifter circuit to an RS232 receiver, whereby it is input to the Z8000 as serial digital data. At system power up, all of the transducers are ± self tested and their coefficient data compared to that which is stored in the ADM EEPROM. The combination of ± self test, and the coefficient data matching, insures that the transducers are operating properly and are a known system element. The various self test modes function as follows:

No Self Test. The self test command line is turned off and the temperature input filter is switched for temperature data. The multiplexer directs signals to the A/D converter as required by the operating program.

Negative Self Test. The self test line is set negative, the temperature input filter is switched for temperature data, and the transducer LED is turned on. The transducer output signal and its temperature signal are both sequentially directed through the multiplexer to the A/D and hence to the Z8000 where their dynamic responses are evaluated.

Positive Self Test. The self test line is set positive, the temperature input filter is switched for coefficient data, and the transducer LED is turned on. The transducer output signal is multiplexed to the A/D and the Z8000 for dynamic evaluation. The coefficient data is directed through the RS232 input to the Z8000 for further processing.

If a transducer fails its dynamic response test, the test is repeated for verification. A confirmed failure sets the system failure status and results in the failed transducer LED staying on. If the coefficient input passes its own checksum verification, yet fails to match the ADM contents, the transducer is treated as a replacement unit, the coefficient data is temporarily stored in RAM, and after flight termination, is overwritten into the air data module as new data. Coefficient checksum failure on both input data attempts results in a failure notification as previously described, with the appropriate transducer LED on.

Built-In-Test

The BIT capability of the WBS is an integral part of the system philosophy and design. BIT is logically divided into two regimes: on aircraft testing for failure detection and LRU fault isolation; and off aircraft testing for failure verification of the LRU, Shop Replaceable Unit (SRU) fault isolation, and post fix verification of both LRU and SRU. The testing philosophy differs considerably depending upon whether the installation is a single or a dual channel. In a single channel system, the BIT functions operate on either a periodic or a continuous basis such that the entire system is verified operational at all times. A dual channel system enhances system failure detection, without introducing non functional hardware and software, by redundancy of the intended function. Dual channel monitoring is based upon the low probability of two systems experiencing identical result failures simultaneously. This is accomplished via the technique of cross monitoring, wherein the intended function results of the two systems are compared. Only when the cross monitor comparison fails are the majority of BIT functions activated. Properly implemented, a dual channel system has minimum BIT active during normal operation so that BIT failures themselves do not contribute to overall failure rates. In a dual channel system, when one channel is inoperative the other channel operates as if it were a single channel installation. The WBS was designed primarily as a dual channel system with the following design goals:

o Utilize inter system cross monitoring, of intended functional results, to serve as the primary failure detection, such that the entire system remains operative unless, and until, the cross monitor detects channel discrepancies.

o Implement BIT in such a manner that, when the system is installed aboard the aircraft, it is executed only upon prior cross monitor failure and then only for channel fault isolation.

o Fault isolate to the channel level and annunciate the failed channel so that the operative remaining channel can be utilized in a single channel mode.

o Implement sufficient BIT to allow LRU fault isolation by maintenance personnel after a channel failure has been identified. All BIT involved in both channel and LRU fault isolation shall be effectively shut down during normal operation so that the BIT itself does not contribute to failure rates of the system.

o Provide an onboard self test feature that is sufficiently powerful to unconditionally verify proper system operation both on and off the aircraft.

o Provide depot level means for SRU fault isolation of LRU, and LRU verification following repair.

o Minimize the system cost of ownership by utilizing the power inherent in a digital computer to effect all levels of self testing. This is accomplished by building the test capability into the LRU in such a way that the required test equipment is minimized, simplified, and not subject to modification due to system LRU changes.

Based upon a failure analysis of the WBS hardware, a set of 35 distinct BIT functions were identified and the appropriate tests defined. These tests implement failure detection and isolation to both LRU and one of five SRU assemblies within the WBS computer.

BIT Sequencing
Given the set of 35 BIT functions, six conditional subsets of BIT were then defined. The totality of these subsets forms the WBS BIT. The subsets and their functional contents are defined below.

Power-Up Tests. Power-up tests are initiated whenever power is initially applied to the WBS or upon recovery from a power interruption. Tests performed are:

Power supply voltages
EPROM checksum
RAM read/write
A/D converter input
A/D converter interrupt
Transducer self test
Transducer coefficient checksum
ARINC-429 interrupt
ARINC-429 internal
Maintenance required output (INOP)

Continuous Tests. These tests are run continuously, on a periodic basis, during normal operation of the WBS. Tests performed are:

Cross monitor
Z8000 processor
ARINC-429 parity
ARINC-429 status
ARINC-429 activity monitor
Transducer reasonableness

Cross Monitor Initiated Channel Isolation Tests. Upon detection of channel disparities by the cross monitor, the following channel isolation tests are initiated:

Power supply voltages
EPROM checksum
RAM read/write
A/D converter inputs
A/D converter interrupt
Transducer self test
Transducer coefficient checksum
ARINC-429 interrupt
ARINC-429 receiver/transmitter tests
Discrete reasonableness
Transducer temperature reasonableness

Aircraft Self Test. By virtue of a self test initiate switch, either on the flight deck or on the WBS computer front panel, the following tests are initiated for LRU fault isolation:

Power supply voltages
Front panel alphanumeric display
Front panel transducer LED display
Maintenance required output (INOP)
Output warning lights
EPROM checksum
RAM read/write
WATCHDOG timer
Real time interrupt
A/D converter inputs
A/D converter interrupt
Transducer self test
Transducer coefficient checksum
ARINC-429 interrupt
ARINC-429 receiver/transmitter tests

Bench Self Test. Upon removal from the aircraft, the various WBS LRU assemblies may be verified using the WBS test set. The bench test closely duplicates the aircraft self tests and is primarily for failure verification. Each of the WBS LRU's can be tested independently on the test set. For a transducer or ADM, the tests are specific for their function only. For the WBS computer the following tests are run:

Power supply voltages
Front panel alphanumeric display
Front panel transducer LED display
Maintenance required output (INOP)
Output warning lights
EPROM checksum
RAM read/write
WATCHDOG timer
Real time interrupt
A/D converter inputs
A/D converter interrupt
ARINC-429 interrupt
ARINC-429 receiver/transmitter tests

Acceptance Tests. The acceptance test for the WBS differs from most other designs in that the computer carries, within its own memory, the test procedure and sequence for its particular configuration. The ATP test set is a relatively simple device that primarily acts to close the computer I/O loops under computer control. Together with the bench self tests, the ATP tests form a complete and rigorous set of tests, that both verifies LRU operation and fault isolates to computer SRU level. Those tests specifically added by the ATP are:

Hardware identification resistor tests
Power supply voltage measurements
ARINC-429 feedback
Discrete inputs
Discrete reasonableness
Transducer inputs
Transducer temperature inputs

WBS Status Display. When installed on the aircraft, the WBS BIT fault isolates to channel level and, by virtue of the aircraft self test, to LRU level. The eight digit alphanumeric display on the computer front panel is used to identify the failed LRU in clear text English messages. During bench self test, and ATP test, the display verifies the LRU failure and identifies the computer SRU that is failed. The display has a vocabulary of 86 words that are presented in combinations to give WBS status.

Flight History. Using the EEPROM in the ADM, historical records of the WBS operation are maintained for readout by maintenance personnel. The flight history function is of special value in identifying intermittent faults that may occur during flight, but which are absent during ground test. Up to 256 occurrances of any given fault are remembered so that fault trend analysis may be made. A similar technique was used on the Digital Ground Proximity Warning Computer for the A310, B757, and B767 aircraft and has proved valuable for detecting intermittent faults.

Conclusion

The Sundstrand WBS embodies a number of advanced features for WBS equipment:

Software modeling of the transducer temperature characteristics, such that temperature compensation can be effected in the computer software rather than in the transducer hardware, results in all applicable transducers having the same part number and configuration, allows full transducer interchangeability, and eliminates the need to recalibrate the WBS whenever a transducer is replaced. This feature reduces the quantity of spares required and simplifies the spares provisioning logistics.

By removing aircraft and system specific data from the WBS computer and storing it in the nonvolatile memory of the ADM, which interfaces with the computer but which remains with the aircraft upon removal of the computer, the computer itself is rendered airframe non specific. Like the transducers, the WBS computers will be of a single part number and configuration regardless of the airframe types they may be used on, and the spares provisioning and logistics are thereby simplified. At initial system calibration, the specific airframe identification and its calibration constants are stored in the ADM, and are also automatically stored on a magnetic tape cartridge by the Calibration Control Unit (CCU). Upon failure of the ADM itself, the CCU can be used to reprogram the new ADM and the system need not be recalibrated.

Dual channel systems provide failure detection capabilities that a single channel cannot provide, albeit often at the cost of decreased reliability due to the addiional components in the dual channel system. Also, most dual channel systems become inoperative upon detection of a failure since two systems are usually incapable of performing fault arbitration and isolation. The WBS hardware and software have been designed to overcome this deficiency. The WBS incorporates a dual channel Built In Test that is structured in a pyramidal manner. When aircraft power is applied to the WBS, an extensive power up BIT sequence is initiated. Having passed the power up BIT, the system cross monitor is activated. The cross monitor itself has seven functional levels, wherein each system computer passes test sequences and test results to the other system computer, in a manner that allows each to evaluate both its own and the others responses. When the seventh interactive level is reached the system is deemed fully operational. If any lower level fails, additional tests are activated to perform channel fault isolation. In a similar manner, if the cross monitor detects a discrepancy during normal operation, the cross monitor drops down one or more levels to verify the discrepancy and initiate channel fault isolation if required. Whenever the WBS is powered up, a set of software initiated continuous tests are run that input results to the cross monitor. These continuous tests form the second level of BIT, while the channel fault isolation tests form the third level of BIT. If the channel isolation tests are successful, a system error is output but the good channel remains operative. Inability to distiguish the failed channel results in a system inoperative output. The fourth BIT level is the on aircraft self test which performs LRU fault isolation, the fifth BIT level is the bench self test which verifies the LRU failure, and the sixth BIT level is ATP which effects SRU fault isolation. In essence, the dual system cross monitoring technique used in the WBS results in a dual system operational reliability that exceeds that of a single channel, and allows single channel operation in the case of a dual channel failure. The acceptance test procedure that is carried within the computer itself results in both hardware and logistical simplification of the associated test equipment.

Having a nonvolatile read/write memory available in the ADM allows the WBS to record any detected faults, including intermittent faults, for later maintenance readout using the computer alphanumeric display. This flight history implementation enables the detection of recurrent intermittent faults and provides data for failure trend analysis use.

84-2607

TIME FLIES

AN IN-SERVICE EVALUATION OF A 4-D FLIGHT MANAGEMENT SYSTEM

Donald A. Moor

Chief Pilot
Lockheed-California Company
Palmdale, California

Abstract

Lockheed and ARMA designed and developed an L-1011 Flight Management System (FMS) over a two year period and certificated this system in 1977 for use with Saudi Arabian Airlines. This original system provides the capability of automatically controlling the airplane/engine combination to an optimum profile throughout the entire flight (climb, cruise, descent, and navigation). This system, still one of the most advanced in the industry, can position the aircraft over an exact geographic location at a precise speed and altitude ("3-D" capability). "4-D" adds the fourth dimension, time, as an aircraft control parameter. This provides the capability of an automated interface with the Air Traffic Control (ATC) system by automatically positioning the aircraft over a pre-determined fix into a terminal area at an exact altitude, speed, and time. This capability, coupled with time based computerized metering of air traffic by ATC, could greatly expedite the flow of traffic into terminal areas and minimize costly delays associated with air traffic congestion at major airports.

A 4-D mode was developed by Lockheed for the L-1011 FMS in response to a NASA request. This was successfully demonstrated in August 1979. Since then, the 4-D FMS was refined, greater flexibility added, and was certified by the FAA in 1983. To determine how a 4-D equipped airplane can exist in today's ATC environment and to also gather data on system performance, a 6 month in-service evaluation was conducted with Delta Air Lines from August to December 1983. This paper will discuss the results of this evaluation and describe the 4-D FMS in detail.

Introduction

The Lockheed Corporation and the Arma Division of AMBAC Industries (now Hamilton Standard) designed, developed, and certified a Flight Management System (FMS) for the L-1011 in 1977. This system provided the capability of automatically controlling the airplane/engine combination to an optimum profile throughout the entire flight (climb, cruise, descent, and navigation). The FMS, in airline service since 1977, was included as standard equipment on the long range L-1011-500 airplane. The original system, still one of the most advanced in the industry, had a 3-D capability in that it would position the aircraft over a pre-determined fix in the terminal area at a precise speed and altitude. "4-D" adds the fourth dimension, time, as an aircraft control parameter. The 4-D mode provides the capability of an automated interface with the Air Traffic Control (ATC) system by positioning the aircraft over a metering fix in the terminal area at an exact altitude, speed, and time. With time based computerized metering and spacing of arrival traffic by ATC, this 4-D FMS could greatly expedite the flow of traffic into terminal areas and minimize costly delays associated with air traffic congestion at major airports.

FMS System Description and Operation

System Description

The 4-D Flight Management System is a fully automatic navigation and performance system coupled to the existing L-1011 autopilot and auto throttles. The FMS can be engaged immediately after takeoff for fully automatic climb, cruise, and descent. The FMS provides optimum flight parameters for any flight condition from climb through descent and visually displays this information to the flight crew, whether the airplane is fully coupled to the autopilot and autothrottles or being flown manually. The design objective from the outset was to have the FMS operate as a logical extension of the basic L-1011 autopilot/autothrottle system.

The FMS hardware consists of a digital computer with 64K words of memory and a Control and Display Unit (CDU) that is the interface between the flight crew and the FMS computer. The FMS computer is an outgrowth of the standard L-1011 Area Navigation (RNAV) computer that was certified in 1971 and has been in the airline use since 1972. The memory has been expanded from 8K to 64K words to handle not only the navigation computations, but also to accommodate the extensive software program required to store airplane and engine performance data and perform the required performance calculations.

A typical dual 4-D FMS installation is shown in block diagram form in Figure 1. Inputs to the computer are: engine parameters, central Air Data System parameters (altitude, airspeed, temperature, etc.), and position information from the navigation sensors. The computer processes these inputs through a software program and provides control signals to the autopilot/autothrottle system. As stated before, the CDU is the crew interface with the computer and consists of a cathode ray tube, function controls and an Alphanumeric keyboard. Performance and navigation information from the computer are visually displayed in a conversational format on the CDU (Figure 2).

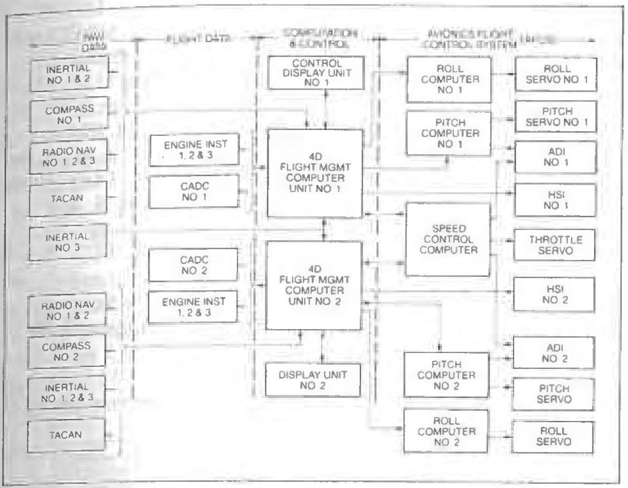


Figure 1. Block diagram of a typical dual 4-D FMS installation.

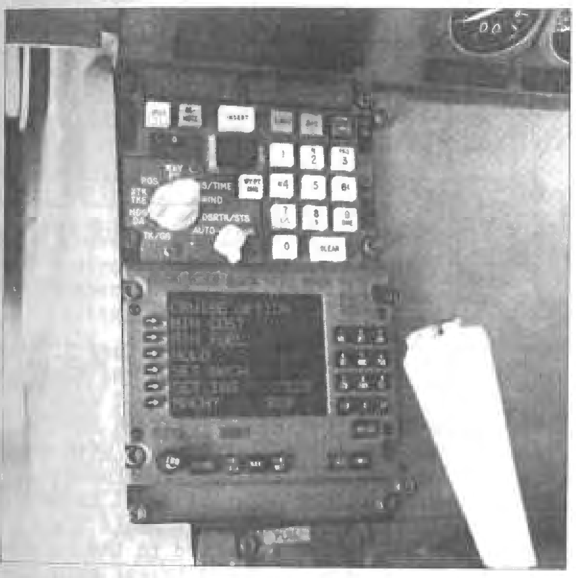


Figure 2. Typical installation of a 4-D Control and display Unit (CDU) in an L-1011.

Time Controlled Cruise

A 4-D FMS flight profile is shown in Figure 3. After a fully-automatic climb, the airplane enters the cruise phase of flight. The FMS CDU indicates the CRUISE has commenced as shown in Figure 4. The mode is indicated on the first CDU line, CRUISE:SET MACH. The second CDU line indicates the CRUISE ALT is 35,000 feet. The seventh line indicates that the Estimated Time Enroute (ETE) to the Beginning of Descent (B*D) is 1 hour, 38 minutes.

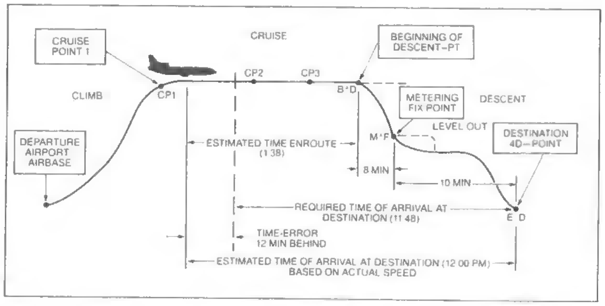
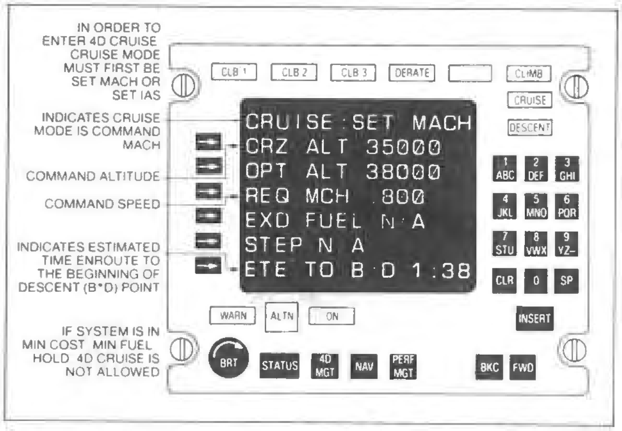

CRUISE POINT 1
CRUISE
BEGINNING OF DESCENT-PT
CLIMB
CP1
CP2
CP3
B*D
METERING FIX POINT
DESCENT
LEVEL OUT
DEPARTURE AIRPORT AIRBASE
ESTIMATED TIME ENROUTE (1 38)
M*F
DESTINATION 4D—POINT
8 MIN
10 MIN
REQUIRED TIME OF ARRIVAL AT DESTINATION (11 48)
E D
TIME-ERROR 12 MIN BEHIND
ESTIMATED TIME OF ARRIVAL AT DESTINATION (12 00 PM) BASED ON ACTUAL SPEED

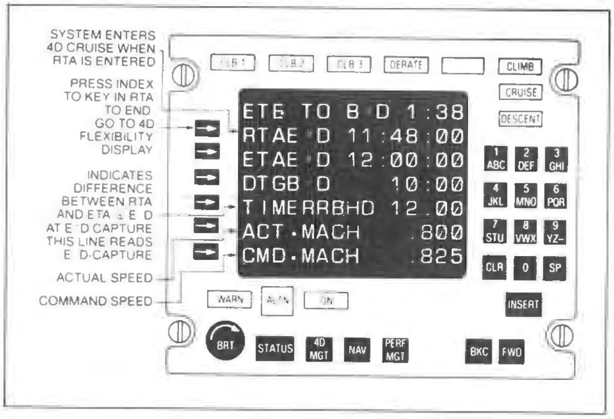
Figure 3. 4-D FMS flight profile.


IN ORDER TO ENTER 4D CRUISE CRUISE MODE MUST FIRST BE SET MACH OR SET IAS
INDICATES CRUISE MODE IS COMMAND MACH
COMMAND ALTITUDE
COMMAND SPEED
INDICATES ESTIMATED TIME ENROUTE TO THE BEGINNING OF DESCENT (B*D) POINT
IF SYSTEM IS IN MIN COST MIN FUEL HOLD 4D CRUISE IS NOT ALLOWED
CLB 1
CLB 2
CLB 3
DERATE
CLIMB
CRUISE
DESCENT
1 ABC
2 DEF
3 GHI
4 JKL
5 MNO
6 PQR
7 STU
8 VWX
9 YZ-
CLR
0
SP
WARN
ALTN
ON
INSERT
BRT
STATUS
4D MGT
NAV
PERF MGT
BKC
FWD

```
CRUISE:SET MACH
CRZ ALT 35000
OPT ALT 38000
REQ MCH .800
EXD FUEL N/A
STEP N A
ETE TO B D 1:38
```

Figure 4. CDU display of 3-D cruise mode status before entering 4-D cruise mode.

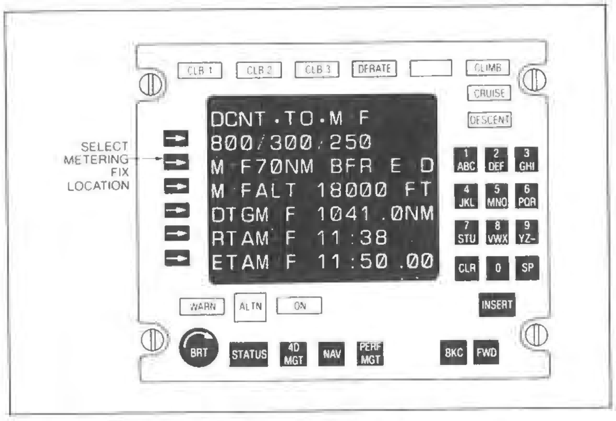
The pilot controls the 4-D function via the CDU "4D-MGT" button. When that button is depressed, the CDU display changes to the 4-D Select mode (Figure 5). The Estimated Time Enroute (ETE) to the Beginning of Descent of 1 hour, 38 miunutes is now indicated on CDU Line 1. The Estimated Time of Arrival (ETA) at the End of Descent (E*D) point (maintaining present speed) is 12:00, as indicated on CDU Line 3.


SYSTEM ENTERS 4D CRUISE WHEN RTA IS ENTERED
PRESS INDEX TO KEY IN RTA TO END GO TO 4D FLEXIBILITY DISPLAY
INDICATES DIFFERENCE BETWEEN RTA AND ETA ≥ E D AT E D CAPTURE THIS LINE READS E D-CAPTURE
ACTUAL SPEED
COMMAND SPEED
CLB 1
CLB 2
CLB 3
DERATE
CLIMB
CRUISE
DESCENT
1 ABC
2 DEF
3 GHI
4 JKL
5 MNO
6 PQR
7 STU
8 VWX
9 YZ-
CLR
0
SP
WARN
ALTN
ON
INSERT
BRT
STATUS
4D MGT
NAV
PERF MGT
BKC
FWD

```
ETE TO B D 1:38
RTAE D 11:48:00
ETAE D 12:00:00
DTGB D     10:00
TIMERRBHD 12.00
ACT.MACH    .800
CMD.MACH    .825
```

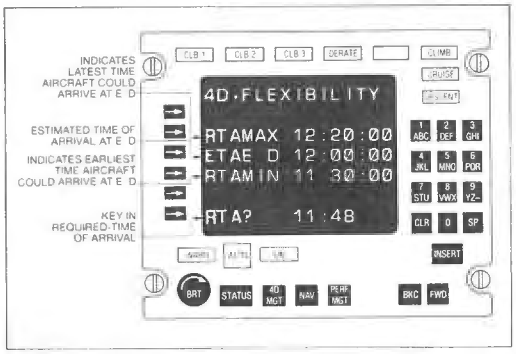
Figure 5. CDU display of 4-D Select mode - relative to End-of-Descent point.

The pilot has the option of having his Required Time of Arrival (RTA) displayed/controlled relative to the End of Descent (E*D) point, as shown in Figure 5 or the Metering Fix Point (M*F) as shown in Figure 6. The RTA to M*F page is selected by paging forward from the RTA to E*D page. In either case, the operation is similar. For purposes of illustration, the operation will be described relative to the End of Descent (E*D) point. When the pilot desires to enter a Required Time of Arrival, he presses the top most CDU index button, as shown in Figure 5. The display then changes to Figure 7 (4-D FLEXIBILITY).



```
DCNT·TO·M F
800/300/250
M F70NM BFR E D
M FALT 18000 FT
DTGM F 1041.0NM
RTAM F 11:38
ETAM F 11:50.00
```

Figure 6. CDU display of 4-D select mode - relative to Metering-Fix point.


INDICATES LATEST TIME AIRCRAFT COULD ARRIVE AT E D
ESTIMATED TIME OF ARRIVAL AT E D
INDICATES EARLIEST TIME AIRCRAFT COULD ARRIVE AT E D
KEY IN REQUIRED-TIME OF ARRIVAL
CLB 1
CLB 2
CLB 3
DERATE
CLIMB
CRUISE
DESCENT
1 ABC
2 DEF
3 GHI
4 JKL
5 MNO
6 PQR
7 STU
8 VWX
9 YZ-
CLR
0
SP
INSERT
BRT
STATUS
4D MGT
NAV
PERF MGT
BKC
FWD

```
4D·FLEXIBILITY

RTAMAX 12:20:00
ETAE D 12:00:00
RTAMIN 11 30:00

RTA?     11:48
```

Figure 7. CDU display of 4-D flexibility with earliest and latest times of arrival.

The system uses the maximum speed (.86 Mach number) that the aircraft can travel along the entire flight plan to calculate the earliest Arrival Time - RTA MIN 11:30 (CDU Line 5). Conversely, the minimum speed (.78 Mach number) that the aircraft can travel along the entire flight plan is used to calculate the Latest Arrival Time - RTA MAX 12:20 (CDU Line 3). The pilot may then select a Required Time of Arrival within these limits; in this case, 11:48, as shown on Line 7 of Figure 7.

After keying in the RTA and pressing the INSERT button, the display changes back to 4-D SELECT (Figure 5). The RTA to Destination is shown on CDU Line 2 as 11:48. The ETA to Destination at present speed of .800 Mach (Line 6) is 12:00 (CDU Line 3). The difference between RTA and ETA naturally generates a time error which is used to modify the speed of the aircraft. This Time Error (12 minutes behind) is shown on CDU Line 5 and is used to generate an aircraft speed command displayed on CDU Line 7 (CMD Mach of .825). The 4-D Cruise Control Law, Figure 8, is a closed loop iteration which is calculated every 1.5 seconds. The speed command will continue to change until the Time Error is driven to zero.

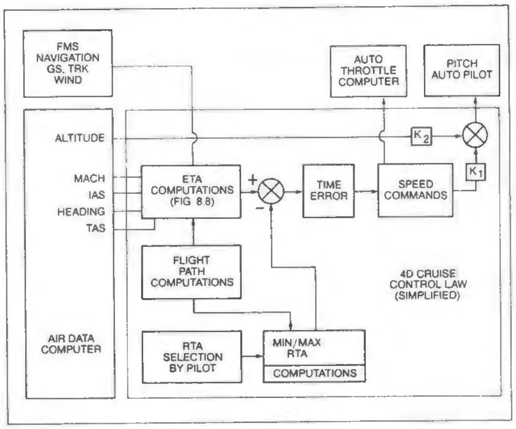


Figure 8. Simplified control law of 4-D FMS cruise mode.

Time Controlled Descent

A 4-D Descent Profile is shown on Figure 9. The Total Descent Time is calculated as the summation of the time for the aircraft to travel each altitude interval (500 feet). At each 500 foot interval, the True Airspeed is summed with the calculated wind at that altitude to produce a Ground Speed. A stored Aircraft Model is used to predict the distance the aircraft would travel from the given altitude and speed to the selected Metering Fix. Then the time to travel each 500 foot interval is calculated. These are summed to produce the total time to descend to the selected altitude.

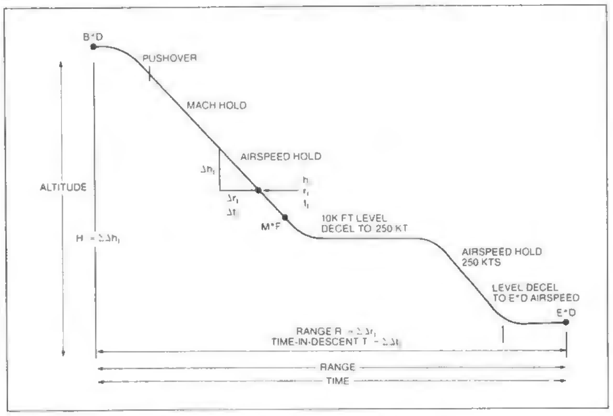


Figure 9. 4-D FMS descent profile.

As the aircraft starts the descent, a range or time error is generated. This translates directly to a speed command to make the Required Time of Arrival. With the throttles essentially at idle, the aircraft is controlled through the pitch axis to speed up or slow down as required. Thrust or drag (speed brakes) is then added as required to maintain the vertical profile.

System Development and Certification

Development of the 4-D system began in 1978 with Lockheed-California Company participating in the NASA Terminal Control Vehicle Program. The existing L-1011 Flight Management System program was modified to include time control algorithms for the descent portion of the profile.

A minimal checkout and flight test program was accomplished at Palmdale during the Spring of 1979 in preparation for a 4-D demonstration for NASA at the Dallas-Fort Worth Terminal Area. Three test flights were all that were required to prove that the descent control laws were working properly. On August 1, 1979, the concept of a 4-D Flight Management System was demonstrated for NASA, the FAA, and an interested potential user, Delta Air Lines, at Dallas-Fort Worth Regional Airport. Three 4-D controlled descents were made to a metering fix Southwest of the airport with the following results: 1st descent, 14 seconds late, 2nd descent, 2 seconds early, 3rd descent, 1 second early.

Having proved the feasibility of the 4-D concept under controlled test conditions, it was agreed that further refinement, followed by evaluation under airline operating conditions, would be worthwhile. Subsequent development of a new 4-D cruise mode, considerably enhanced system flexibility, making it more compatible with the real world ATC environment. This improved system was certified by the FAA in the Spring of 1983 to permit operation of 4-D FMS on scheduled airline flights. Contacts were then made with Delta Air Lines Flight Operations and Engineering management to discuss an in-service evaluation of 4-D on Delta L-1011 revenue flights. Delta had been following 4-D development with great interest and was eager to participate in the program.

In-Service Evaluation with Delta Air Lines

Delta Air Lines' large fleet of L-1011's equipped with Flight Management System was ideal for this FMS in-service evaluation of a 4-D navigational system. The purpose of this operational evaluation program was twofold; gather data on system performance, and qualitatively assess the impact of a single 4-D aircraft in a non 4-D environment.

Since the 4-D software program was considered pre-production by the FAA, some limitations were imposed on the certification basis. One was a 6 month limit on the certification, with renewable options. Another was a restriction that only Delta management pilots who had completed a Lockheed 4-D training course be allowed to operate this program.

Arrangements were made to put the 4-D modified FMS navigation computer on selected Delta flights to Denver, Dallas-Fort Worth, and Atlanta. Denver and Dallas-Fort Worth were chosen as destination airports because the FAA Air Traffic Control Centers (ATCC) in these areas were leaders in a new air traffic program of computerized time-based metering and spacing of arrival aircraft into terminal areas. Metering is the responsibility of the enroute or high altitude air traffic controllers and is enacted whenever the airport acceptance rate is exceeded. During Metering operations, the ground based computer program assigns a Metering Fix arrival time for each airplane. This time is given to the Air Traffic Controller and it is his duty to meter his non 4-D traffic over a geographical position, designated as the Metering Fix, at the proper time. Spacing of arrival aircraft is the duty of the Airport Approach Control or low altitude controllers. These controllers take traffic handoffs from the enroute controllers after the traffic has crossed the Metering Fix. They then sequence this traffic for proper spacing into the landing pattern at the airport. Although Denver and Fort Worth Centers have been using this Metering and Spacing program for the last several years, the remaining ATCC's are just beginning to inaugurate the program. Air Traffic Controllers at Atlanta, Denver and Fort Worth Centers were briefed on the intent of the evaluation and responded with enthusiasm. Arrangements were made to use the Rome, Ga. Variable Omni Range (VOR) facility as the metering fix for 4-D descents into the Atlanta Terminal Control area. Similar plans were made to use Kiowa and Scurry VOR's as the metering fixes for Denver and DFW respectively (Figures 10, 11 & 12).

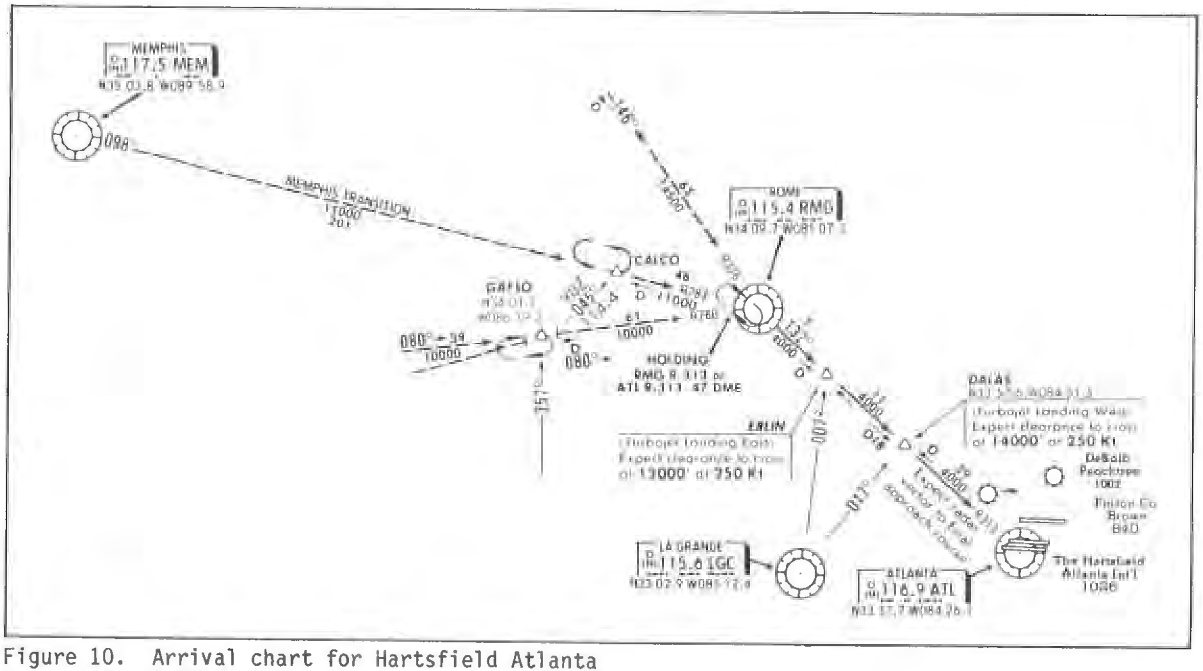


Figure 10. Arrival chart for Hartsfield Atlanta International Airport via Rome VORTAC Metering-Fix.

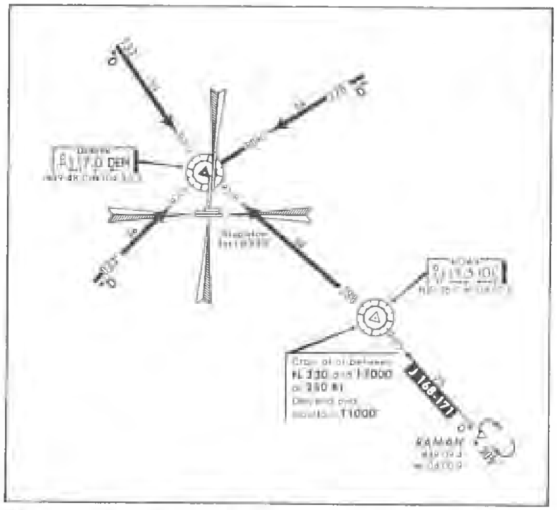


Figure 11. Arrival chart for Denver's Stapleton International Airport via Kiowa VORTAC Metering-Fix.

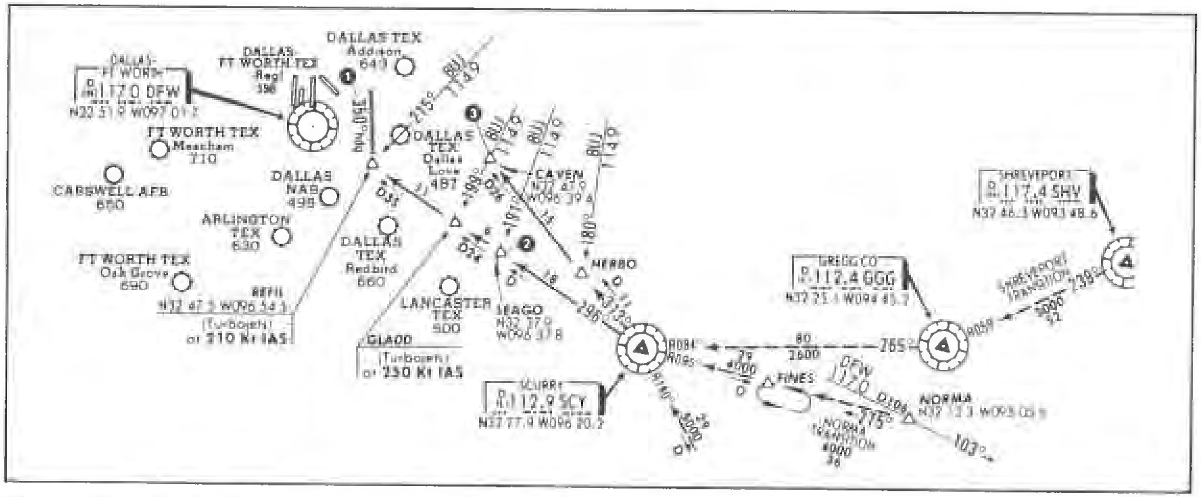


Figure 12. Arrival chart for Dallas-Fort Worth Regional Airport via Scurry VORTAC Metering-Fix.

The first 4-D flight in normal revenue service was an Atlanta-Denver-Atlanta turn-around on 24 August 1983. This was a very convenient schedule, since the same airplane and crew returns to Atlanta following a short turnaround in Denver. The scheduled Atlanta departure and return times (11:44 A.M. and 6:26 P.M.) also made installation and removal of the 4-D computer quite convenient for Delta Maintenance.

A total of 23 4-D descents were attempted during the period 24 August through 15 December 1983. Fifteen descents were successfully completed without ATC interruption. The remaining eight could not be completed to 4-D conclusion due to Air Traffic Control vectors or speed control (off track vectors or speed reductions) to avoid conflict with non 4-D traffic. There were no 4-D system failures or faults on any of the 23 descents attempted. It is interesting to note that if 4-D equipped airplanes could be given discrete metering fix locations separate from those assigned the non 4-D equipped airplanes, the rate of success for this in-service program would have been nearly 100%. Figure 13 shows the successful descents and Metering Fix time errors.

Figure 13

4-D OPERATIONAL EVALUATION DESCENT PERFORMANCE

	Destination Airport	Metering Fix	Time Error over Metering Fix
1.	Atlanta	Rome	2 Seconds - Early
2.	Denver	Kiowa	3 Seconds - Early
3.	Atlanta	Rome	3 Seconds - Early
4.	Atlanta	Rome	4 Seconds - Early
5.	Denver	Kiowa	3 Seconds - Late
6.	Atlanta	Rome	8 Seconds - Late
7.	Atlanta	Rome	0
*8.	Atlanta	Rome	69 Seconds - Early
9.	Denver	Kiowa	6 Seconds - Early
10.	Atlanta	Rome	9 Seconds - Early
11.	Denver	Kiowa	6 Seconds - Early
12.	Atlanta	Rome	12 Seconds - Early
13.	Denver	Kiowa	6 Seconds - Early
14.	Atlanta	Rome	6 Seconds - Early
15.	Denver	Kiowa	11 Seconds - Late

*Large error was due to erroneous winds entered in flight plan.

The average Metering Fix time error was 6.1 seconds early for the 15 descents. If descent #8 was omitted because of the erroneous winds entered in the flight plan, the average Metering Fix time error shrinks to 1.6 seconds early. Over this small sample, the design goal of a ± 8 second 2 time error is nearly achieved.

NOTE: The time errors appear to be within the accuracy tolerances of the navigational sensors (VOR/DME or DME/DME) of the Delta fleet.

Five of the eight Delta pilots trained for this system were able to make at least one flight. Crew comments were very favorable, particularly about how accurately the system performed. Very little additional crew workload was required; to enter the 4-D Cruise mode, the pilot simply entered the Metering Fix Required Time of Arrival (RTA) that he received from the ATC Controller. The system did the rest. FAA Controller comments were also favorable - most saw a reduction in their workload and also expressed some surprise at the 65% successful completion rate in a non 4-D environment. These comments were made by a representative of Atlanta Center during a program debriefing at Delta Air Lines on 16 December 1983. Observers from NASA-Langley also attended the 16 December 1983 debriefing and were impressed that the program was a susccessful as it was. They are convinced that 4-D traffic can co-exist with all other traffic in today's ATC environment. A representative from the FAA, Washington Systems Studies and Advanced Concepts Division, was an observer on the 14 December flight. In addition to being favorably impressed with the accuracy of this system, he felt that there should be a way to provide economic advantages to 4-D equipped airplanes, as he is convinced that airborne avionic equipment will be far ahead of any FAA ground based equipment for the foreseeable future. Since the 4-D descent success rate varied directly with traffic conditions in the terminal area, a plan to assign variable metering fixes to 4-D equipped aircraft was proposed. This would separate 4-D from non 4-D traffic and give 4-D traffic, with its inherent arrival accuracy, priority for landing. A follow-on program to evaluate this variable Metering Fix proposal in the Denver area has been recommended and is under consideration.

SESSION 3 ADVANCED AVIONIC SENSOR SYSTEMS

Chairmen:

Elton A. Hopper
Westinghouse Electric Corp.

Ronald B. Longbrake
ASD/ENA

This session provides technical and programmatic information on the research and development of advanced digital avionic sensor systems including radar, IR, EO, and EW.

84-2613

IMAGING SENSOR AUTOPROCESSOR

Steven C. Sawtelle

Richard J. Jennewine

Automatic Target Classification Group

AFWAL/AARI

WPAFB, OH 45433

Abstract

The Imaging Sensor Autoprocessor (ISA) effort is to augment the sensor operator's target search task and optimize his/her information handling capability. This project will develop target recognition algorithms that are retrainable and/or operate in multi-scenario environments, and which will then be implemented in readily programmable Very High Speed Integrated Circuits (VHSIC) or VHSIC-like hardware. The ISA must recognize five target types: tanks, trucks, jeeps, APCs and mobile guns/missile launchers. The ISA must then cue the target on the sensor operator's display. The performance goals of the ISA are: 90% or greater detection, 90% or greater classification given detection, 10% or less misclassification and one (1) or less false alarms per complete field of view or scene change. There are two contractors working separately on this program: Rockwell International, Autonetics Strategic Systems Division, Anahiem, CA and Honeywell Systems and Research Center, Minneapolis, MN. Both efforts started in late FY79 and will continue through late FY86. The tasks being conducted include algorithm development, system design, hardware fabrication and test. The Rockwell system uses over 100 microprocessors configured in a pyramid architecture. Fabrication of the Rockwell hardware was completed mid 1984. The Honeywell system will use VHSIC technology and hardware fabrication should be complete by mid 1986.

Introduction

The sophistication of reconnaissance and strike systems is continually increasing due to efforts to cope with the high threat operational environments that will be encountered in the 1980s and beyond. Typically, an electro-optical sensor operator or a photo interpreter is required to detect and recognize targets in the displayed imagery of the sensor while simultaneously pointing the sensor and adjusting controls to maintain optimum imagery. With the application of advanced Forward Looking Infrared (FLIR) sensors on high performance aircraft, the information rate and task loading has increased to the point where it is very difficult for a human to perform the target search task in real time. As a result, the limiting factor for successfully applying advanced electro-optical sensors on high performance reconnaissance and strike systems is the sensor/operator interface. However, with the judicious application of image enhancement, target screening and auto cuing, the information of the sensor can be better matched to that of the operator by emphasizing and transferring only relevant target information. The Imaging Sensor Autoprocessor (ISA) program was initiated in 1979 to simulate and test algorithms and design, fabricate and test hardware to perform the automatic target recognition functions mentioned.

ISA History

Three contracts were awarded under the ISA program in September, 1979 to Honeywell Systems and Research Center, Rockwell International Strategic Systems Division and Westinghouse Defense and Electronic Systems Center. The first phase of the program involved the conceptual design and simulation of an algorithm suite that would automatically detect and recognize tactical targets. At the end of the first phase, a simulation test was given to each contractor and the results were analyzed and scored by the Air Force. At this point, the funding for the ISA program was cut and a decision was required concerning the continuation of with all three contracts and the Air Force decided that one of the contracts be terminated. The simulation test results and the proposed Phase II effort of all three contractors were evaluated and the Westinghouse effort was dropped. Rockwell's performance and proposed Phase II effort were satisfactory; therefore, Rockwell continued on with hardware design. Honeywell's performance was marginal and so Honeywell continued on with simulation in hopes of improving performance.

After several more months of effort, Honeywell was given a second simulation test with satisfactory results. Honeywell presented at this time a hardware scheme to house the improved simulation algorithms. The hardware was a modification to the original Prototype Automatic Target Screener (PATS) and the Air Force decided that the performance of the simulation and the previous performance of the PATS did not justify the construction of a slightly improved PATS. Instead, the Air Force decided to have Honeywell continue with advanced, multi-scenario algorithm development in simulation. The processing requirements of the algorithm suite were targeted towards a Very High Speed Integrated Circuit (VHSIC) processing environment. The very high throughput of a VHSIC machine would allow the design of a very complex and computationally intensive, multi-scenario algorithm suite.

The simulation process continued for another year and a half, at which time the Air Force gave Honeywell another simulation test. The results of this test showed improvements over the previous generation of algorithms. Honeywell then presented to the Air Force a preliminary hardware design based on their VHSIC image processing chip set. Based upon the concept presented, the Honeywell ISA hardware task was reinitiated with the requirement to build a VHSIC configured target recognizer. The design effort is scheduled for completion by mid FY85 with fabrication complete in FY86. After optimization of the algorithms and a real-time laboratory test, the hardware will be interfaced with the Advanced Target Acquisition Sensor (ATAS) and flight tested.

Rockwell recently finished fabrication of hardware and integration of system software. Rockwell is currently in the process of optimizing the system to process 10 bit digital data collected by the Air Force. The Air Force will test the ISA in the laboratory during FY85 and FY86. The Rockwell ISA may also be flown in FY87.

ISA Requirements

The Air Force will insure compliance with statement of work requirements during all phases of test. The first requirement is the data types and data formats that ISA is required to process. The ISA must process analog video, in both 525 and 875 line formats. The ISA must also process 10 bit digital data obtained from a digital FLIR output, in either horizontal or vertical scan format. The ISA must process this digital data either live from the sensor or from a High Bit Rate (HBR) recorder using the Advanced Target Recognizer Working Group (ATRWG) HBR format.

The second requirement is the sensor types that the ISA must interface to and process data from. These sensors are an AN/AAQ-9 FLIR, the ATAS FLIR or a television camera in either an analog or digital output format. However, initial performance of the algorithms will be tested against AN/AAQ-9 imagery.

The third requirement is the processing rate of the system. Since the processing rate is dependent upon the actual imagery content, a processing rate range has been required of each contractor. Honeywell must process between 15 and 30 frames of imagery each second and Rockwell must process between 3 and 5 frames of imagery each second. Each contractor is required to process a full field of view from the AN/AAQ-9. Honeywell must also process the full 1008X840 field of view from the ATAS sensor, but Rockwell will process only a 512X384 portion of the ATAS field of view.

Performance specifications of the Honeywell system require detection of 80% of all targets discernible in the field of view. Discernible targets are determined and ground truthed by the Air Force. The target types of interest include tanks, trucks, jeeps, Armored Personnel Carriers (APCs), mobile guns and missile launchers. The Honeywell recognizer is required to correctly recognize 80% of those targets detected. There must not be more than one false alarm per complete field of view change and the probability of misrecognizing a target must be 10% or less. However, Honeywell system design goals and Rockwell system requirements are 90% detection, 90% recognition of those targets detected, 1 false alarm or less per complete field of view change and 10% or less misrecognition of those targets classified.

The Rockwell system must meet the above specifications, but is required to do so only against imagery similar to data that the algorithms have been trained to operate on. Therefore, any time environmentally different imagery is processed by the ISA, then the training set may require an update in order to obtain satisfactory performance. This level of technology was obtainable at the time the Air Force decided to allow Rockwell to design and fabricate hardware. The hardware design that Rockwell implemented is completely reprogrammable and, in so far as is known, is somewhat advanced over other target recognizers being designed and fabricated at that time. However, algorithm technology improvements have occurred since the decision to build the Rockwell ISA, such as with Honeywell. The Rockwell hardware is a step ahead of current systems, but the Rockwell algorithms are not as robust as the Honeywell second generation algorithms; therefore, the Rockwell ISA is considered a 1.5 generation system.

Even though the Honeywell ISA algorithms should be an improvement over the Rockwell ISA, the Air Force has continued the Rockwell ISA 1.5 generation system. The reason for this is that the Rockwell ISA is a completely reprogrammable system that will be invaluable in laboratory simulation, algorithm testing and near-term applications. Also, the Rockwell system is completely modular and can be adapted to almost any sensor rate, frame size and mission requirement. The Rockwell ISA can immediately fill some Air Force automatic target recognizer needs and with the application of VHSIC technology, the Rockwell ISA will fill some of the future needs.

Rockwell System

To accomplish these capabilities, Rockwell has implemented a suite of adaptable algorithms. The first function that the ISA accomplishes is segmentation which generates fields of like intensity using a region growing scheme. All pixel intensities in a local area are compared to a central pixel intensity. Those pixels whose intensities are within system controlled thresholds are included in the region containing the central pixel. Eventually, the entire image has been divided into regions. These regions are then examined using a more refined set of features that are calculated from the region pixel values. Those neighboring regions with features that fall within system controlled thresholds are merged into a single region. Once all possible region merges have occurred, then the regions are windowed for size as a function of range and those regions that are of expected target size are processed further. The window process does not require but is enhanced by using range data that is available from aircraft sensors, manual inkeying, previously stored range values or from passive ranging computations done within the ISA that are based on imagery optical flow. The Rockwell ISA currently does not have a passive ranging capability; therefore, for near-term, in-lab testing, the range value will be obtained from a previously stored data file.

Following the segmentation of the image, all target sized regions are examined to generate a set of features that are used in object recognition. Some of the features calculated include target extent, average intensity value, intensity moments and shape measures. The list of calculated measures are then passed to the logic based classifier for use in target recognition. The logic based classifier examines each calculated feature value via a series of logical program statements which were generated during analysis of training data. For example, if during training, a tank was found to have a feature 5 value of between 40 and 75 and a feature 12 value of between 25 and 55, then an unknown having a feature 5 value of 50 and a feature 12 value of 35 would be called a tank. The example is, of course, a very simple two dimensional case. The actual classifier has several hundred logic statements. Confidence values are also calculated based on the nearness of the unknown's feature value to the center of the feature value range generated during the training process.

All of the features listed above are calculated using pixel values from a single frame. However, one of the features that is a powerful discriminant between man-made tactical targets and false alarms, is the movement of the targets of interest. The Rockwell ISA takes advantage of this powerful discriminant through analysis of the relative position of objects within the image. To accomplish this, the ISA performs a crude segmentation of the image and looks for minimum and maximum pixel position extensions of various segmented image areas. These extensions of the object can be thought of as fingers extending from the object. These extensions then go through a simple statistical process that allows the system to correlate objects extracted in previous frames with objects extracted in the current frame. Once the objects have been correlated, the system can calculate the motion of each object and generate an image scene motion vector. Any object which does not match the scene motion vector is considered moving and is indicated to the rest of the system as a moving target. Since this process requires the segmentation of the image prior to extraction of object motion, this type of motion analysis is called Moving Target Indication (MTI).

All of the information extracted from the image is then put into a feature/object track file. This file may contain up to 50 objects and is used to improve the confidence of results, to smooth the output of the ISA and to dynamically control processing of imagery. An extension of the track file allows the system to perform a Multiple Target Track (MTT) function. This function is performed over the field of regard and for up to ten targets. The track file limit of 50 objects and the MTT limit of 10 targets is due to the current hardware configuration. These files can be expanded, as required by adding more memory to the system.

The Rockwell algorithms were tested by the Air Force in simulation, prior to the construction of the hardware. The test was conducted in the Rockwell Image Processing Laboratory using a VAX 11/780. The processing time for each frame was from fifteen to twenty minutes; therefore, the number of frames used in test was limited to eighty. The results indicated that the algorithms should work in an environment similar to the training data. Since the test, Rockwell has been fabricating the ISA hardware and integrating system software. An evaluation of the Rockwell ISA will occur after optimization of the algorithms in late FY85. Since this test will utilize the real-time hardware, a larger statistically significant number of frames will b used in the test.

The Rockwell hardware design utilizes micro-processor based pyramidal architecture. Th bulk of the processing resides at the base (leve 0) of the pyramid where the pixel processing i accomplished. The regions and informatio extracted at level 0 are passed up the pyramid t the level 1 processors. The level 1 processor perform limited pixel processing across the leve 0 processing boundaries and then merge region generated by the level 0s and then pass thes regions and information to the level 2 processors The level 1 processors also control the level 0 based upon commands from the Segmentatio Executive (SEX) processor. The level 2 processor control the level 1 processors based upon command from the segmentation executive, process the leve 1 boundaries, merge regions generated by the leve 1, process object boundaries and pass region an boundary information to the SEX. The level processors are controlled by the segmentatio executive at the peak of the pyramid. Th functions performed by the segmentation executiv are: 1) pass extracted object boundaries to th feature executive processor, 2) control dynami selection of system processing parameters an thresholds based on information extracted from th imagery and 3) control the configuration of th system. Currently, system configuration i limited to algorithm flow and parameter selection and limited hardware reconfiguration. Th hardware can be reconfigured to process the cente of the field of view if one or more processor fail. The failed processor(s) portion of th field of view is 'moved' to the edge of the image

The feature processors receive processin instructions and object processing boundaries fro the Feature Executive (FEX) which receives objec boundary information from the segmentatio executive. Each feature processor is assigned nearly equal number of objects. These processor extract features from the objects and pass th information to the feature executive which run the logic based classifier. The classificatio results are passed to the display control whic smooths the results and generates target cues fo display on the system monitor. The cue indicating the target types are overlayed on th original video. Currently, evaluation of the IS performance will be done by automaticall comparing previously stored target locations wit the ISA output. In depth analysis of the IS output will be done manually. No performanc evaluation or analysis of the output has been don to date on the system fabricated on this contract

Honeywell System

Under the other ISA contract, Honeywel designed an algorithm suite with the assumptio that VHSIC processing speeds would be available i the foreseeable future. The result is a ver complex set of algorithms requiring 2 to 4 Billio Operations Per Second (BOPS). This complex suit of algorithms should be able to process almost an data, extract objects of interest and classif targets without retraining. The extraction o objects from the background, no matter what th background or target may look like, is capability that is required for a robust targe recognizer. Up until now, the extraction and classification of objects could only be expected on data similar to data that the algorithms had been trained to operate on. Since the Honeywell system will provide a multi-scenario capability, the Honeywell system will be considered to be a second generation system.

Because of the high potential of VHSIC technology, the Air Force decided to reinitiate the ISA hardware task to build a VHSIC configured target recognizer using the Honeywell developed algorithms and chip sets. This second generation system will demonstrate state-of-the-art target recognition capabilities. This system is not intended for use in every mission/scenario because the requirements of many missions may not need the full capabilities of the Honeywell ISA, and in these instances a processor such as the Rockwell system or a limited function version of the Honeywell ISA would fulfill mission needs.

As with most target recognizer systems, the first step is the detection of objects. The detection of objects, or Region Of Interest (ROI) generation is accomplished in several ways. The first method is a statistical method that involves the generation of an edge image followed by an analysis that indicates pixels which are within an objects interior. The second method of ROI generation relies on the output from the Moving Target Detection (MTD) algorithm. In this case, all pixels that are indicated as moving are tagged as an ROI by the system. The third method relies on the fact that an object has already been found, classified and tracked, but, for some reason, was not found by the interior pixel or MTD method in the current frame. If the object has been temporarily lost, then the system will predict where an ROI should be in the current image, based upon the history of the object. Once the ROIs have been found, then the segmentation of the object is done.

The segmentation of the object is accomplished by an analysis of the local area gradient. Since it is known that an object exists at that point in the image, but the exact boundary of the object is not known, then it is possible to find the best gradient boundary position for the object even though the boundary is very weak and undetectable using the edge algorithm that generated the edge image used in ROI generation. This local gradient analysis will extract boundaries that are weak and may not otherwise be found. For example, the dust/smoke trail that billows out from behind a moving vehicle may have the same intensity and similar texture as the object of interest. This results in an ambiguous boundary. The segmentation algorithm can find the optimum boundary of vehicle and smoke by examining the pixels in detail at the image position where a boundary is predicted to exist. The boundary prediction is accomplished through analysis of region pixel expansion rates, edge encounters and local texture measures.

Once the object of interest has been extracted from the image background, a series of features are calculated for further analysis. The first analysis entails the rejection of objects obviously not targets. This is accomplished via size and gross shape measures and all objects that pass these tests are assumed to be targets and further analysis results in the classification of the object as belonging to one of the five target types previously listed. Two types of classifiers, the statistical and syntactic classifiers, are used in the target classification process.

The statistical classifier used in this process is a two part classifier. The primary classifier is a Bayes classifier which is first trained on a series of target objects to generate a feature vector probability curve. This curve indicates the probability that a particular feature vector associated with a particular object is an object of class N. If the decision confidence of the Bayes classifier is low, then the K nearest neighbor classifier is invoked. The K nearest neighbor classifier calculates which K objects in the training feature space are closest to the unknown object. The unknown object is classified as belonging to a class based upon the distance to training objects and the frequency of training class occurrences. Current analysis of the Bayes classifier indicates that the K nearest neighbor classifier may not be needed.

The syntactic classifier may be invoked depending on the target size and the internal detail that is available. This classifier examines the internal structure of the target looking for target components such as a turret, track or hot engine. These components are not classified as a turret, track or hot engine but as components of interest within the boundary of the object under scrutiny. The resultant object is classified as belonging to one of the five target classes based upon the relative placement of components within the object of interest.

To aid in the classification process, several algorithm aides have been developed. One such aid is Moving Target Detection (MTD). The MTD module examines a series of images for sensor platform motion and scene expansion due to platform motion. Once this motion and expansion is known, then the image distortion, known as optical flow, can be compensated for in accurately registering frames of imagery. Once the frames are registered, then the images are subtracted and the difference image will indicate areas where motion has occurred. This process will extract objects that are moving in the scene but are undetectable in a single frame. These results are passed on to the segmentor as ROIs.

A second algorithm aid is passive ranging. Range information is required by the target recognizer in order to calculate the expected size of targets within the image. This range information is available passively by further processing of the optical flow results calculated by the MTD module. It is possible to calculate a range to each pixel in the field of view and current analysis indicates that accuracies of within five percent are possible and that only within fifteen percent accuracy is required for object size discrimination.

With all of the information available from all of these processing modules, it is possible to reduce the bandwidth of the image data. Since the target recognizer knows the areas of interest (targets) and the image motion, then the target recognizer can select image areas for full bandwidth transmission and areas for low pass transmission, along with motion predictions. With this information, a ground unit can reconstruct an image from information updated at much less than a 30 frames per second update rate and from information that has been compressed. This scheme is able to reduce the required transmission bandwidth 10000:1.

The Honeywell ISA algorithms were tested by the Air Force in simulation prior to the design of hardware. The test was conducted in the Honeywell Image Processing Laboratory using a Honeywell mini-computer, an array processor and an image manipulation system. The processing time for each frame required about an hour per image, depending upon image content; therefore, the number of frames used in the test was extremely limited. The results did indicate, however, that the algorithms should perform as required. Honeywell is currently in the design phase of hardware which should be finished by February, 1985 with hardware fabrication complete by February, 1986. At that time, the Air Force will conduct a larger, statistically significant test of the hardware and algorithms.

The preliminary hardware design Honeywell has done under the ISA contract utilizes their VHSIC chips which perform the computationally intensive pixel processing. During this phase of the processing, the pixel information is transformed and reduced to symbolic information. The resulting symbolic information is further processed by MIL-STD 1750 instruction set processors. The 1750 processors analyze features and classify objects as belonging to one of the target classes listed earlier. The 1750 processors also control the VHSIC chips and system processing.

The VHSIC chips are connected on a ring bus and operate on the image in a Single Instruction Multiple Data (SIMD) fashion. The 1750 processors process each object individually, with each object being assigned to a different processor, in a Multiple Instruction Multiple Data (MIMD) fashion. The 1750 processors are connected via a global bus. This architecture allows dynamic reconfiguration of the processors depending upon hardware failure or data processing requirements.

The size of the Honeywell brassboard VHSIC processor is projected to be under 4 cubic feet, weigh less than 200 pounds and consume less than 1250 watts. The Honeywell VHSIC image processor will be delivered late FY 86 and will be tested in the laboratory by the Air Force at that time. This processor is expected to be flight tested during FY 87 by the Air Force.

Conclusion

In conclusion, the two processors currently under development by the Air Force will fill some of the current and future automatic target recognition needs. The architectures developed under these programs allow dynamic and flexible allocation of resources that will be able to adapt to the new and faster processors and sensors, and new and more rigorous algorithms of the future.

84-2610

INFRARED SEARCH AND TRACK SYSTEM DEVELOPMENT

William E. Moore*
Project Engineer
Wright-Patterson AFB, OH

Marvin Spector*
Branch Chief
Wright-Patterson AFB, OH

Vernon D. Best
Program Manager
Wright-Patterson AFB, OH

Abstract

This paper discusses the objectives of the Air Force's Infrared Search and Track (IRST) Advanced Development Program and the basic elements of an infrared system that must be analyzed prior to the design of an IRST system.

Why Is An IRST Needed?

The development and deployment of IRST systems is essential to meet the critical needs of modern air-to-air warfare. These needs are:

- o Operation in jamming environment
- o Long-range detection

The high-velocity targets at high altitude creates a severe time compression problem for successful intercept which imposes a need for long range detection. Because of the intense IR radiation of the high fast target, an Infrared Search Track System can provide this long-range detection capability. It performs this task passively, on both large and small radar cross-section targets, without being hindered by jamming.

The prime features of the IRST against the low altitude tactical targets is again passive detection without degradation by jamming. IRST detection range at low altitude is much less than for the high altitude targets because the targets fly slower, radiate less IR energy and there is significantly more atmospheric attenuation.

IRST Program Overview

Infrared Search and Track Systems (IRSTS) are now under development by the Air Force (AFWAL/AART-1). General Electric and ITT Avionics were awarded contracts in 1981 for the design, fabrication and flight test of an IRST system. Both IRST systems are packaged in a pod that will be mounted for flight test on the left forward inboard missile (AIM-7) location on the F-15 test aircraft. The flight test is a feasibility demonstration to determine the relative quantitative and qualitative effectiveness and utility of an IRST as a complement to an airborne intercept radar. The IRSTS will be evaluated under a wide range of offset angles, relative velocities, ranges and altitudes, emphasizing realistic operational scenarios.

What Is An Infrared System?

When we speak of the infrared, we mean that portion of the electromagnetic spectrum that lies between visible light and the microwave region. Expressed quantitatively, it is the region that extends from a wavelength of .75 to 1000 μm. Therefore, an IRSTS is simply a physical system that searches and tracks targets that emit in the infrared band. The elements of an infrared system are shown in block diagram form in figure 1.

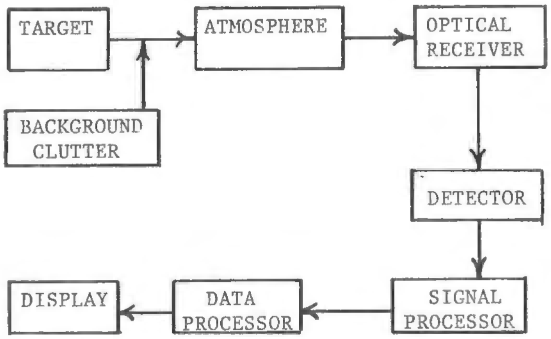


Figure 1

In order to optimally select the spectral band for an IRST, it is necessary to take into account the various factors that impacts the various bands in the infrared region. A predominate factor is the target signature to clutter ratio.

Targets refer collectively to those objects that infrared systems are designed to detect. Primary targets of interest to the Air Force are airborne targets such as aircraft and missiles. The source signature of a target can be expressed as follows:

Source Signature (J_o) =

Reflected earthshine	(J_a) +
Reflected skyshine	(J_b) +
Reflected sunshine	(J_c) +
Skin thermal emission	(J_d) +
Aircraft hot part emission	(J_e) +
Aircraft plume emission	(J_f)

The earthshine is defined as any radiation from the ground or cloud below the aircraft and also includes scattered solar radiation. The skyshine is defined as radiation from the sky or clouds

* IEEE MEMBER

above the aircraft, including any solar scatter from the components. The reflected sunshine is the diffuse and specular scatter by the airframe of directly illuminating sunlight. Thermal emissions result from the airframe that is heated by the aerodynamic flow of air over the fuselage and by solar radiation. The major hot parts are those components in direct contact with the exhaust plume gases. The plume radiation depends strongly on the temperature, composition and velocity of the exhaust gases.

Along with the target signatures it is important that the sources and characteristics of the background clutter environment be examined. Against high altitude targets, the background will consist basically of sky, a few scattered clouds, the horizon and the sun. In the look down missions, the sources of clutter are many and are rapidly changing. Typical background clutter for the look down missions consists of sunlit coulds and cloud edges, the earth and all the terrain features either natural or man made.

The last influential factor for selection of spectral band that will be discussed here is the attenuation of the signal (radiant flux) as it passes through the atmosphere. This general process is called extinction. The transmittance of a path through the atmosphere can be expressed as

$$\tau_a = e^{-x\sigma}$$

Where σ is called the extinction coefficient and x is the path length. Under most conditions, more than one process contributes to extinction, so that

$$\sigma = a + b$$

where a is the absorption coefficient and b is the scattering coefficient. Both a and b vary greatly with wavelength. The molecules mainly responsible for each absorption band are water vapor, carbon dioxide, or ozone. In the infrared region, scattering caused by aerosols is the main contributor. These aerosols include particles such as salt from ocean spray, fine dust blown from the surface of the earth and various carbon particles resulting from combustion. Between the absorption and scattering processes, the absorption process poses a far more serious problem than does scattering.

With the various factors taken into account the spectral bands within the infrared spectrum selected for an IRST is the 3-5 um and 8-12 um band.

The next element in an infrared system is the optics. The purpose of the optics is to collect the radiant flux and deliver it to the detector. In considering the amount of radiant flux collected by an optical system, it is important to know the diameter of the largest bundle of rays that can pass through the optics without obstruction. The physical object that limits this bundle is called the aperture stop. If the diameter of the aperture stop is designated D_o then the area, A_o, is $\pi D_o^2/4$.

The next important area in the design of the optics is to maximize its transmission efficiency, τ_o. This is simply the ratio of the radiant flux at the detector to the radiant flux incident to the IR system. Elements of the optical system that impacts the efficiency is the dome, lenses and mirrors, spectral filter and the dewar window. The dewar is the unit that houses the detector.

The next element is the detector. The detector is simply the device that converts the radiant energy into electrical current. One of the simplest descriptions of detector performance is its responsivity, the detector output per unit input. The responsivity is

$$R = \frac{V_s}{HA_d}$$

Where V_s is the rms value of the fundamental component of the signal voltage, H is the rms value of the fundamental component of the irradiance on the detector in Vcm^{-2} and A_d is the sensitive area of the detector in cm^2. Responsivity, however, does not give an indication of the minimum radiant flux that can be detected. The missing information is the amount of noise in the output of the detector that will ultimately obscure the signal. The noise equivalent power (NEP) is the radiant flux necessary to give an output signal equal to the detector noise. NEP is usually calculated from

$$NEP = \frac{H\, A_d\, V_n}{V_s}$$

where V_n is the rms value of the noise voltage at the output of the detector. Since the noise in the output of the detector contains many frequencies, it is obvious that the noise voltage is a function of the electrical bandwidth of the circuitry.
Using this reasoning, the quantity D^* is introduced:

$$D^* = \frac{(A_d \Delta F)^{1/2}}{NEP}$$

The sensitivity of an IRST system can therefore be characterized from the optical and detector elements. The most common term used to quantify the sensitivity is Noise Equivalent Irradiance, NEI. This is expressed by:

$$NEI = \frac{NEP}{A_o \tau_o} = \frac{(A_d \Delta F_n)^{1/2}}{Ao\, \tau_o\, D^*}$$

With the information known so far, detection range can then be calculated from:

$$R_o^{\,2} = \left[J_o \tau_a\right] \left[\frac{A_o\, \tau_o\, D^*}{(A_d \Delta F_n)^{1/2}}\right]$$

This simplistic equation is to illustrate how the basic elements come together to form what is usally the objective of an IRST, to detect targets at maximum range. In order to accurately predict range more complicated models are necessary than the simple equation. For instance, remember many of the elements are a function of the spectral band therefore they must be integrated over that band.

The next two levels of the infrared system are the signal and data processors. The objective of these processors is to separate the target from the background clutter which is the key to developing a successful IRST system with a low false alarm rate and high probability of detection. Techniques for background discrimination such as multispectal comparisons, adaptive thresholding, pulse width discrimination, scan-to-scan correlation and electronic bandpass filtering are being explored and developed. All of these techniques which involve spatial, temporal or spectral processing have been proven adequate on ground base systems and will improve with improvements in processor hardware. However, the IRST system is an airborne system and today, very little background/clutter data exists which, prevents either contracted IRST system having a proven signal processing technique. Therefore, in addition to the main objective of the flight test which is to test and evaluate the two systems, it becomes necessary to collect as much data as required in order to fully understand the various dimensions that influence the performance of an IRST. In support of this need, the Air Force Geophysics Laboratory will fly their NKC-135A Flying Laboratory at Elgin AFB to collect data for measurement of the target signatures, background and long path transmission between the sensor and the target.

The displays that are being used during the advanced development flight test are the Vertical Situation Display and the Head-up Displays. Unique symbology for the IR detections in search and in track will be provided.

The elements of an infrared system were treated very generally within this paper. However, it is still apparent that many issues must be addressed to optimize the design of an IRST. Other important issues that must be considered that are not the elements of an infrared system are affordability, reliability and maintainability. The Air Force IRST advanced development system is attempting to address and understand many of these issues.

References

(1) Hudson, R. D., "Infrared System Engineering," New York: Wiley and Sons, 1969.

(2) Wolfe, W. L. and Zissis, G. J., "The Infrared Handbook," Washington DC: Office of Naval Research, Department of the Navy.

84-2609

THE APG-66 RADAR AND ITS DERIVATIVE APPLICATIONS

L.J. Kuchinski
T.R. Patton
Westinghouse Electric Corporation
Baltimore, Maryland

Abstract

The APG-66, an airborne radar originally designed for the F-16 A/B aircraft is described here. Applications of this radar have included the U.S. Customs where the radar was used for locating drug traffic, DIVAD in which the radar provided the fire control radar function, the Japanese F-4 and various shipboard and helicopters systems.

Currently, a second generation APG-66 is being developed for the F-16 C/D aircraft. This radar, the APG-68, is even more versatile because the individual LRU's have a greater degree of flexibility. The programmable signal processor has the capacity to accommodate numerous air-to-air and air-to-ground modes plus growth. The transmitter is capable of producing high, low, and medium PRF waveforms with the same efficiency through use of a dual mode tube. The modular receiver is designed for ease of maintenance and incorporates significant advancements in stability and ECCM. A variant of the APG-68 is being produced for use on the B-1B.

Developments have been initiated to further improve this radar by incorporation of an electronically agile antenna to provide superior multitarget performance and insertion of VHSIC technology to provide more processing power and higher reliability. Both these changes can be more accommodated without disturbing the basic form factor of the radar.

The APG-66 radar, designed for the F-16 aircraft is an excellent example of an avionics system with the functional modularity to handle a diverse number of applications (see figure 1) ranging from tactical fighter fire control to battlefield division air defense. The functional modularity is a result of hardware modularity combined with digital processing and control of the individual radar units. This, combined with another important consideration-design-to-cost- suggests why the APG-66 radar has seen such widespread and diverse usage.

The genesis for the APG-66 radar was an in-house Westinghouse-developed radar labeled the WX-200. Radars up to this time exhibited low reliability, were difficult to maintain, and, because of their analog nature, did not have the flexibility to handle new threats, new modes or new applications. The WX-200 radar incorporated the first programmable signal processor (PSP) designed for airborne radar applications, was modular and provided a unique feature-digital control of all the radar line replaceable units (LRUs).

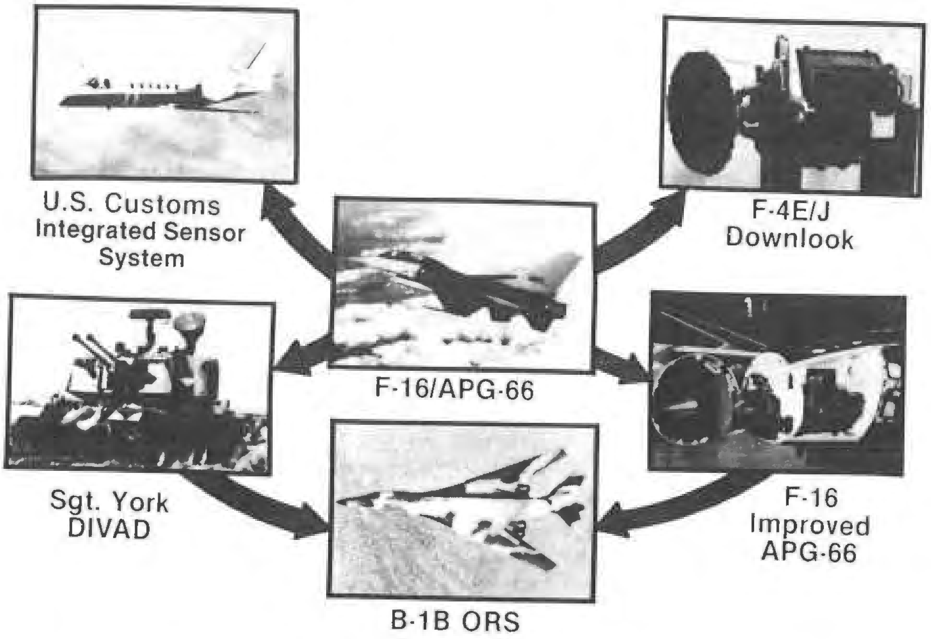
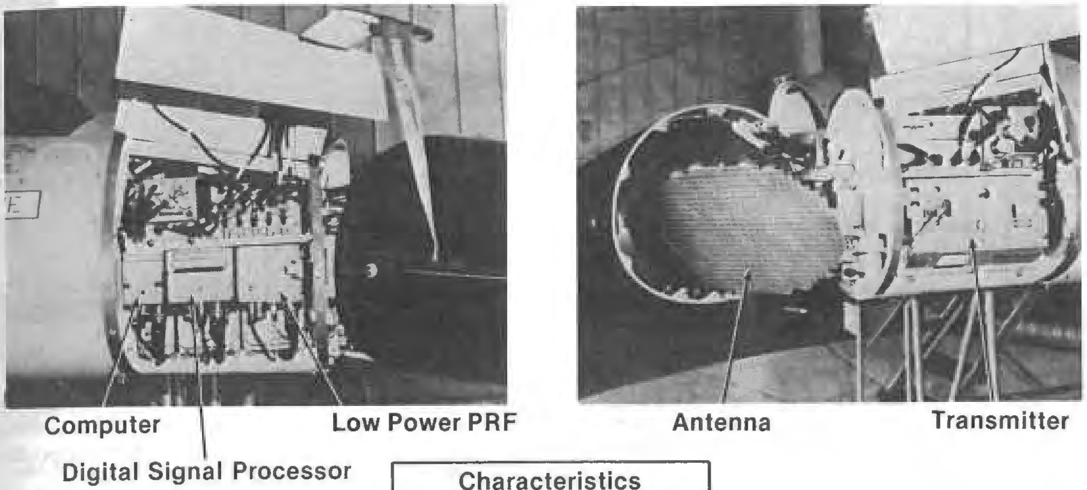


Figure 1. The APG-66 Radar and Its Applications

The product of that development, shown in figure 2, was the APG-66 radar, a modular, all digitally controlled, coherent, pulse-doppler radar. Over 1000 APG-66 radars have been produced to date. Its inherent high reliability and automatic fault isolation capability minimize total life-cycle cost. Demonstrated reliability exceeded 70 hrs in the second year of USAF operational use.

The APG-66 radar consists of six functional LRU's each with its own self-contained power supply. The major elements of the radar are shown in figure 3. A digital multiplex bus system provides communications between the radar computer and the other LRU's. The digital signal processor (DSP) is connected via a dedicated high-speed parallel bus; the other LRU's communicate with the radar computer over a serial bus.



Characteristics	
Weight	296 lb
Power	3.6 kVA
Cooling Air	2.8 kW
Volume	$3.63\ ft^3$

Figure 2. APG-66 Radar Characteristics

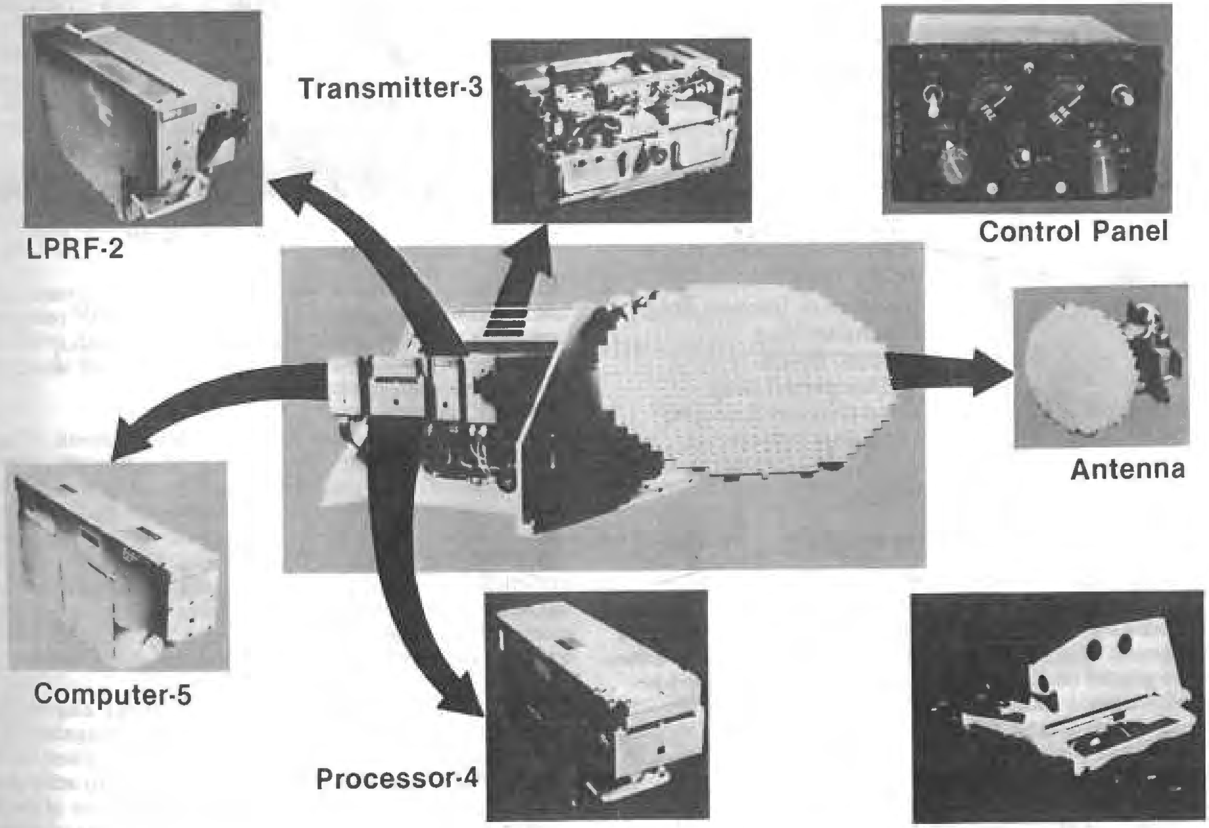


Figure 3. APG-66 Radar Elements

The planar array, gimballed in two axes, provides high gain and low sidelobes over all scan angles. Its balanced electrical drive system makes it light weight, highly reliable and easily maintained.

The low-power receiver unit contains the stable local oscillator, low noise amplifier (LNA), receiver, A/D converters and system clock. All necessary analog processing at RF and IF is performed in this unit.

The transmitter contains an air-cooled traveling wave tube (TWT), a solid-state grid pulser, high voltage power supply and regulators and the protection and control circuitry. The entire transmitter is solid state with the exception of the TWT output tube.

Clutter rejection, digital filtering and detection processing are performed in the DSP. The DSP uses standard integrated circuits mounted in dual in-line packages; LSI devices are used wherever possible. The result is high circuitry density with attendant reductions in cost and weight.

The radar computer configures the radar system for the various modes, directs the DSP to embed symbols in the video output, performs specialized mode processing, routes data to the outside world and controls all self-test and built-in test functions of the radar. The radar is equipped with 48K of programmable, read-only memory and provides throughputs in excess of 350 KOPS for representative instruction mixes.

The APG-66 radar incorporates a number of air-to-air and air-to-surface modes as shown in figure 4. The key air-to-air features are the ability to detect and accurately track low flying targets in rain and high clutter environments and to rapidly acquire and track high-speed, highly maneuvering targets in close-in dogfight engagements. The high peak power, medium PRF waveform permits highly accurate range, angle and doppler tracking at all target aspects.

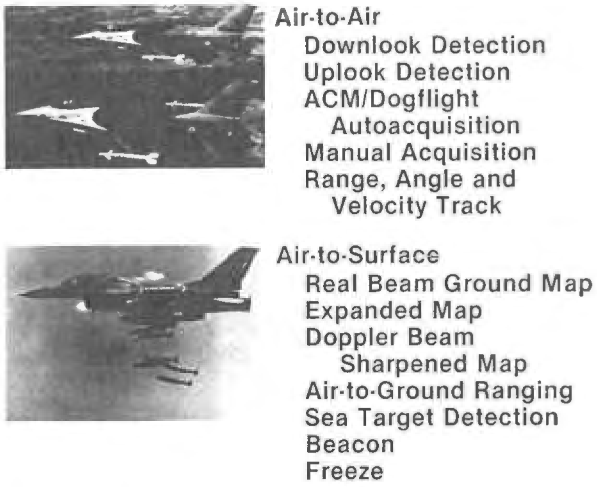


Figure 4. APG-66 Radar Modes

The APG-66 radar also incorporates a wide variety of air-to-surface mapping and tracking modes. The noncoherent real-beam map, beacon and sea target detection modes provide the ability to acquire ground targets and perform all weather weapon delivery; the air-to-ground ranging mode, in combination with an optical sight, allows precise delivery of air-to-ground ordnance. The higher resolution doppler beam sharpened mode provides an 8:1 improvement in resolution.

The physical and operating parameters of the APG-66 radar are summarized in table 1. The radar is an X-band, pulse doppler system. Its weight is just under 300 lbs. Demonstrated reliabilities of 97 hrs have been achieved. The total parts count is just under 9500. The design allows easy access to all LRU's for flight-line maintenance.

Volume	3.6 ft^3 (0.102 m^3)
Weight	296 lb (123.3 kg)
Frequency	X-Band Pulse Doppler
Reliability	97 Hour Demonstration MTBF
Maintenance	5 Minute Flightline MTTR
Electronic Parts	9500
Cooling	Air Cooled at 12 lb/min
Input Power	3580 VA, 400 Hz, 245 Wdc
Range Scales	10,20,40,80 nmi
Elevation Coverage	1,2, or 4 bar
Antenna Azimuth Scan	± 10, ± 30, ± 60 degrees

Table 1. APG-66 Radar Parameters

The success of the APG-66 radar spawned a number of diverse applications, both military and commercial. Probably the most significant of these was the Division Air Defense (DIVAD) radar for the SGT York air defense gun currently being produced for the U.S. Army (see figure 5).

Figure 5. The DIVAD Radar

Worldwide, our armored and infantry units are increasingly threatened by enemy ground attack A/C and missiles. Of particular concern is the attack helicopter which can approach from any direction, pop-up from clutter, acquire a target and launch its air-to-ground missiles in seconds.

The SGT York Air Defense Radar, a direct outgrowth of the APG-66 radar, is a fully coherent, pulse doppler radar that can simultaneously detect and track armed pop-up or hovering helicopters and fixed wing aircraft at all aspects and altitudes. Operating at X-band this radar provides the SGT York DIVAD gun with a fully automatic, fast reaction capability in all weather, clutter, battlefield smoke and dust and ECM environments. Advanced digital technology and software provide a significant reduction in radar costs with ensuring high reliability. The radar has been extensively tested by the Army and is currently in production.

The DIVAD radar consists of six LRU's. A block diagram of the radar is shown in figure 6. The transmitter is a ruggedized copy of the APG-66 radar transmitter. The receiver/stalo, based on the APG-66 design, has been completely repackaged to achieve a new level of modularity in packaging. With the exception of the stalo and the low-noise amplifier (LNA), the unit consists of plug-in modules which allow easy access and removal. Improvements have been made to the stalo to achieve better stability and operation in the demanding battlefield environment. A second receiver channel was added to provide full monopulse tracking capability.

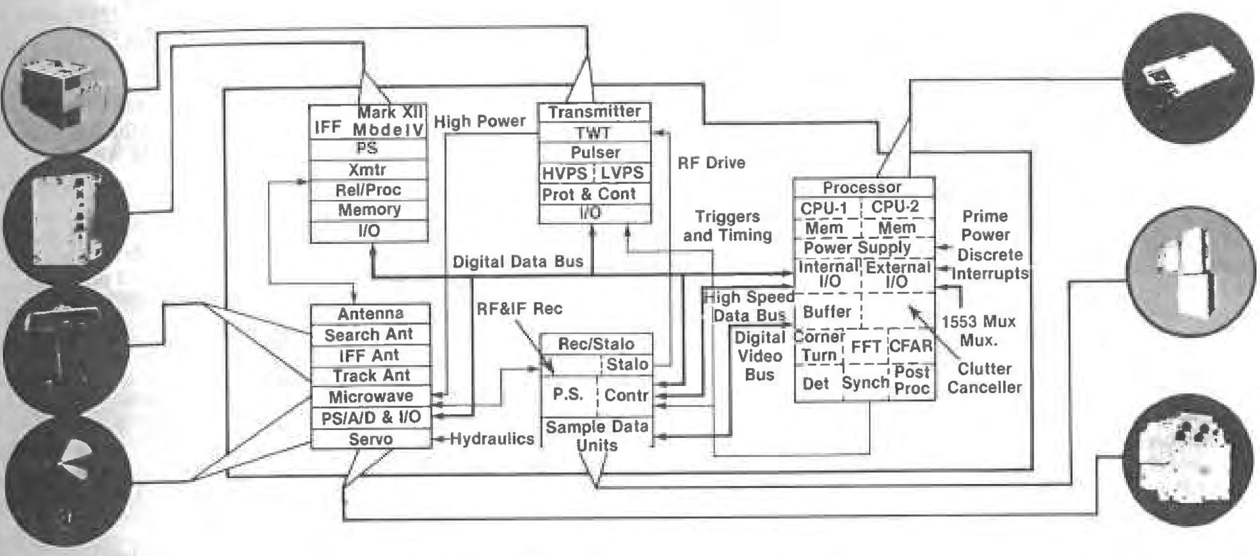


Figure 6. DIVAD Radar Simplified Block Diagram

The DSP and radar computer are housed in a single unit. The DSP is very similar to that in the APG-66 radar; additional post processing capability provides an extra level of programmability and, hence, flexibility. The radar computer is a dual CPU unit with 64K of EPROM and 26K of RAM memory. The processor uses standard integrated circuits mounted in dual in-line packages. The radar computer is programmed in higher order language (JOVIAL J73-I) providing easier software maintenance.

The fire control computer (FCC) is a self-contained, high-speed computer built by Westinghouse and programmed by the SGT York prime contractor, Ford Aerospace. The FCC computes the FCC solution, and controls the gun based on radar, laser, and other sensor inputs. The unit is functionally identical to the radar computer, the only difference being the packaging.

All communications within the radar are via a high-speed serial mux bus with the exception of the high-speed parallel bus between the DSP and radar computer. Communications between the radar computer and FCC are via a standard 1553 mux bus.

The antenna LRU's are unique to DIVAD. Separate stowable search and track antennas are provided. The search antenna uses three, end-fed, slotted, low sidelobe waveguide feeds for low, mid and high beams, providing hemispherical coverage. The antenna structure is armored to protect it against shell fragments and small arms fire. The radar uses a monopulse track antenna which is time-shared with the search antenna. Special processing features are incorporated to sense and compensate for multipath.

By using innovative time-sharing methods with two independently controlled antennas, simultaneous search-while-track capability is achieved (see figure 7). Such simultaneous search-while-track operation is imperative to maintain quick reaction time and effective battlefield management. While accurate track is maintained on one target, the radar search function continues to detect, classify and display other threats for immediate follow-on engagement.

Other features of the radar include detection and resolution of multipath effects, excellent clutter rejection capability, missile detection, helicopter classification and track, and excellent ECCM features.

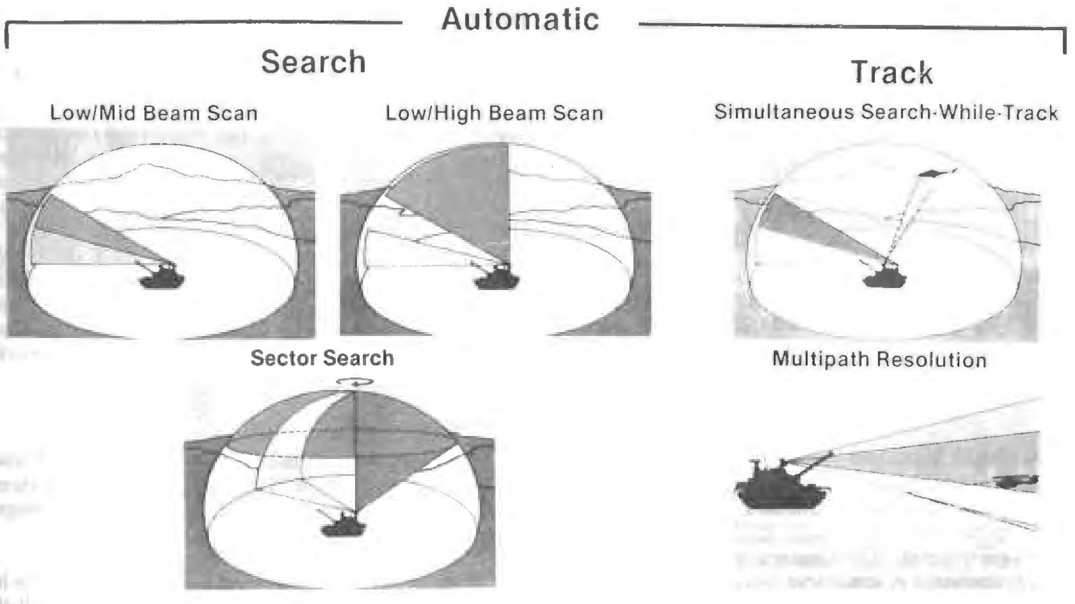


Figure 7. DIVAD Radar Modes

An example of a nonmilitary application of the APG-66 radar is in its use for drug traffic monitoring for the U.S. Customs Service. Housed in a Citation II, high performance A/C (see figure 8) and integrated with other sensors and displays and controls, the APG-66 radar provides a highly effective tool for detecting low flying A/C in all weather and clutter environments. The initial radar achieved a total of >300 operational hrs without a single radar failure.

A significant step in the evolution of the APG-66 radar occurred with the recognition by the Air Force that the F-16 C/D aircraft required a radar with increased performance and flexibility to deal with the increasing sophistication and numbers of the enemy threat. With the advent of the newer and more advanced air-to-air missiles, the need for a longer range radar with multitarget tracking capabilities was recognized. This resulted in an improved radar with three major changes (see figure 9).

First and foremost was the substitution of a single, highly programmable signal processor for the current, separate fixed-program DSP and radar computer.

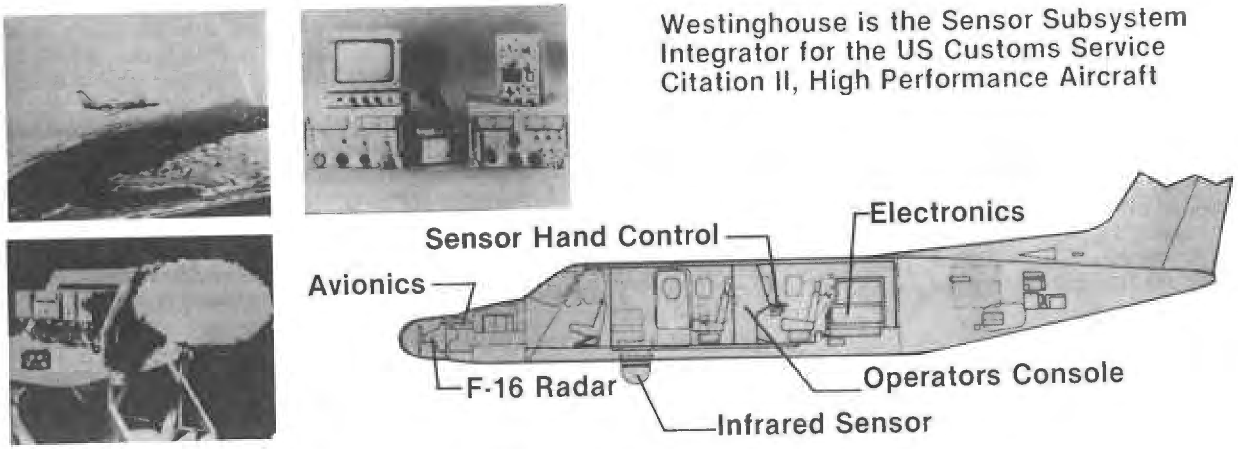

Westinghouse is the Sensor Subsystem Integrator for the US Customs Service Citation II, High Performance Aircraft
Electronics
Sensor Hand Control
Avionics
F-16 Radar
Operators Console
Infrared Sensor

Figure 8. F-16 Radar for U.S. Customs

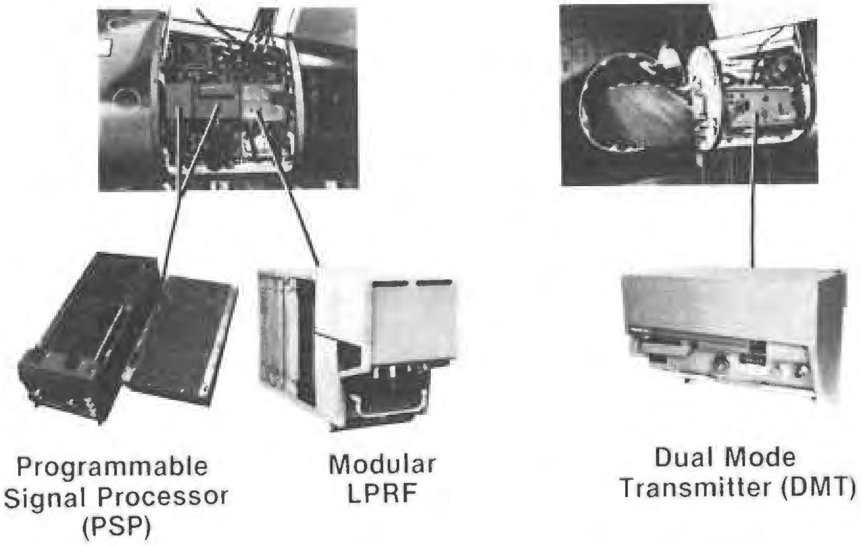

Programmable Signal Processor (PSP)
Modular LPRF
Dual Mode Transmitter (DMT)

Figure 9. APG-68 Radar

Less apparent, but equally important was the introduction of a new dual-mode transmitter that could operate efficiently with a low duty cycle, high peak power waveform or a high duty cycle, low peak power waveform, the choice being tailored to the radar's operating mode at that moment. This combination of low, medium and high PRF operation is achievable without resorting to pulse compressed waveforms.

Finally, the modular low PRF used in the DIVAD radar was adapted to the F-16 to provide improved stability, ease of maintenance and improved ECCM features.

Production deliveries of this improved APG-66 radar, now designated as the APG-68 began in early 1984. Since the new radar is designed to occupy the same space as the existing APG-66 radar, retrofit is easily achievable.

The APG-68 is functionally configured like the APG-66 radar. The major change is the incorporation into one box of all digital processing. The key to the flexibility and increased performance of this new radar is the PSP shown in figure 10. The processing capabilities reside in 31 board pairs (modules) of flatpack construction. Twenty-one of these modules are dedicated to array (signal) processing while the remaining 10 are dedicated to the radar computer. 384K of nonvolatile block-oriented random access memory provides bulk storage for the program instructions.

The unit weighs ≃100 lbs, occupies ≃ 1.0 cu. ft. and dissipates close to 3000 watts. Eight spare slots are available for future processing growth. A combination of MSI and LSI technology are incorporated in this unit. This LRU is divided into three functionally distinct subunits: an array processor, a radar computer and power supply.

The array processor (shown in figure 11) provides the high-speed, digital processing necessary to perform clutter cancellation, digital filtering, detection processing and post detection processing such as range and doppler correlation. The array processor (AP) can perform 192 million operations per second. The AP consists of eight identical signal processing modules which act on incoming data in parallel. The incoming data is stored in a 128K program bulk memory. All operations are under the control of a dual-CPU array controller with 16K of RAM memory. Processed data is stored in the output buffer for access by the radar computer or other AP subunits.

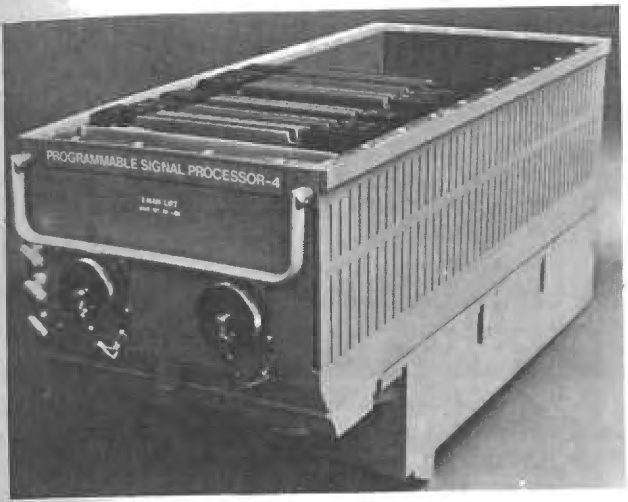


Figure 10. Programmable Signal Processor

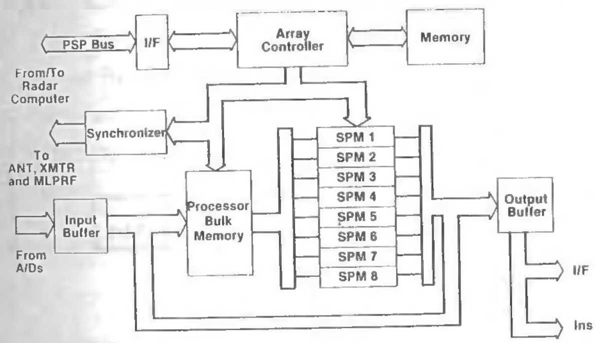


Figure 11. Array Processor Architecture

The radar computer (shown in figure 12) is a dual-CPU, 1750A computer with 48K of high-speed memory allocated to each CPU. Memory can be shared by each CPU under the control of a separate direct memory access controller. All program instructions are stored in a 384K word, nonvolatile, block-oriented random access memory (BORAM). Each CPU provides an effective instruction rate in excess of 1 Megops. The radar computer, as well as the AP, are programmed in higher order language (JOVIAL-J73), providing the ease of software maintenance that comes with higher order language programming.

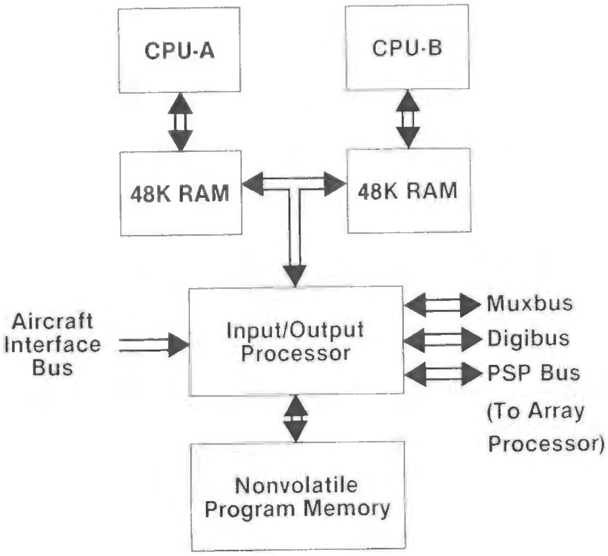


Figure 12. Radar Computer Elements

The APG-68 radar has over twenty-two A/A and A/G modes with the capability to accommodate future tactical fighter needs. Figure 13 depicts the additional air-to-air features incorporated in the APG-68. The most significant capability added is a track-while-scan mode which provides the multiple target tracking and situation awareness capability required to handle the growing and more sophisticated enemy threat. To complement this mode, long-range identification and raid cluster resolution capabilities have also been added. Finally the new dual mode transmitter allows incorporation of a long-range velocity search mode to detect incoming targets.

Similarly the APG-68 provides a number of new air-to-ground modes as shown in figure 14. This includes ground moving target identification and track modes, a fixed target tracking mode and an improved doppler beam sharpening mode providing a 64:1 improvement in resolution. In addition improvements have been made to the air-to-ground ranging mode thus providing even better air-to-ground weapon delivery accuracies. The hardware is designed such that automative terrain follow/terrain avoidance (TF/TA) and synthetic aperture radar (SAR) modes can be easily added in the future.

Just as the APG-66 radar found additional applications, so too has its successor, the APG-68 radar. A prime example of this was its serving as the foundation for the multimode radar on the B-1B airplane. Under full-scale development for the past 2-1/2 yrs, the first radars designated as the APG-164 were delivered early this year.

The APQ-164 (shown in figure 15) was built upon the foundation of existing hardware. The APG-68 provided the transmitter, LPRF, and PSP designs while the two-axis electronically scanned antenna was a direct result of the Electronically Agile Radar (EAR) radar developed and flight tested by Westinghouse for the Air Force Avionics Lab. The two principal modes, TF/TA and SAR, were demonstrated as part of the EAR flight test on the B-52 A/C.

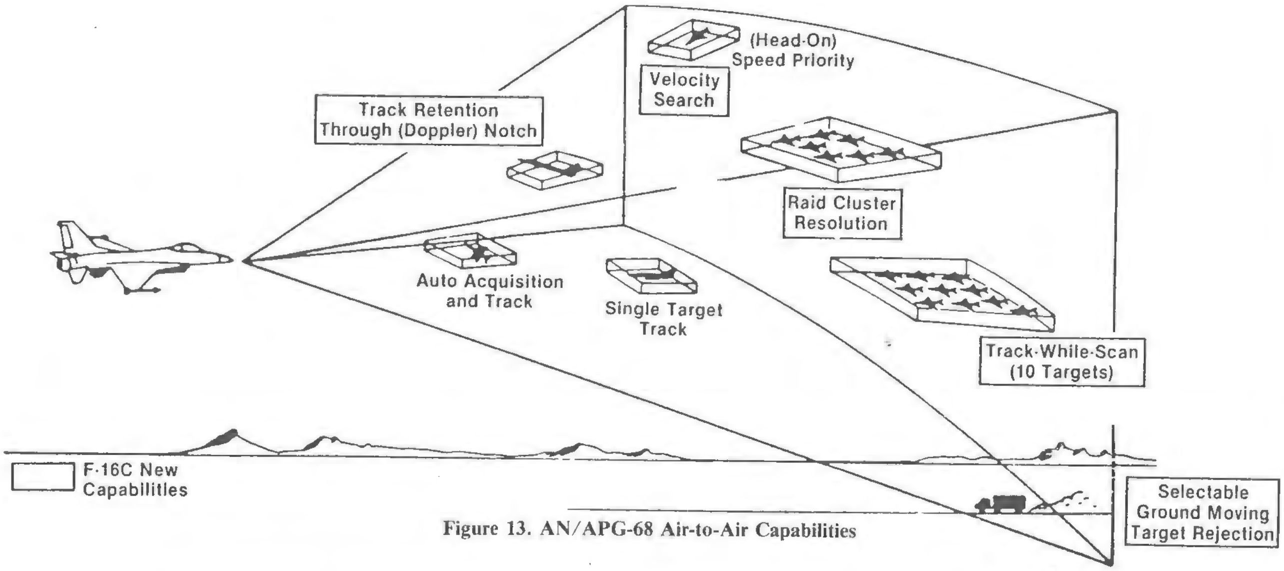


Figure 13. AN/APG-68 Air-to-Air Capabilities

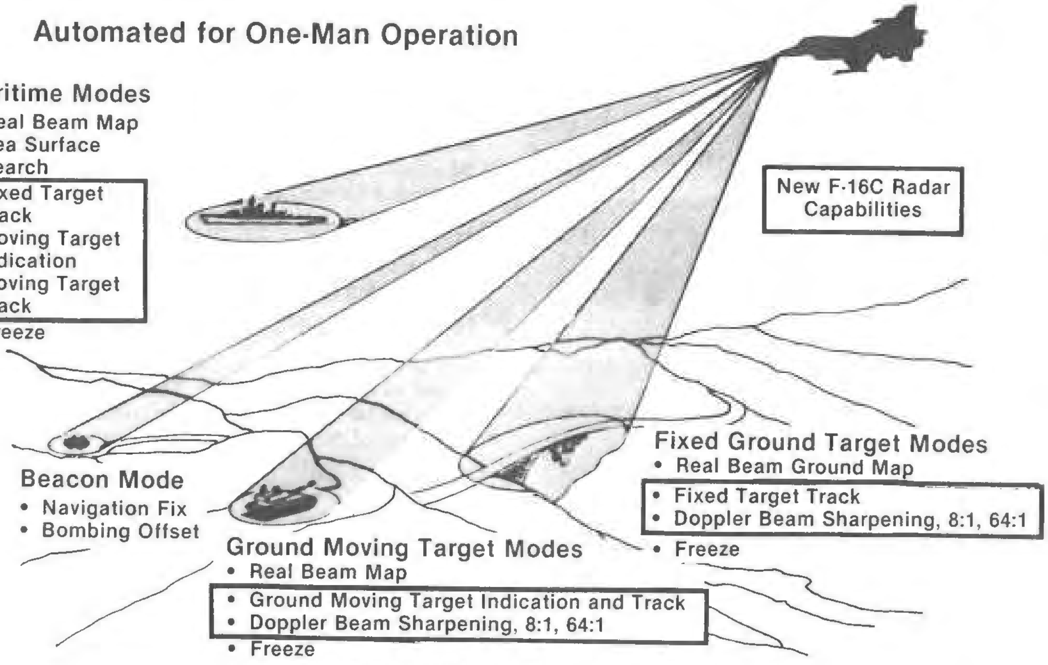
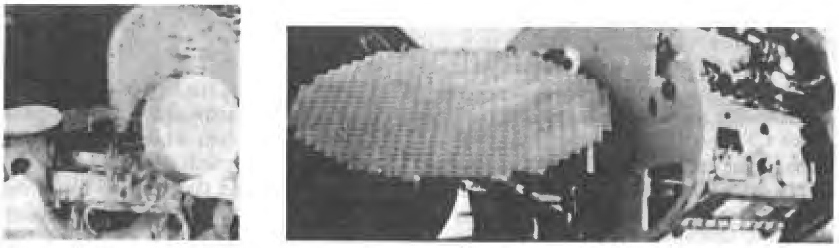

Automated for One-Man Operation
Maritime Modes
• Real Beam Map
• Sea Surface Search
• Fixed Target Track
• Moving Target Indication
• Moving Target Track
• Freeze
New F-16C Radar Capabilities
Beacon Mode
• Navigation Fix
• Bombing Offset
Fixed Ground Target Modes
• Real Beam Ground Map
• Fixed Target Track
• Doppler Beam Sharpening, 8:1, 64:1
• Freeze
Ground Moving Target Modes
• Real Beam Map
• Ground Moving Target Indication and Track
• Doppler Beam Sharpening, 8:1, 64:1
• Freeze

Figure 14. AN/APG-68 Air-to-Surface Capabilities



Electronically Agile Radar

Provides

— New Mode Technology
- Terrain Following
- Real Time SAR

— Demonstrated Performance
- Accurate Navigation
- Low Altitude Flight Tested on B-52 1979

Improved APG-66

Provides

— New Hardware
- Programmable Processor
  - Outgrowth of EAR
  - High Speed
- Modularity
  - Can Add and Delete Functions

Figure 15. APQ-164 Radar

The modes of operation for the APQ-164 are shown in figure 16. The two principal modes of operation are TF/TA and SAR. In addition the APQ-164 radar provides noncoherent mapping modes, ground moving target indication and ground moving target track, precision position update, precision velocity update, and beacon weather modes.

The key feature of the APQ-164 radar is its use of a two-axis electronically phased array antenna. The chief benefits are shown in figure 17. Better performance arises because of the rapid beam switching times permitting interleaving of the various air-to-ground modes with TF/TA. The MTBF of the antenna is estimated to be in excess of 10,000 hrs, a number demonstrated on the EAR and High Energy Laser Radar Acquisition and Tracking (HELRATS) programs. More importantly the design permits graceful degradation in performance in the event of an antenna module failure.

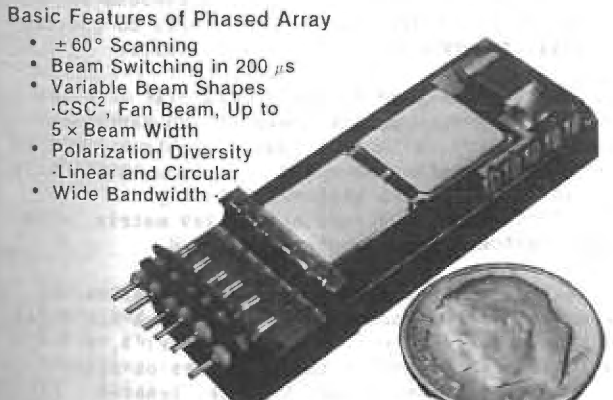
—Synthetic Aperture Map
—Real Beam Map
—Precision Position Update
—Velocity Update
—Terrain Following
—Terrain Avoidance
—Beacon
—Weather
—Ground Moving Target Detection
—Height Above Target

Figure 16. APG-64 Modes of Operation

- Low RCS
- Better Performance
- Higher Reliability
- Reduced Support Cost
- Flexibility for Growth

Figure 17. Benefits of phased Array

The heart of the electronically scanned array are the phase control modules shown in figure 18. The phase control module consists of a ferrite phase shifter and driver; phase shift commands are stored in the integral driver chip. The individual phase control modules plug directly into the array. Beam steering commands are provided by a separate beam steering controller housed on the back of the antenna.



Basic Features of Phased Array

- ± 60° Scanning
- Beam Switching in 200 μs
- Variable Beam Shapes ·CSC^2, Fan Beam, Up to 5 × Beam Width
- Polarization Diversity ·Linear and Circular
- Wide Bandwidth

Figure 18. Electronically Agile Antenna

What will the future evolution of the APG-66/APG-68 radars be? I foresee two major developments. The first is the widespread insertion of VHSIC technology into the processing elements of the radar. The second is the probable introduction of beam agility into fighter A/C. In the near term this will take the form of passive arrays similar to the antenna on the APQ-164; in the 1990's we will see the widespread use of active aperture radars such as the Ultra Reliable Radar (URR); Ultimately we should see the use of broad band active arrays radars integrated with the ECM and ESM systems to make a formulable fighting machine.

The benefits of VHSIC technology are not difficult to foresee: reduced acquisition and support costs, higher reliability, better supportability in the field, and improved performance. Figure 19 shows the improvements that could be realized by replacing the PSP currently in the APG-68 with a PSP using a combination of high density configurable gated arrays and memories available now from the VHSIC program. The results are quite dramatic: four-fold increase in reliability; a factor of seven reduction in power dissipation; a greater than 2:1 reduction in size and weight; and a better than 2:1 increase in speed. With the advent of the full VHSIC technology including the introduction of Ada the results should be even more dramatic.

	Current PSP	VHSIC PSP
• Board Assemblies	31	13
• Weight (lb)	98	40
• Volume (ft^3)	1.45	0.7
• Prime Power (W)	2948	400
• MTBF (Hr)	280	1200
• AP Mcops	16	40
• RC Mips	3.2	6.8
• Maintenance Plan	3-Level	2-Level or 3-Level

Figure 19. PSP Statistics Comparison

The benefits afforded by beam agility are not quite as evident. Although costs for the passive agile radar are reasonable, the cost of active aperture arrays is still high. But the future portends lower costs, greatly improved reliability and much improved performance.

Figure 20 summarizes the benefits of beam agility as afforded by a passive agile array sized for a fighter application. Improved reliability and ease of maintenance have already been demonstrated. Less evident is the dramatic improvement in multitarget tracking capability afforded by beam agility through the decoupling of the search and track functions. Further, true interleaved radar operation is only possible with beam agility.

- Air-to-Air
  - Increased Tracking Accuracy on Multiple Targets Due to High Update Rates
  - Independent Search and Track Coverage of Target
  - Increased Situation Awareness and Prioritization
  - Increased Tracking Range
- Air-to-Ground
  - Instantaneous Mode Interleaving for High-Speed, Including Terrain Clearance and Associated Weapons Delivery Modes
  - Electronic Roll Stabilization
- Overall
  - Independent Positioning for All Modes
  - Improved Reliability Through Graceful Degradation

Figure 20. Benefits of Beam Agility

The APG-66 radar furnishes an excellent example of how a radar can grow and evolve to not only expand its capabilities but also take on new and diverse applications. The key to this flexibility lies in the efficient and widespread usage of digital technology and processing and the modular design of the hardware. This concept has been evident in the design of the APG-66 radar from the start; the primary payoff is reduced cost to the military and civilian users without sacrifice in performance.

84-2614

INTEGRATED TRACKING SOFTWARE FOR MULTIMODE OPERATION

James L. Farrell and David A. Hedland

Westinghouse Defense and Electronic Systems Center
Baltimore, Maryland

ABSTRACT

Modern estimation algorithms can readily use common software for Air-to-Air, Air-to-Ground (moving target), and Air-to-Ground (fixed target) modes. Integration of these algorithms is further enhanced with the use of Bierman's UD factorization.

1.0 Introduction

The use of radar, E/O, ESM, and other sensor subsystems for location of stationary targets and tracking of moving targets, with optimal or suboptimal estimation algorithms, is a well established procedure. In a growing number of applications, many of the pertinent operations coexist in multimode time-shared fashion. A multifaceted implementation may include multisensor fusion and/or several different operations such as air-to-air-(A/A) interleaved with air-to-ground (A/G) tracking; the A/G operation may subdivide further into repeated observation of stationary as well as moving targets. The latter entails what has become known as ground-moving target identification (GMTI) and -tracking (GMTT), while the former ("tracking" of a stationary target) is really a navigation ("nav)" mode. Apparent differences between these functions might prompt the use of different software algorithms, especially if separate modes are assigned to different designers.

The intent of the ensuing discussion is to demonstrate the benefits of an integrated estimation algorithm to cover the above tracking functions (and, for surface applications, to cover ground-to-air (G/A) and ground-to-ground (G/G) tracking as well). In an optimal (Kalman) estimation algorithm, those few parameters that are mode-dependent (e.g., process noise spectral densities, conservative measurement covariance values) can be held in small arrays while over 90% of the operational software code can remain unchanged as modes are switched. An analogous situation holds for suboptimal estimators, with covariance values replaced by bandwidth control parameters. Much of the sensor "raw data" preprocessing can likewise be made uniform across modes.

With this degree of commonality, benefits in expediting development and software validation are immediately apparent. An illustrative example is presented for A/A (mode #1), A/G (moving target; mode #2), and nav update (A/G with a stationary target; mode #3).

2.0 Rationale and Approach

One consideration that influences designers to devise separate software for the three modes just mentioned is a different set of outputs ultimately sought from these operations. For example, ownship is generally placed at the display origin in A/A while the target generally occupies that location for A/G. Also, target velocity and acceleration are hardly applicable to mode #2. Nevertheless, there is the overriding consideration that all three modes under normal conditions receive the same tracker inputs, i.e., azimuth and elevation line-of-sight (LOS) and range from the target sensor (plus range rate when that sensor is a coherent radar), in addition to ownship velocity and attitude data. This collection of information constitutes the set of observables and dynamics that directly comprise the essentials of an estimator algorithm. Origin translation and reinterpretation of states are merely simple mode-dependent operations (defined at the end of this Section) performed on the estimator outputs. The tried-and-true formulation from Refs. 1 and 2, present in a growing list of applications, uses nine states, containing Cartesian components of the range vector (ownship to target), the relative (target w.r.t. ownship) velocity vector, and the total target acceleration vector. The methods used, and the reasons for them, are documented sufficiently well to justify brevity in summarizing some key features:

1. Instead of a 9x9 covariance matrix representation, only the diagonal 3x3 partitions are propagated, with principal directions resolved along sensor rather than INS reference axes. This is tantamount to repetitive similarity transformations performed on the 9x9 matrix, with LOS rotations between updates ignored.

2. Also ignored are the covariance adjustments from small terms connected with rotation and decay of the target acceleration vector (Ref. 3, p.917, Eq.(5b)). Influence of these factors on state VECTOR propagation is not, however, ignored.

3. Observables are formulated in terms of projections along essentially known unit vectors

(i.e., columns of the orthogonal matrix of direction cosines between sensor and INS reference axes). This provides (a) conformance to item 1. above, (b) linearity between observables and states, and (c) isolation of estimation errors from imperfect stabilization effects (Ref. 2).

4. Bierman's UD factorization algorithms are used. This approach enhances multimode adaptation and clarifies its ramifications. Most of the computations defined in the routines given in Ref. 4 are unchanged from one mode to another. In the decrementing routine, only the measurement variances are mode-dependent. The only additional mode-dependent parameters, for the extrapolation routine, are the process noise spectral densities; these are computed from Eq. (5-57) of Ref. 5.

5. Abnormal residual patterns (e.g., large magnitudes or consistent algebraic signs) are easily recognized (through low pass filtering) and counteracted (through resetting the UD matrix elements). Pertinent threshold and timing parameters are also mode-dependent and, once again, readily integrated into a unified algorithm with a small array of control variables.

All that remains to complete the description of approach is a definition of postprocessing operations to be performed in each mode. Beyond the equations given in Section 5 of Ref. 1 (of which the first and third apply only to moving targets), there are simple conversions applicable to stationary targets (mode #3):

A. The negative range vector as defined in this formulation represents a refined estimate for ownship location in a stationary target-centered coordinate frame.

B. The negative of the relative velocity vector as defined in this formulation represents a refined estimate for ownship velocity. Its deviation from INS output velocity at any instant thus corresponds to an instantaneous optimal estimate of ownship INS velocity error (Section 6.5 of Ref. 5). Note that a relationship also exists between the acceleration states obtained from the mode #3 estimator and the INS tilt but, due to the short data averaging intervals being considered, this relationship is not pursued here; acceleration uncertainty effects are outweighted by velocity offsets in the short term.

For moving targets (modes #1 and #2) the designer must be cognizant of postprocessing effectively performed by controller transfer functions or other band-limited devices that receive the target state estimates. This by no means implies that all such effects are undesirable but, whether the designer pays attention or not, spectral shaping does take place --- and it is the overall (estimator-cum-output device) response characteristic that determines system performance.

3.0 Sample Simulation Results

The performance of the tracking algorithm described has been rigorously simulated operating in the three modes mentioned. A single input specifies the mode, which in turn controls setting of initial covariance matrix diagonal elements, measurement variances, and filter averaging time. Another input specifies whether or not range rate (doppler) measurements are available. Except for these initial settings, the software for all modes is identical.

Simulation parameters include the following:

1. The update rate is 30 hertz. All available measured data is assumed to be available at this rate.

2. Sensor pointing uncertainty (ownship IMU attitude errors and sensor gimbal resolution) is one milliradian in each channel (azimuth and elevation).

3. IMU velocity inputs have bias errors of 4, 3, and -4 feet per second in the north, east, and down directions, respectively, with a 0.1 foot per second standard deviation random error.

4. Range measurements have a zero mean, 30 foot standard deviation error.

5. Azimuth and elevation measurements have 10 milliradian RMS error in mode 1 and a 1 milliradian RMS error in modes 2 and 3.

6. Range rate measurements have a 5 feet per second RMS error. Range rate measurements are not available in mode 3.

3.1 Air-to-Air Tracking

The air-to-air (mode 1) scenario consists of an interceptor (ownship) and a target aircraft at an initial range of 11,000 feet and an aspect angle of -150 degrees. The target flies at a constant speed of 800 feet per second and the interceptor at 820 feet per second. At time zero, the target begins a 4 G right turn, and reverses to a 4 G left turn at 10 seconds. The interceptor flies straight until 9 seconds, when it initiates a 4 G right turn, which is reversed to a 4 G left turn at 18 seconds. The latter turns are maintained throughout the 30 second scenario. Both planes maintain a constant 10,000 feet altitude. Figure 1 is a plot of the ground track of each plane.

Figures 2 through 4 show the true and estimated target velocity for the north, east, and down directions, respectively. (Position estimates are routinely so close to the true values that they are indistinguishable on a plot. Hence, they are not reproduced here.) The north and east velocities, where the target motion takes place, show acquisition transient errors lasting a few seconds, and smaller transients caused by the target maneuver reversal at 10 seconds. At other times, the estimated values nearly "overprint" the true values when plotted on this scale. A measure of the noise on the estimate is shown in the down direction plot, figure 4, where the true velocity is a constant zero. After the acquisition transient, the RMS error is less than 15 feet per second.

Figures 5 through 7 show the true and estimated target acceleration. Again, there are transient errors indicated at acquisition and just after the target maneuver reversal. A time delay of one to two seconds between the true and estimated acceleration after the reversal is seen. Figure 6 indicates an RMS error in the acceleration estimate of less than 10 feet per second squared after the acquisition transient.

3.2 Ground Moving Target Scenario

The ground moving target scenario (mode 2) demonstrates the application of the same algorithm to a case with greatly different expected target maneuver characteristics and sensor accuracy. It consists of an interceptor aircraft flying a closing path at an altitude of 2000 feet to a moving target on the ground. The range to the target at time zero is 8000 feet, and it is 40 degrees to the left of the interceptor's ground track. The interceptor flies at a constant 400 feet per second, and initiates a 2 G left turn at 12 seconds. The turn is reversed to a 2 G right turn at 25 seconds, which is maintained until the end of the 30-second scenario. The target is traveling at a constant 45 feet per second, and begins a 0.33 G left turn at time zero. The turn is reversed to a 0.33 G right turn at 10 seconds, and again to a 0.2 G left turn at 17 seconds. The range to the target at the end of the scenario is about 2300 feet. Figure 8 shows the ground tracks of the interceptor and target for this scenario.

Figures 9 and 10 show the true and estimated north and east velocities, respectively. Moderate transient errors are indicated for a few seconds at acquisition and after maneuver reversals at 10 and 17 seconds. During other times (steady state), the velocity tracking error is less than 5 feet per second RMS.

Figure 11 shows the true and estimated north direction acceleration. Again, acquisition and target maneuver reversal transients lasting a few seconds are indicated. The steady state acceleration track error is less than 2 feet per second squared in all directions.

3.3 Air-to-Ground Fixed Target Tracking

The air-to-ground fixed target track scenario (mode 3) shows the use of the same algorithm for a navigation update function. In this use, its purpose is to detect fixed or slowly varying biases in IMU velocity measurements. The scenario consists of a fixed ground target at an initial range of 10,000 feet being approached by an interceptor flying at an altitude of 2000 feet and a speed of 400 feet per second. The interceptor initiates a 3 G left turn at 12 seconds and reverses it at 20 seconds. The range at the end of the scenario (30 seconds) is about 2700 feet. Figure 12 shows the ground track of the interceptor with respect to the target for this scenario. Range rate measurements are assumed to be unavailable in this scenario.

Figures 13 through 15 show the north, east, and down direction target velocity estimates, respectively. However, since the target is known to be fixed, they really represent an estimate of the error in ownship IMU velocity information. The results show that after less than 10 seconds of tracking, the velocity errors are reduced from around 4 feet per second per channel to well under one foot per second. Acceleration estimates remain less than one foot per second squared.

4.0 Conclusions

It has been demonstrated that a single common tracking algorithm can be used to accomplish widely varying mission modes and differing input sources and accuracies. Only initial settings of covariance matrix diagonal elements, filter time constants, and measurement variances are mode and/or sensor dependent. The adaptability of this Kalman tracker is provided by use of Bierman's UDU factorization algorithms.
Use of such a common algorithm for multimode operation can result in great savings in designing, coding and checking out operational software.

References

(1) Farrell, Quesinberry, Morgan, and Tom, "Dynamic Scaling for Air-to-Air Tracking," NAECON, Dayton, OH, 1975.

(2) Farrell and Quesinberry, "Track Mechanization Alternatives," NAECON, Dayton, OH 1981.

(3) Asseo and Ardila, "Sensor-Independent Target State Estimator Design and Evaluation," NAECON, Dayton, OH 1982.

(4) Bierman, FACTORIZATION METHODS FOR DISCRETE SEQUENTIAL ESTIMATION, Academic Press, 1977.

(5) Farrell, INTEGRATED AIRCRAFT NAVIGATION, Academic Press, 1976.

(6) Ramage and Lydick, "AFTI F-16 Automated Maneuvering Attack System - A Concept in Combat Automation," 36th AGARD Guidance and Control Symposium, Toulouse, France, May 1983.

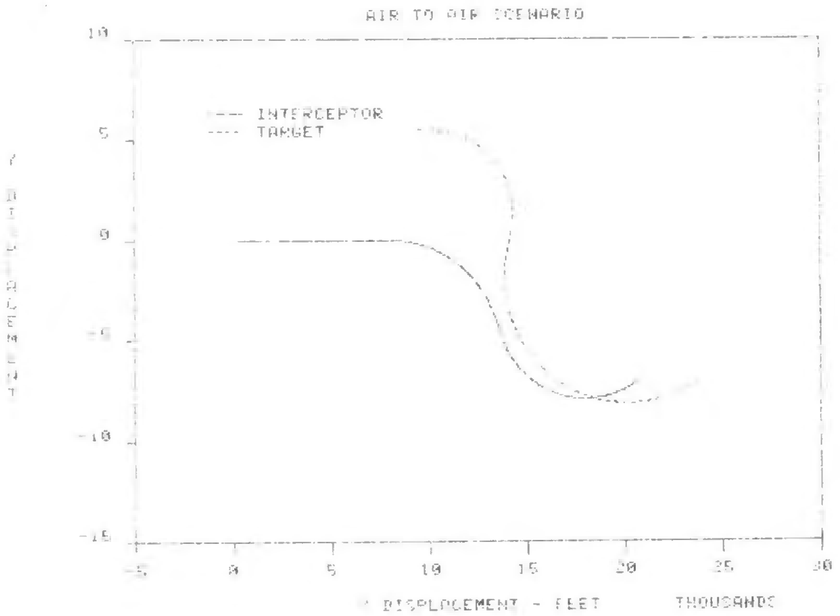


Figure 1. Mode 1 Scenario

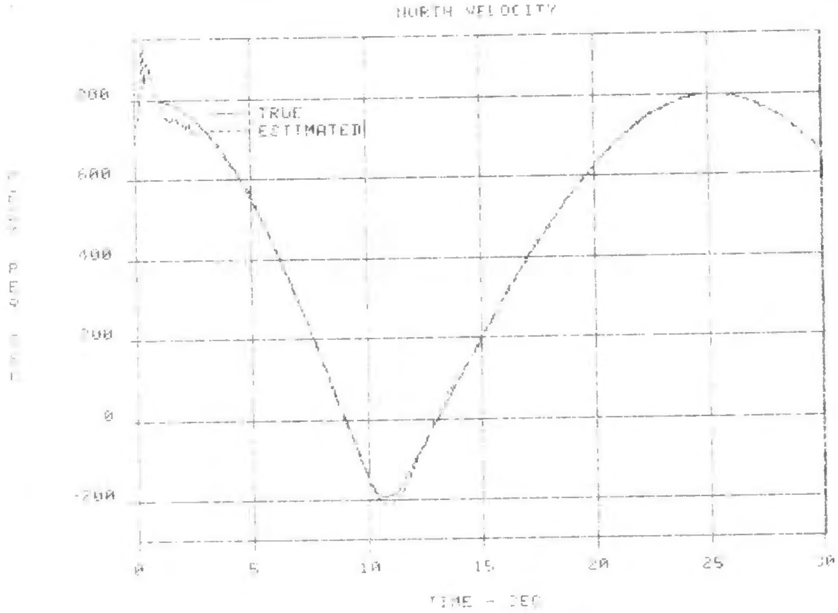


Figure 2. Mode 1 North Velocity History

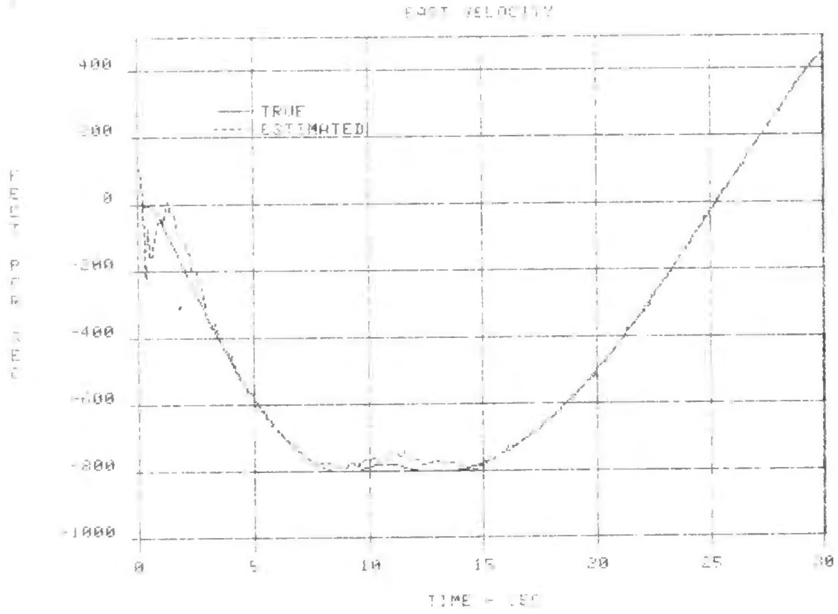


Figure 3. Mode 1 East Velocity History

Figure 4. Mode 1 Down-Velocity History

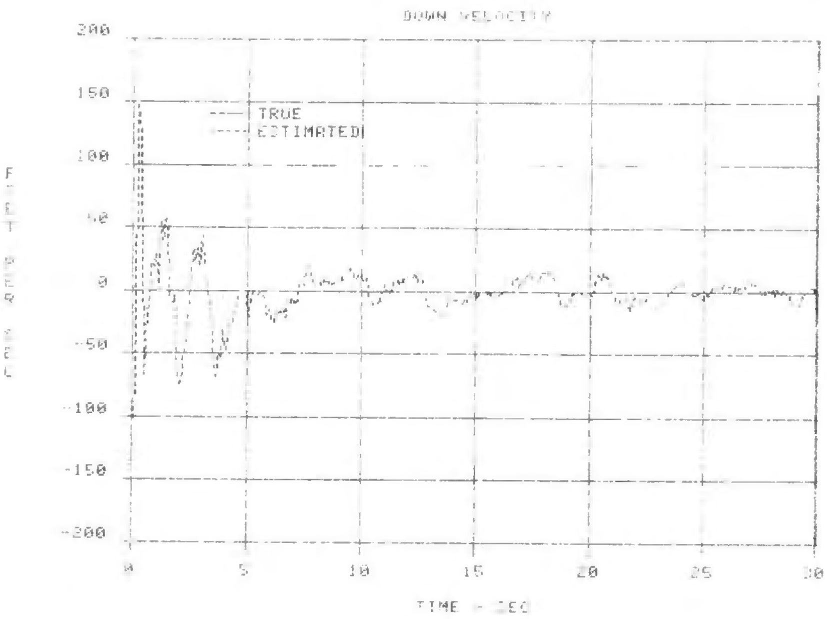


Figure 5. Mode 1 North Acceleration History

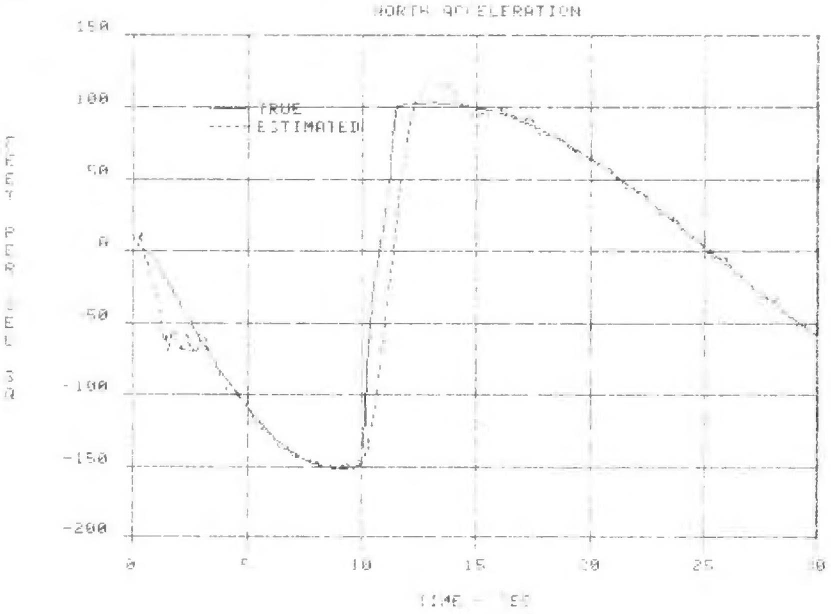
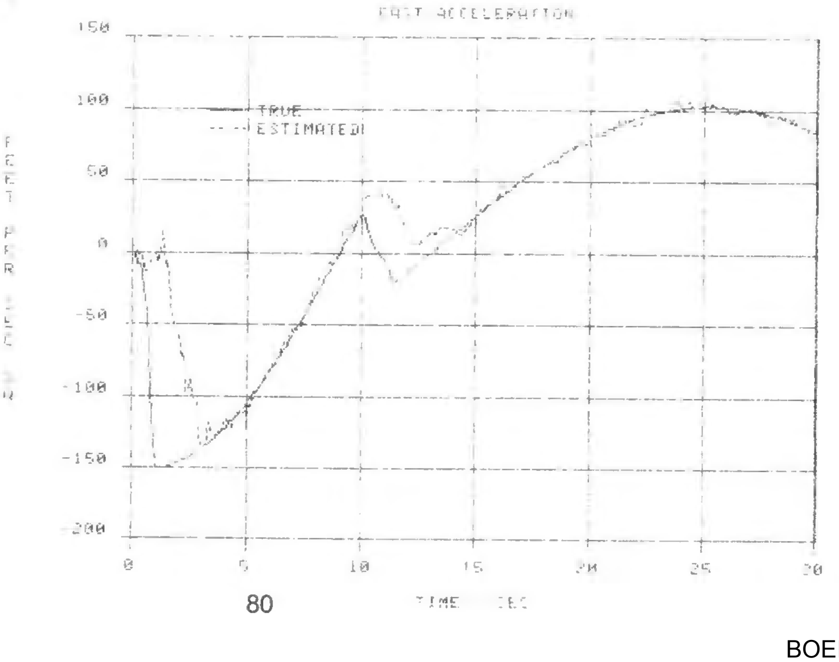


Figure 6. Mode 1 East Acceleration History


EAST ACCELERATION
TRUE
ESTIMATED
FEET PER SEC SQ
TIME - SEC