P-44

DOT/FAA/CT-86/32

FAA TECHNICAL CENTER
Atlantic City International Airport
New Jersey 08405

Analytical Sensor Redundancy Assessment

D.B. Mulcare
L.E. Downing
M.K. Smith

Lockheed-Georgia Company
A Division of Lockheed Corporation
Marietta, Georgia 30063

April 1988

Final Report

This document is available to the U.S. public through the National Technical Information Service, Springfield, Virginia 22161.

(NASA-CR-182892) ANALYTICAL SENSOR REDUNDANCY ASSESSMENT Final Report (Lockheed-Georgia Co.) 44 p CSCL 01D

N88-22901

Unclas
G3/06 0142548

U.S. Department of Transportation
Federal Aviation Administration

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
DOT/FAA/CT-86/32		

4. Title and Subtitle	5. Report Date
Analytical Sensor Redundancy Assessment	APRIL 1988
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
D. B. Mulcare, L. E. Downing, M. K. Smith	DOT/FAA/CT-86/32
9. Performing Organization Name and Address	10. Work Unit No.
Lockheed-Georgia Company Marietta, Georgia 30063	NAS2-11853
	11. Contract or Grant No.
12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
U.S. Department of Transportation Federal Aviation Administration Technical Center Atlantic City Airport, New Jersey 08405	Contractor Report
	14. Sponsoring Agency Code

15. Supplementary Notes

Point of Contact: W. E. Larsen/MS 210-2
Ames Research Center
Moffett Field, CA 94035

16. Abstract

This report describes the rationale and mechanization of sensor fault tolerance based on analytical redundancy principles. The concept involves the substitution of software procedures, such as an observer algorithm, to supplant additional hardware components. The observer synthesizes values of sensor states in lieu of their direct measurement. Such information can then be used, for example, to determine which of two disagreeing sensors is more correct, thus enhancing sensor fault survivability. Here a stability augmentation system is used as an example application, with required modifications being made to a quadruplex digital flight control system. The impact on software structure and the resultant revalidation effort are illustrated as well. Also, the use of an observer algorithm for wind gust filtering of the angle-of-attack sensor signal is presented.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Analytical Redundancy, Digital Flight Controls, Gust Filtering, Redundancy Management, Relaxed Static Stability, Sensor Fault Tolerance, System Re-validation	Unlimited Subject Category 38

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

*For sale by the National Technical Information Service, Springfield, Virginia 22161

FOREWORD

This report describes the rationale, mechanization, analysis, and testing of an analytical sensor redundancy modification to a double fail-operational Stability Augmentation System. As a task in the FAA-funded contract NAS2-11853, this effort focused on alleviating flying qualities degradation for a relaxed static stability airplane on a system level, and on program modifications and revalidation on a software level.

The intent of this task was to illustrate the potential benefits of software-implemented fault tolerance as applied to sensor hardware, and to investigate the overhead incurred in flight software and its airworthiness determination.

TABLE OF CONTENTS

TABLE OF CONTENTS (CONT'D)

LIST OF FIGURES

LIST OF TABLES

1.0 INTRODUCTION

An existing quadruplex digital flight control system (DFCS) was modified to incorporate analytical sensor redundancy for improved sensor hardware fault tolerance. Basically, this is a tradeoff of software functionality/overhead for a reduction in system components. Such a tradeoff may be justified in the case of a stability augmentation system (SAS) for a relaxed static stability (RSS) airplane. If the SAS function is critical, associated protective redundancy requires substantial sensor redundancy. In the case of air data and inertial sensors, a useful degree of redundancy can be obtained through an observer algorithm (Reference 2), rather than through further replication of components.

Basically, an observer relies on knowledge of the airplane's dynamic behavior and the flight control effector inputs, together with sensor signals as available, to mathematically reconstruct unmeasured or questionable sensor states. Such concepts call for new sensor voting schemes and reconfiguration logic, but the computational requirements are not very demanding. Perhaps the biggest problem for practical utility is that of ensuring that the observer always captures a workable representation of the airplane dynamics, for otherwise the estimated states will be skewed. In an actual DFCS, some degree of inaccuracy is permissible in the form of tolerances in the fault decision logic. Note also that additional software overhead required to periodically update the software model of the airplane dynamics.

Since here the example application of analytical sensor redundancy was retrofitted to an existing system, software modifications were designed to minimize the architectural/revalidation impact. This effort focused largely on the definition and control of interfaces of DFCS program units, and on a clear delineation and discernment of the functionality contained within the respective units.

1.1 Executive Summary

For an augmented fly-by-wire (AFBW) system, there is no way to predict or estimate what the pilots' input command transducers will or should be, but the sensors of the airframe dynamics are rather accurately anticipatable through workable knowledge of the airplane dynamics. The latter can therefore be captured in a mathematical model called an observer system (References 2 and 3), which is implemented in the DFCS computers to execute in real-time in parallel with the actual airplane motion. Such a model is depicted in Figure 1, where flight hardware sensor measurements are periodically compared with observer projections of the appropriate sensor values.

Analytical redundancy fault detection is achieved in a manner as shown in Figure 2. One or more of a given type airplane sensor is applied to a voter type fault detection logic along with the associated observer output estimate. The observer signal is then used in determining a faulted sensor, under the assumption that only a single fault event occurs. Conventional hardware sensor voting is normally employed until only two remaining sensors disagree. The observer therefore constitutes a tie-breaker in the event that two remaining sensors of a given type disagree. In a conventional DFCS, both would be discarded as untrustworthy.

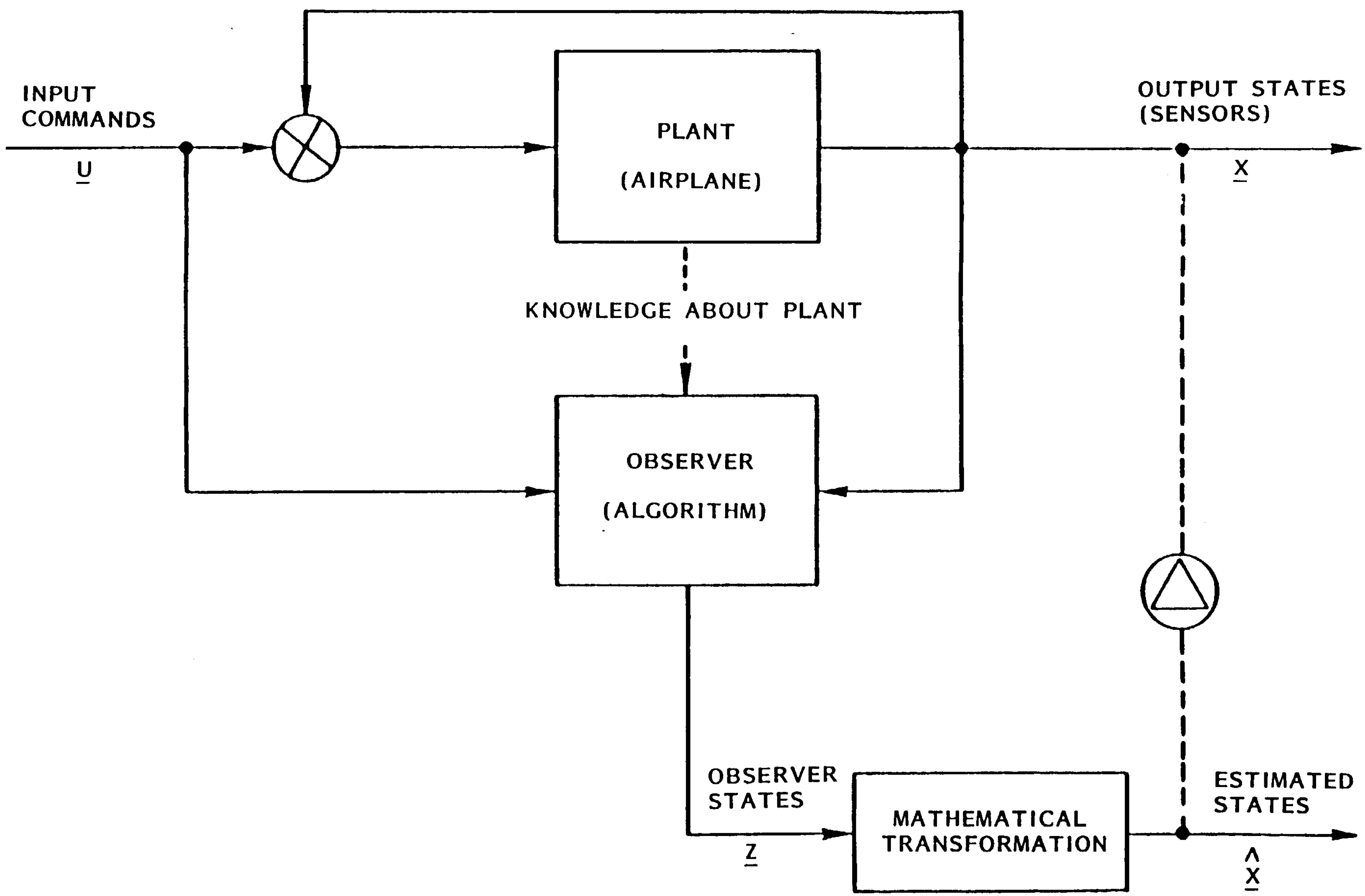


Figure 1. Analytical Redundancy Concept

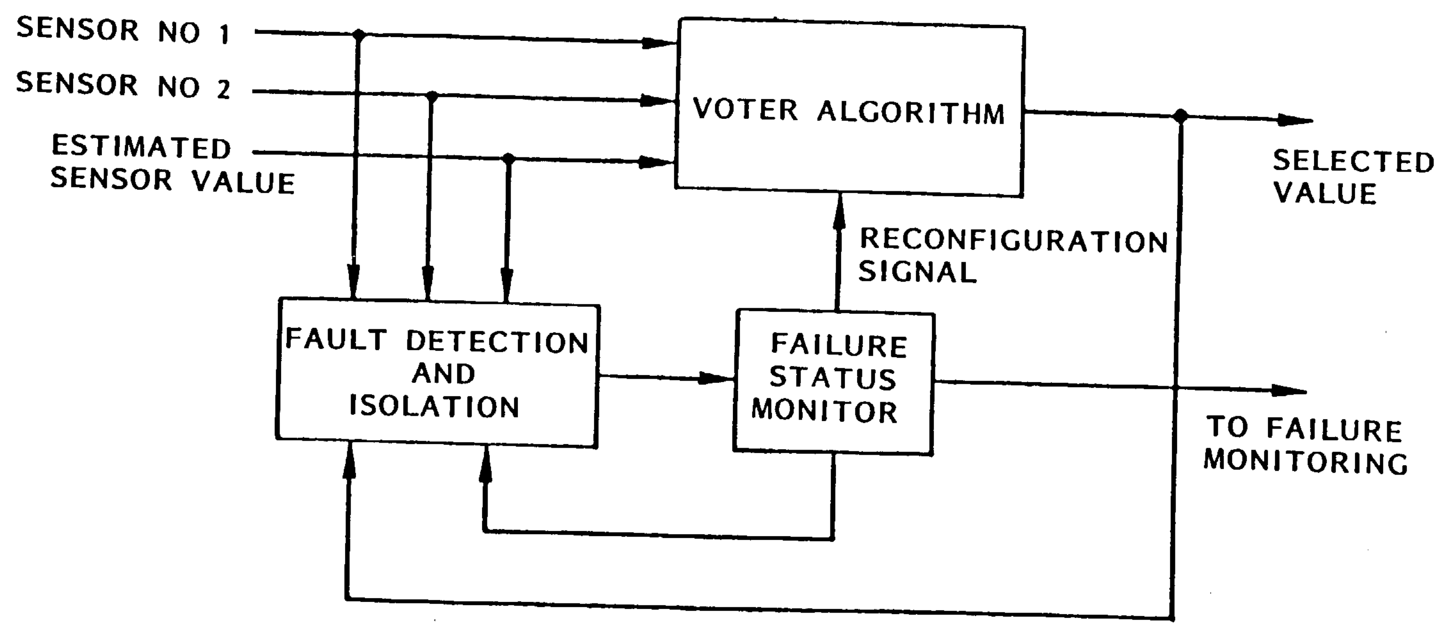


Figure 2. Analytical Sensor Fault Detection

In the case of a previously implemented DFCS at NASA Ames' Reconfigurable Digital Flight Control System (RDFCS) simulator (see Reference 1), the SAS function for an RSS airplane was crucially dependent on angle-of-attack (AOA) feedback to preclude a rapid pitch-axis divergence. Pitch rate was used for improved short-period flying qualities, but this was not critical to safe flight. Here the DFCS software was modified to incorporate an analytical redundancy function, and the resultant sensor fault tolerance was assessed from a reliability standpoint and demonstrated from a system failure effects standpoint. In both cases, general effectiveness was evident.

Clearly, analytical sensor redundancy can improve system reliability under given conditions, but the definitive issue is really life cycle costs. Do the economies of reducing line replaceable units (LRUs) on an aircraft warrant the additional software complexity and overhead? In the event they do, airworthiness concerns evoked by the expanded software c mplexity and the additional reliability assessments must be addressed. Such concerns are explicitly treated in the remainder of this report.

Much of the software design and implementation work under the sponsoring contract has been accomplished using the programming language Ada. Although originally oriented toward U.S. Department of Defense applications, Ada is now envisioned as a civil aviation standard language under ARINC auspices. This direction is in accord with the experience of the subject work in that the options and impact of Ada software modifications are clarified and well controlled through the use of Ada-based design. Although the RDFCS simulator necessitated the use of the AED (Algol Extended for Design) language for software implementation, Ada was still used for the associated design. Resultantly, Ada was supportive of the software redesign and revalidation test case definition.

As a by-product of the observer implementation, the basic algorithm was shown to be very effective in filtering wind gust components from the angle-of-attack (AOA) sensor signals. In cases where the gust component is not desired, the observer can effectively remove sensor noise which cannot be accomplished through linear filtering without significant loss of signal content. The instance of AOA- or airspeed-based autothrottles is an apt example. Here, the gust components tend to result in spurious and disconcerting throttle activity, which can be essentially eliminated by an observer. For ride quality improvement, however, atmospheric sensor noise would be valid signal content, for the DFCS would attempt to suppress the associated airplane motion.

1.2 Assessment Problem

Three main assessment tasks were undertaken: analytical redundancy design analysis, system reliability assessment, and test evaluation of the modified system mechanization. The design analysis was performed via Ada program unit examination, the reliability assessment using a substantially revised version of CARSRA (Computer-Aided Redundant System Reliability Analysis program (Reference 4), and the testing via non-realtime simulation. Real-time RDFCS simulator testing was precluded by flight computer problems at the facility, which persisted until its de-commissioning.

1.3 Relevance to Other Contractual Tasks

Two other tasks under the FAA-sponsored contract, NAS2-11853, were closely related. The aforementioned quadruplex DFCS tasks addressed a rather conventional double fail-operational DFCS for AFBW, so it reflected the customary hardware replication approach to hardware fault tolerance. An N-version flight software task (Reference 5) confronted the issue of the tolerance of software faults in a manner that complemented and extended the quadruplex design. This entailed appreciable software additions to the overall DFCS program, which constitute the cost of achieving another dimension of fault tolerance.

Like N-version software, the present task involves software-implemented fault tolerance. But here the focus is on the tolerance of sensor hardware faults. The major unifying aspect of the three tasks is the context of the same basic quadruplex system architecture, as reflected in the same high-level software design. This is rendered and modified as appropriate in Ada and graphics-type representations. Here the actual implementation of the DFCS software for the non-realtime was done in Ada as well.

1.4 Conclusions and Recommendations

The present example of analytical sensor redundancy was largely illustrative of the architectural and software aspects of realizing a practical DFCS application. Certain basic aspects, such as updating the observer system dynamics in accord with that of the airplane's, have not been addressed fully. Nonetheless, the viability of observer type algorithms has been indicated in some meaningful degree. In the absence of the originally intended DFCS system demonstration (because of the de-commissioning of the RDFCS facility), it is now possible to demonstrate the analytical sensor redundancy through non-realtime simulation. This could be instructive in a tutorial context, but the need and justification does not appear to exist. For the present, it seems sufficient to discern the basic concepts through reports such as this one, and to defer further depth of inquiry until specifics a full-scale implementation arise.

2.0 BACKGROUND

Observer algorithms are derived from a state-space formulation of a controls problem as described for example in Reference 6. Airplane dynamics are traditionally represented as a system of continuous-time, second-order ordinary differential equations (e.g., see Reference 7), and are solved on a digital computer by numerical integration. The corresponding state-space form is normally a system of first-order differential equations expressed in matrix form, as shown in Figure 3. This system of differential equations is usually converted to a corresponding set of first order difference equations to realize a discrete-time solution on a digital computer.

In Figure 3, the forcing functions are horizontal stabilizer, or flying tail, and engine throttles. These inputs are used to trim the airplane and to effect changes in airplane motions to achieve intended flight paths. The basic response of the airplane is reflected in the state vector $\underline{x}$, in state variables such as vertical velocity or pitch rate. Since certain states like vertical velocity are not readily measured or are not of much interest, the associated response is actually captured by the measurement vector $\underline{y}$, which comprises the variables actually measured by the airplane sensors. Here, true airspeed and AOA are called air data: pitch, pitch rate, and normal acceleration are termed inertial data.

As noted earlier, an observer algorithm in a DFCS computer must accurately reflect airplane dynamics, and it does so through recourse to a state-space representation. Observer development is presented in Section 4.0, but here it is appropriate as background to understand the generalities of modeling airplane dynamics. Note that the airplane dynamics are obviously continuous time phenomena, even if simulated in discrete time. Of necessity, the observer dynamics are implemented in discrete time in a DFCS, building upon the representation in Figure 3.

2.1 Abbreviations

AFBW	Augmented Fly-by-Wire
AOA	Angle-of-Attack
CG	Center-of-Gravity
DFCS	Digital Flight Control System
DOF	Degrees of Freedom
LRU	Line Replaceable Unit
RDFCS	Reconfigurable Digital Flight Control System
RSS	Relaxed Static Stability
SAS	Stability Augmentation System
WRT	With Respect To

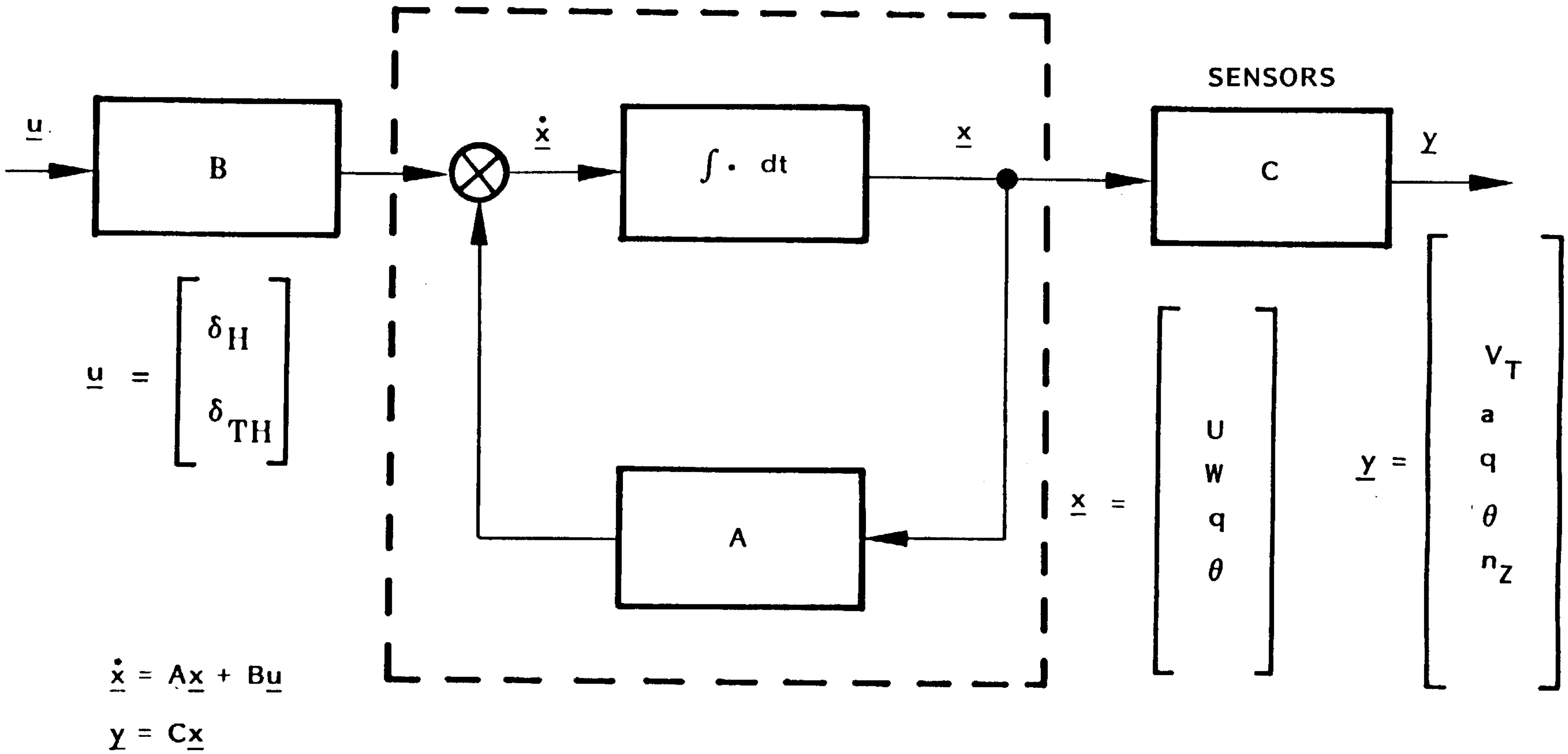


Figure 3. Airplane Dynamics Block Diagram (Sheet 1 of 2)

A --> (AIRFRAME) SYSTEM MATRIX

B --> FORCING FUNCTION MATRIX

C --> SENSOR MATRIX

$\underline{u}$ --> INPUT VECTOR

$\underline{x}$ --> AIRCRAFT STATE VECTOR

$\underline{y}$ --> MEASUREMENT VECTOR

δ_H --> HORIZONTAL STABILIZER DISPLACEMENT (ABOUT TRIM)

δ_{TH} --> INCREMENTAL THRUST (ABOUT TRIM)

U --> LONGITUDINAL-AXIS VELOCITY WITH RESPECT TO (WRT INCIDENT AIRSTREAM)

W --> VERTICAL-AXIS VELOCITY (WRT INCIDENT (AIRSTREAM)

q --> PITCH RATE

θ --> PITCH ATTITUDE

V_T --> TRUE AIRSPEED

a --> ANGLE-OF-ATTACK

n_Z --> VERTICAL-AXIS LOAD FACTOR (PER NORMAL ACCELEROMETER)

Figure 3. Airplane Dynamics Block Diagram (Sheet 2 of 2)

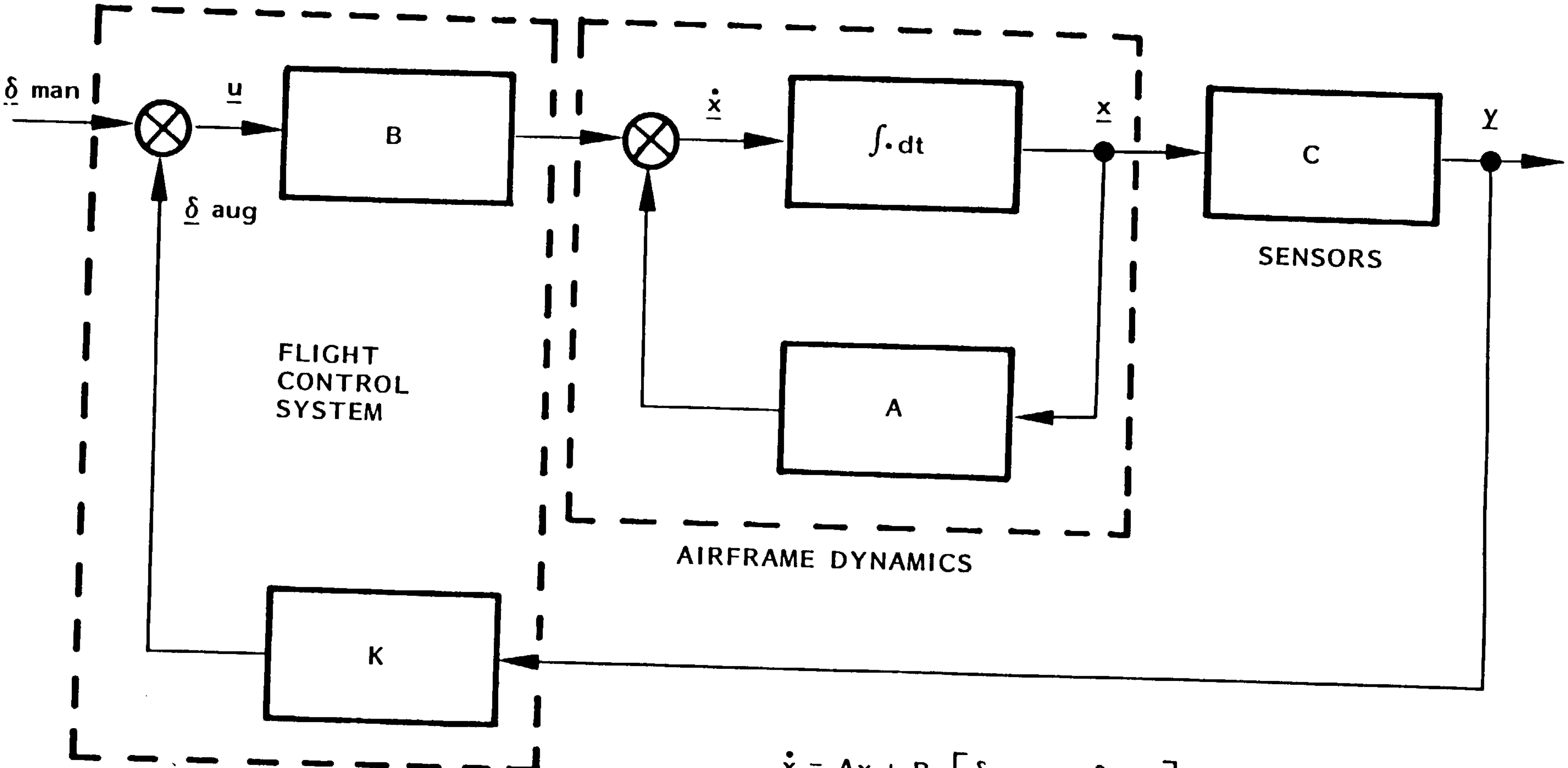
2.2 Managed Flying Qualities Degradation

With the increasing prevalence of RSS airplanes, pitch-axis flying qualities have become a more compelling concern because of the accompanying reduction or loss of inherent airframe stability. As an airplane's CG moves aft, the capacity of its wing to generate a nose down moment in response to an increase in AOA diminishes. At the neutral point, no pitching moment change results from an AOA change. If the CG is moved still farther aft, an increase in AOA yields a destabilizing pitch-up moment. This condition, which is referred to as a negative longitudinal stability margin, produces an absolute pitch-axis divergence. If at all rapid, such a divergence renders the airplane difficult or impossible to fly.

Consequently, a pitch stability augmentation system (SAS) function is normally added to the flight control system to restore airframe stability to adequate levels through active controls. Stability augmentation involves the feedback of inertial or air data sensor signals. This "inner loop" feedback improves the apparent stability of the airframe and provides acceptable flying qualities. A general pitch SAS mechanization is depicted in Figure 4, which is an expanded version of the pitch-axis state variable representation of the rigid-body equations of motion presented in Figure 3. The state variables are the elements of the vector $\underline{x}$, and the sensor measurement variables are the elements of the vector $\underline{y}$. In the latter case, true airspeed and AOA correspond to air data sensors, and pitch attitude, pitch rate, and incremental vertical acceleration constitute inertial sensors. The SAS feedback is routed through the gain matrix K. Intuitively, the poor flying qualities of the free, unaugmented RSS airplane that are inherent in the system matrix A, are offset by the augmentation provided by sensor feedback through K.

Since the SAS function is often necessary for safe readily controllable flight, over all or most of the flight regime, its continued proper operation must be ensured through redundant system components. In a redundant SAS architecture, there are generally two or more pitch-axis sensor signals fed back for the SAS function, e.g., AOA, normal acceleration, and pitch rate. So long as two of a given type of sensor are operating in agreement, that signal is available for the SAS computation. The differing nature of the various types of signals means that they make different contributions to flying qualities. Hence, the failure effects for the total loss of a signal type vary as well, but in general each loss results in some degradation of SAS performance. The intent of managed degradation is to minimize the degree as well as the frequency of performance degradation.

Analytical sensor redundancy supports a managed degradation strategy in that it permits at least one additional LRU failure of a given type before total loss of the corresponding signal. This can be accomplished through the use of the observer algorithm to determine which of two disagreeing sensors is discrepant. Then the operable sensor can be kept on line. Otherwise, the onset of flying qualities degradation, and perhaps loss of a critical function, would occur more often. Although analytical sensor redundancy is possible as a DFCS software add-on, the overall system architecture and the fault degradation profile need to be designed in a comprehensive manner.



$\underline{\dot{x}} = A\underline{x} + B\left[\underline{\delta}\ \text{man} + \underline{\delta}\ \text{aug}\right]$

K -- AUGMENTATION GAIN MATRIX

$\underline{\delta}$ man PILOT'S MANUAL COMMANDS

$\underline{\delta}$ aug STABILITY AUGMENTATION COMMANDS

Figure 4. Stability Augmentation System Block Diagram

2.3 Sensor Fault Tolerance Concepts

Sensor fault tolerance is the accommodation of hardware faults which is usually based on some type of comparator scheme. One typical example is provided by the companion investigation on N-version programming (Reference 5). Here two to four sensors were voted using a low median select strategy wherein all presumably normal voter inputs were compared with the selected signal. Essentially the same scheme is depicted for triplex pitch rate gyro sensors in Figure 5. Due to admissible sensor variations, both time and amplitude thresholds are employed by the comparators to eliminate nuisance trips. As noted earlier, this scheme is unable to discriminate which of just two disagreeing sensors is correct, unless other information exists to support the decision. This might be provided by a sensor validity signal, which may not be available or dependable.

More encompassing supplementary fault decision information can be furnished by analytical redundancy mechanisms, as previously reported in Reference 6. There are many possible variations, particularly with respect to fault decision logic. In Figure 2, a conventional comparator mechanization was modified to accept an analytically synthesized sensor signal into an associated software voter. The logic could be set up such that the synthesized signal would be invoked only when needed to discriminate a disagreeing pair of sensors. Note that the illustrative calculations in Section 4.0 indicate worthwhile improvements in system reliability where a triplex set of sensors is flight critical.

2.4 Observer Theory

The reconstruction of a non-measured signal can be achieved using Kalman filtering or an observer algorithm. The latter is simpler and quite satisfactory for airplane inertial and air data sensors. Observer algorithm theory per se was developed in References 1 and 2, based on a mathematical modeling characterization of the dynamic behavior of a deterministic physical system and its forcing functions. For an airplane, this relates to airplane motion in response to flight controller inputs, e.g., elevator surface displacement, as evident in measurements of dynamic states, e.g., pitch rate. Where a measured variable is not state variable, the associated transformation must exist and be known from measurements to all states. In particular, the states must be observable in a mathematical sense (e.g., see Reference 6), as is normally the case.

The foregoing relationships are best indicated through visualizing the two separate systems shown in Figure 6. The rigid body airplane dynamics are the same as presented in Figures 3 and 4. The second system is the observer, described here as a system of first-order differential equations similar to those descriptive of the airplane dynamics. The observer system, however, exists only as corresponding difference equations in computer software. Note that its system matrix is D, and that it has two forcing functions, the sensor measurement vector $\underline{y}$ as well as the flight controller vector $\underline{u}$. The matrices D, E, and F of the observer system are derived from the airplane's behavior as captured in matrices A, B, and C. Hence, the observer in general contains the essential information about the airplane's behavior to reconstruct unmeasurable states, e.g., those whose sensors are inoperable.

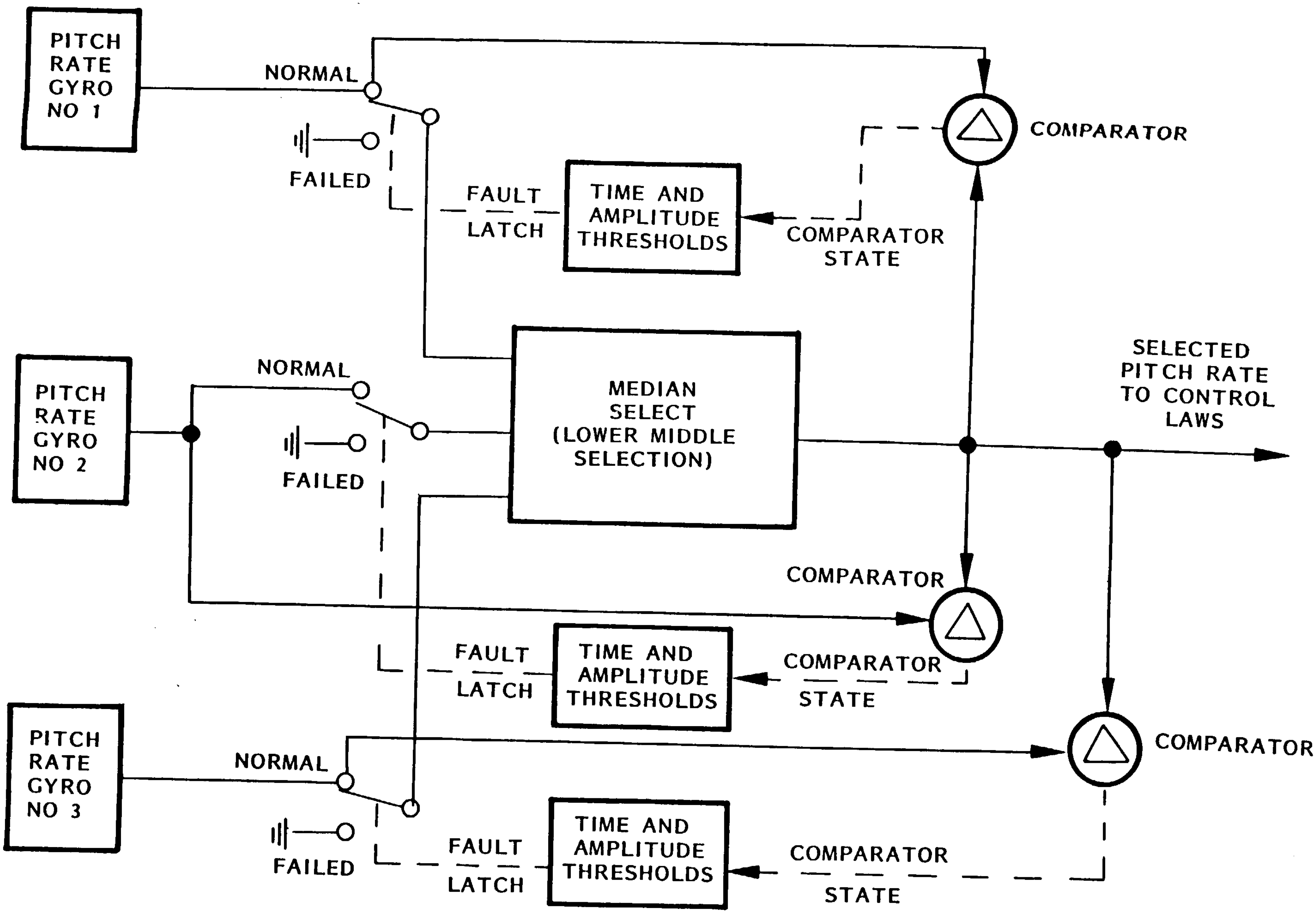


Figure 5. Triplex Sensor VOTER/Comparator

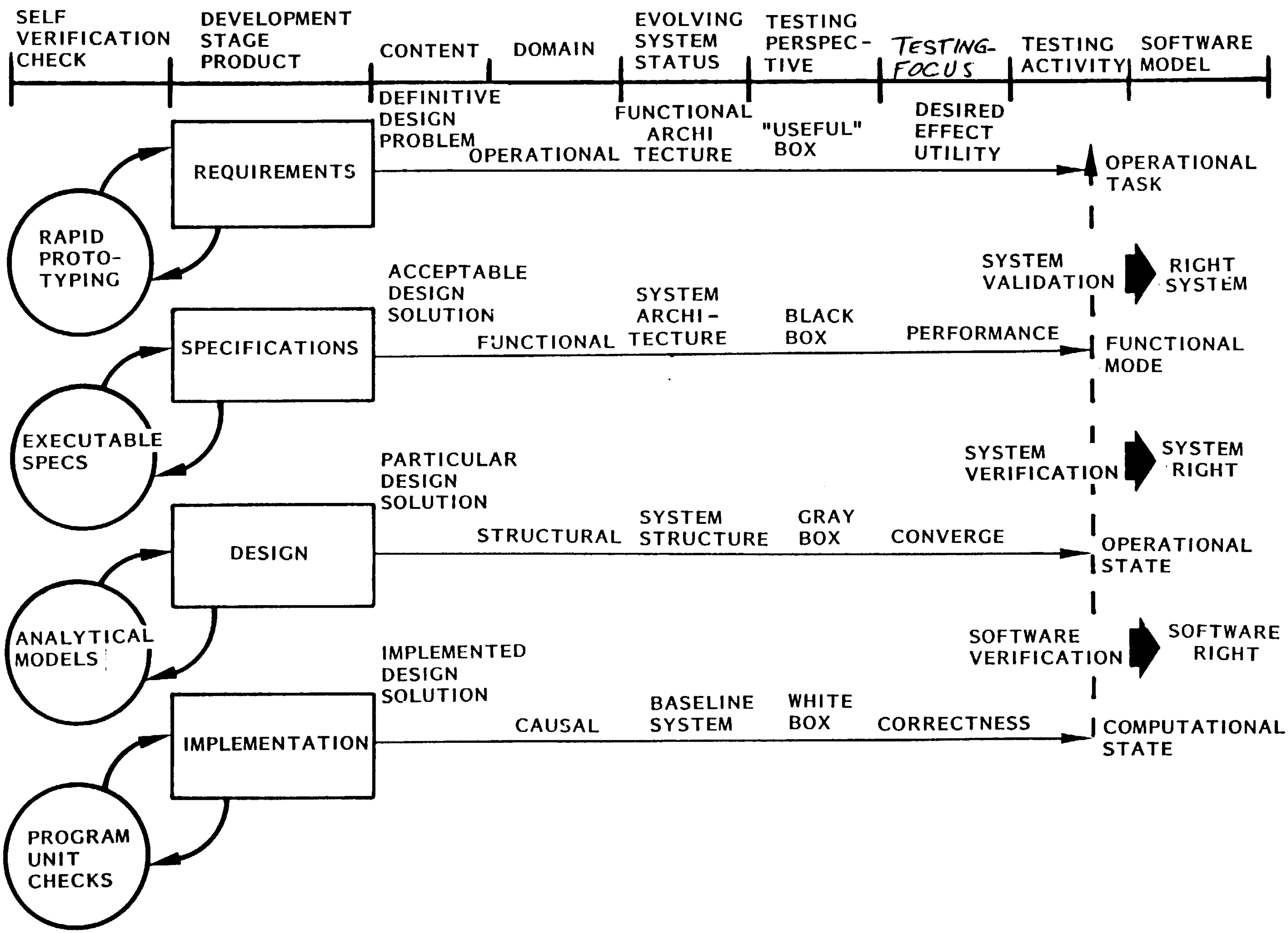


Figure 6. Multi-Level Testing Closure

For the observer derivation shown in Figure 6, there is a one-to-one correspondence between airplane states $\underline{x}$ and observer states $\underline{z}$. Thus, the observer states need only be converted using the measurement matrix C to enable a direct cross-comparison with the various sensor signals. Ideally, the the observer system can function without the measurement vector $\underline{y}$ feedback, but uncertainties regarding the actual airplane dynamics compromise such a modeling approach. Sensor feedback driving the observer, therefore, captures actual state coupling inherent in the actual airplane dynamics.

Of course, as the airplane traverses a given flight profile, its dynamic behavior will vary quite appreciably. This is reflected in the elements of matrices A, B, and C. It is therefore necessary to update the observer system matrices, D, E, and F, over a corresponding sequence of airplane trim points or operating points. Otherwise, the sensor comparison becomes invalid. The observer system updating also involves discretization of the observer dynamics to permit digital computation. Normally, this is a background mode computation that entails non-negligible overhead.

2.5 Software Modifications and Revalidation

Here the analytical redundancy was added to the existing quadruplex DFCS (Reference 1), with the predominant impact, affecting the flight software. The need then is to determine the scope of software modifications, and in turn, the focus of the revalidation effort. The latter calls for multilevel testing, and hence, analytical test case definition. At the system level, sensor failure effects testing is ostensibly the same as for strictly replicated hardware redundancy. At the lower levels of testing, observations are directed toward confirming consistency with the verified system/software re-design. With properly selected test cases, such consistency assurances strengthen the confidence that the modified DFCS mechanization can cope satisfactorily with conditions and inputs not actually applied during testing.

Basically, exhaustive testing of software is not possible, even with automated testing, so the best use must be made of test time and resources. The revalidation test case definition must emphasize new requirements, modified interfaces, and new or altered software. It must confirm that the rules or assumptions used in reliability assessment are valid. For overall, general confidence in the modified DFCS, some regression testing must be performed as well to ensure that unintended changes have not inadvertently occurred, especially those outside the defined scope of modifications.

2.6 Testing Activities

Figure 7 summarizes the types and orientations of testing activities that originate in various stages of software development. The development results associated with each of these stages contribute to the definition of the multilevel test cases as implemented functions (e.g., see Reference 7). These respective testing contributions can be related to the analytical sensor redundancy modifications to indicate the nature and scope of testing necessary. All the levels of testing can to some extent be accomplished during real-time system simulation, as is described here. But this assumes that extensive low-level software testing and flight software load module integration have already been accomplished.

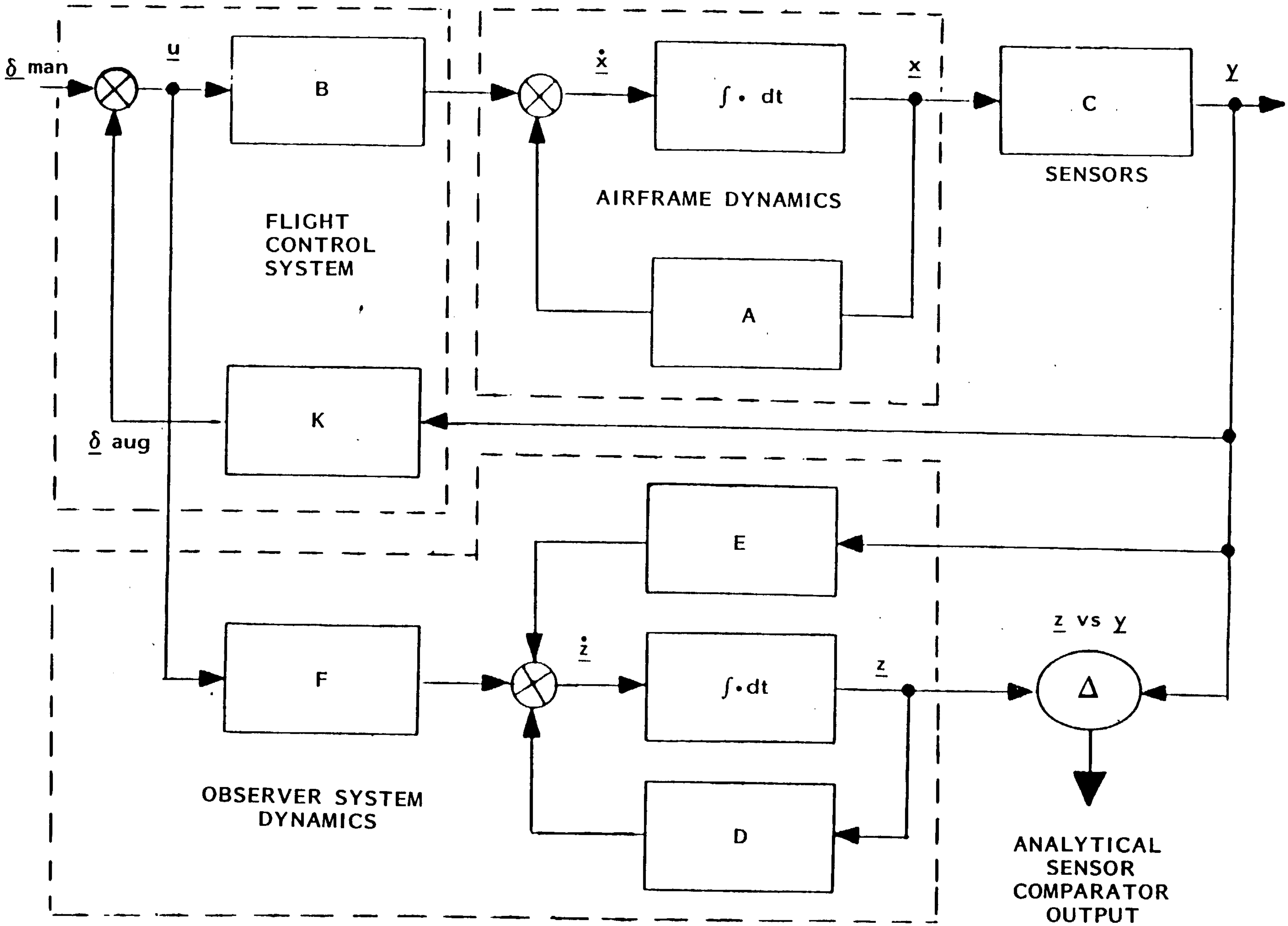


Figure 7. Observer System Sensor Fault Detection (Sheet 1 of 2)

D --> OBSERVER SYSTEM MATRIX

E --> SENSOR FORCING FUNCTION MATRIX

F --> FLIGHT CONTROLS FORCING FUNCTION MATRIX

$\underline{z}$ --> OBSERVER STATE VECTOR

OBSERVER SYSTEM FORM: $\dot{\underline{z}} = D\underline{z} + E\underline{y} + F\underline{u}$

SUBSTITUTE OBSERVER ASSUMPTION: $\underline{z} = \underline{y}$; $\dot{\underline{z}} = \dot{\underline{y}}$

--> $\dot{\underline{y}} = D\underline{y} + E\underline{y} + F\underline{u}$

NEXT, SUBSTITUTING: $\underline{y} = C\underline{x}$ & $\dot{\underline{y}} = C\dot{\underline{x}}$

--> $C\dot{\underline{x}} = (D + E)C\underline{x} + F\underline{u}$

PRE-MULTIPLYING THE AIRFRAME DYNAMICS EQUATION BY C

--> $C\dot{\underline{x}} = CA\underline{x} + CB\underline{u}$

EQUATING THE COEFFICIENTS OF $\underline{x}$ and $\underline{u}$ IN THE LAST TWO EQUATIONS:

--> $F = CB$ & $CA = (D + E)C$

NOTE THAT MATRICES A, B, AND C ARE KNOWN A PRIORI, AND HENCE MATRIX F CAN BE SOLVED FOR DIRECTLY. MATRICES D AND E ARE NOT DEFINED AT THIS POINT, BUT THERE IS ONLY ONE EQUATION FOR THESE TWO UNKNOWNS. HENCE, D IS SOMEWHAT ARBITRARILY SELECTED TO BE A DIAGONAL MATRIX, AND IN TURN, E IS SELECTED TO BE A MATRIX WITH ALL MAIN DIAGONAL ELEMENTS SET TO ZERO.

Figure 7. Observer System Sensor Fault Detection (Sheet 2 of 2)

The addition of analytical sensor redundancy to an existing DFCS would normally involve system specification changes, but in all likelihood the system requirements would not be changed. In turn, the system/software design and the software implementation would necessarily be modified. The software re-design would reflect structural program changes in the form of both added and modified software units, with accompanying interface changes. Although some source code changes would appear in the program structure, most would occur within the new or modified software units. Ada program unit specifications aid in highlighting the scope of such changes, and explicitly define the revised interfaces.

Multilevel testing then would focus primarily on the changes purposefully introduced into the DFCS software. These would include performance, coverage, and correctness oriented test case definition strategies to examine the functional, structural, and causal aspects of the modified software. These aspects of testing are expanded in Table 1 relative to the analytical sensor redundancy modification. Obviously, considerable sensor fault cases are necessary to exercise both attendant logic and the control algorithms. regression testing would address other DFCS modes and functions outside the scope of the intended modifications, and would include for example some requirements oriented testing. The purpose would be to ensure that no side effects or inadvertent changes had been introduced.

TABLE 1 - MULTILEVEL TEST CASE EXAMPLES

DEVELOPMENT STAGE LEVEL	TESTING FOCUS	TYPE OF TESTING	DFCS FEATURES OF INTEREST	TYPE OF OBSERVATIONS
Requirements	General Utility	Regression	System Operation	Overall Acceptability
Specification	Explicit Design Requirements	Faulted Performance	Managed Flying Degradation	Pitch SAS Performance
Design	Software Design Features	Structural, Path & Interfaces	Mode Switching, Status & Timing	Path Traversal & Function Invocation
Implementation	Source Code Behavior	Control & Logic Functions	Control Laws, Deriving Observer & DFCS Logic	Logic, Count & Control Variables

3.0 OBJECTIVES

As with the other tasks under the sponsoring contract, the general objective of this task was to explore certain aspects of fault tolerant DFCSs that might be employed where the performance of critical functions must be ensured despite multiple faults. In particular, the emphasis here was on sensor faults for the pitch SAS function. The intent has been to illustrate the mechanization of analytical redundancy in extending sensor fault tolerance. Additionally, issues of re-design and revalidation have been introduced because the analytical redundancy features were added to an existing DFCS configuration.

3.1 Goal

The goal of this task has been to motivate, illustrate, and critique a representative example of analytical sensor redundancy similar to that which might appear in a DFCS submitted for airworthiness certification. It is intended that certain pivotal issues and viable approaches to enhanced or more economical system reliability will thereby be exemplified. More specifically, the mechanics and efficacy of observer algorithms will hopefully be shown to be plausible and a worthwhile usage of digital processing capacity.

3.2 Task Objectives

The specific objectives of this task are threefold:

- o to explain the operation of an observer algorithm as used for DFCS analytical sensor redundancy
- o to indicate the system architecture tradeoffs in terms of system reliability and implementation overhead
- o to illustrate the revalidation process attendant to a major DFCS modification.

3.3 Scope

As with several other tasks, the analytical sensor redundancy effort has been limited to part of a typical DFCS pitch axis. It is felt that the basic principles of sensor fault tolerance can be better understood by a suitably bounded application example. Also, a straightforward observer algorithm is used rather than a much more complicated Kalman filter. Additionally, not all of the software needed for updating the observer dynamics has been included.

PRECEDING PAGE BLANK NOT FILMED

4.0 ASSESSMENT RESULTS

A system architecture was selected based on an analytical sensor redundancy modification to the previously developed quadruplex DFCS (Reference 1). The new configuration was based on minimizing the impact on the existing DFCS implementation, while rendering a safe yet more economical design. Although the system simulator implementation at NASA Ames was programmed using Rockwell-Collins' augmented AED (Algol Extended for Design) instruction set, the software design was in Ada for improved clarity.

4.1 Analytical Redundancy Concepts

In addition to the requirement for observability of airplane states, Sheet 2 of Figure 6 noted two matrix equations whose was satisfaction was necessary in the development of the particular observer used here. The concluding statement there also indicated that Matrix D was to be a diagonal matrix and in juxtaposition, Matrix E was to have all zero elements on the main diagonal.

The D-Matrix choice was based on two aspects of simplifying observer system updating during flight: term-by-term exponentiation along the main diagonal, rather than matrix exponentiation; and similarly, term-by-term inversion along the main diagonal, rather than matrix inversion. The associated choice of zeroing the main diagonal of the E-Matrix means that no observer state is a direct function of its measured value, which after all may be erroneous or in question.

The expanded derivation and general result of this development in the form of the observer matrices definitions are presented in Figure 8. Here the variables are as defined in Figure 6. Note that certain reasonable simplifications have been made, e.g., the longitudinal airplane equations of motion have been reduced from six to four degrees-of-freedom. The forms of the Matrices A, B, and C were dictated, respectively, by: general airplane dynamics, the particular flight control effectors, and the selected complement of sensors. These are known quantities, expressed as variables, for purposes of defining the observer matrices.

Matrix-F is readily defined element-by-element in terms of Matrices B and C as noted at the bottom of Sheet 1 of Figure 8. Matrices D and E then take three sheets to develop under the above stated constraints. On Sheet 4, 30 matrix coefficients are then expressed in terms of 25 simple equations. Accordingly, five variables are parameters, selected here to ensure no zero values for this set of matrix elements. Of course, this computation was automated, as was the discretization algorithm to transform these continuous-time observer matrices into those for the corresponding discrete-time system, as preferred for digital computation.

4.2 Reliability Assessment

Analytical redundancy, in the subject case at least, was not invoked until only two of a given type of sensor remain on-line. Then the observer served as a tie-breaker in the event that the two sensors disagree. Hence, the reliability assessment here is in terms of dual sensors, in the context of system reliability improvements provided by the observer-based fault

I. Airframe Equations

From the Airplane Equations of Motion:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & 0 \\ 0 & 0 & a_{43} & 0 \end{bmatrix} \qquad [4x4]$$

$$B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & 0 \\ b_{31} & b_{32} \\ 0 & 0 \end{bmatrix} \qquad [4x2]$$

For the Selected Sensor Set:

$$C = \begin{bmatrix} c_{11} & 0 & 0 & 0 \\ 0 & c_{22} & 0 & 0 \\ 0 & 0 & c_{33} & 0 \\ 0 & 0 & 0 & c_{44} \\ 0 & c_{52} & 0 & 0 \end{bmatrix} \qquad [5x4]$$

II. Observer Equations

For the Given Observer Deviation:

$$F = CB = \begin{bmatrix} c_{11} & 0 & 0 & 0 \\ 0 & c_{22} & 0 & 0 \\ 0 & 0 & c_{33} & 0 \\ 0 & 0 & 0 & c_{44} \\ 0 & c_{52} & 0 & 0 \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & 0 \\ b_{31} & b_{32} \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} b_{11}c_{11} & b_{12}c_{11} \\ b_{21}c_{22} & 0 \\ b_{31}c_{33} & b_{32}c_{33} \\ 0 & 0 \\ b_{21}c_{52} & 0 \end{bmatrix} \qquad [5x2]$$

Figure 8 - Observers Matrix Determination (Sheet 1 of 4)

$$D = \begin{bmatrix} d_{11} & 0 & 0 & 0 & 0 \\ 0 & d_{22} & 0 & 0 & 0 \\ 0 & 0 & d_{33} & 0 & 0 \\ 0 & 0 & 0 & d_{44} & 0 \\ 0 & 0 & 0 & 0 & d_{55} \end{bmatrix} \qquad [5x5]$$

$$E = \begin{bmatrix} 0 & e_{12} & e_{13} & e_{14} & e_{15} \\ e_{21} & 0 & e_{23} & e_{24} & e_{25} \\ e_{31} & e_{32} & 0 & e_{34} & e_{35} \\ e_{41} & e_{42} & e_{43} & 0 & e_{45} \\ e_{51} & e_{52} & e_{53} & e_{54} & 0 \end{bmatrix} \qquad [5x5]$$

$$\boxed{CA = (D + E)C}$$

$$CA = \begin{bmatrix} c_{11} & 0 & 0 & 0 \\ 0 & c_{22} & 0 & 0 \\ 0 & 0 & c_{33} & 0 \\ 0 & 0 & 0 & c_{44} \\ 0 & c_{52} & 0 & 0 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & 0 \\ 0 & 0 & a_{43} & 0 \end{bmatrix} = \begin{bmatrix} a_{11}c_{11} & a_{12}c_{11} & a_{13}c_{11} & a_{14}c_{11} \\ a_{21}c_{22} & a_{22}c_{22} & a_{23}c_{22} & a_{24}c_{22} \\ a_{31}c_{33} & a_{32}c_{33} & a_{33}c_{33} & 0 \\ 0 & 0 & a_{43}c_{44} & 0 \\ a_{21}c_{52} & a_{22}c_{52} & a_{23}c_{52} & a_{24}c_{52} \end{bmatrix} \qquad [5x4]$$

$$(D+E)C = \begin{bmatrix} d_{11} & e_{12} & e_{13} & e_{14} & e_{15} \\ e_{21} & d_{22} & e_{23} & e_{24} & e_{25} \\ e_{31} & e_{32} & d_{33} & e_{34} & e_{35} \\ e_{41} & e_{42} & e_{43} & d_{44} & e^{4}5 \\ e_{51} & e_{52} & e_{53} & e_{54} & d_{55} \end{bmatrix} \begin{bmatrix} c_{11} & 0 & 0 & 0 \\ 0 & c_{22} & 0 & 0 \\ 0 & 0 & c_{33} & \\ 0 & 0 & 0 & c_{44} \\ 0 & c_{52} & 0 & 0 \end{bmatrix}$$

Figure 8 - Observers Matrix Determination (Sheet 2 of 4)

$$(D+E)C= \begin{bmatrix} c_{11}d_{11} & (C_{22}e_{12}+C_{55}e_{15}) & c_{33}e_{13} & c_{44}e_{14} \\ c_{11}e_{21} & (c_{22}d_{22}+c_{52}e_{25}) & c_{33}e_{23} & c_{44}e_{24} \\ c_{11}e_{31} & (c_{22}e_{32}+c_{52}e_{35}) & c_{33}d_{33} & c_{44}e_{34} \\ c_{11}e_{41} & (c_{22}e_{42}+c_{52}e_{45}) & c_{33}e_{43} & c_{44}d_{44} \\ c_{11}e_{51} & (c_{22}e_{52}+e_{52}d_{55}) & c_{33}e_{53} & c_{44}e_{54} \\ c_{11}e_{41} & (c_{22}e_{42}+c_{52}e_{45}) & c_{33}e_{43} & c_{44}d_{44} \\ c_{11}e_{51} & (c_{22}e_{52}+c_{52}d_{55}) & c_{33}e_{53} & c_{44}e_{54} \end{bmatrix} \begin{bmatrix} a_{11}c_{11} & a_{12}c_{11} & a_{13}c_{11} & a_{14}c_{11} \\ a_{21}c_{22} & a_{22}c_{22} & a_{23}c_{22} & a_{24}c_{22} \\ a_{31}c_{33} & a_{32}c_{33} & a_{33}c_{33} & 0 \\ 0 & 0 & a_{43}c_{44} & 0 \\ a_{21}c_{52} & a_{22}c_{52} & a_{23}c_{52} & a_{24}c_{52} \\ 0 & 0 & a_{43}c_{44} & 0 \\ a_{21}c_{52} & a_{22}c_{52} & a_{23}c_{52} & a_{24}c_{52} \end{bmatrix}$$

Observer Matrix Element Solutions:

Assign All Non-Zero

$$\begin{array}{ll} d_{11} = a_{11} & \\ e_{21} = a_{21}c_{22}/c_{11} & \\ e_{31} = a_{31}c_{33}/c_{11} & \\ e_{41} = 0 & \\ e_{51} = a_{21}c_{52}/c_{11} & \end{array} \qquad \left.\begin{array}{ll} e_{12} = (a_{12}c_{11}-c_{52}e_{15})/c_{22} & ;\ \text{Let } e_{15} = \\ d_{22} = (a_{22}c_{22}-c_{52}e_{25})/c_{22} & ;\ \text{Let } e_{25} = \\ e_{32} = (a_{32}c_{33}-c_{52}e_{35})/c_{22} & ;\ \text{Let } e_{35} = \\ e_{42} = -c_{52}e_{45}/c_{22} & ;\ \text{Let } e_{45} = \\ e_{52} = (a_{22}c_{52}-c_{52}d_{55})/c_{22} & ;\ \text{Let } d_{55} = \end{array}\right\} \begin{array}{l}\text{Parameters}\\ \text{Selection}\end{array}$$

$$\begin{array}{ll} e_{13} = a_{13}c_{11}/c_{33} & e_{14} = a_{14}c_{11}/c_{44} \\ e_{23} = a_{23}c_{22}/c_{33} & e_{24} = a_{24}c_{22}/c_{44} \\ d_{33} = a_{33} & e_{34} = 0 \\ e_{43} = a_{43}c_{44}/c_{33} & d_{44} = 0 \\ e_{53} = a_{23}c_{52}/c_{33} & e_{54} = a_{24}c_{52}/c_{44} \end{array}$$

Figure 8 - Observers Matrix Determination (Sheet 3 of 4)

III. Particular Assignments

Simplifications per the Present Case:

$$F = \begin{bmatrix} b_{11}c_{11} & 0 \\ b_{21}c_{22} & 0 \\ b_{31}c_{33} & 0 \\ 0 & 0 \\ b_{21}c_{52} & 0 \end{bmatrix} \qquad \text{For Throttles Fixed}$$

$$c_{11} = c_{33} = c_{44} = 1 \qquad \text{For Our Case}$$

$$\left.\begin{array}{lllll}
d_{11} = a_{11} & e_{12} = (a_{12} - c_{52}e_{15})/c_{22} & e_{13} = a_{13} & e_{14} = a_{14} & e_{15} = \\
e_{21} = a_{21}c_{22} & d_{22} = (a_{22} - c_{52}e_{25})/c_{22} & e_{23} = a_{23}c_{22} & e_{24} = a_{24} & e_{25} = \\
e_{31} = a_{31} & e_{32} = (a_{32} - c_{52}e_{35})/c_{22} & d_{33} = a_{33} & e_{34} = 0 & e_{35} = \\
e_{41} = 0 & c_{42} = -c_{52}e_{45}/c_{22} & e_{43} = a_{43} & d_{44} = 0 & e_{45} = \\
e_{51} = a_{21}c_{52} & e_{52} = (a_{22}c_{52} - c_{52}d_{55})/c_{22} & e_{54} = a_{23}c_{52} & e_{54} = a_{24}c_{52} & d_{55} =
\end{array}\right\} \text{Parameters Selection}$$

Figure 8 - Observers Matrix Determination (Sheet 4 of 4)

isolation. Figure 9 then illustrates three variants of a dual system, where contrasts are to be made regarding the contributions of self test/analytical redundancy combinations.

In Sheet 1 of Figure 9, faulting of certain components are assumed to be obvious, e.g., non-engagable servo, and this is represented by parallel boxes. Note that paralleling denotes logical OR-ing to attain a operable system, and serial boxes denote logical AND-ing. As evident in Sheet 2, self test increases the extent of parallelism, which translates into improved system reliability. Sheet 3 reflects the contributions of analytical redundancy in that pairs of sensors are not linked in serial fashion, i.e., both sensors need not be working to have a usable signal output.

Quantitative assessment results for these three configurations are represented in Table 2. The fault isolation capacity of self test is seen to only modestly improve system reliability, in part because it does not affect sensor aspects of reliability. Analytical redundancy, however, does improve reliability quite significantly (by three orders of magnitude), as the results show. Of course, the benefits of analytical redundancy would be much less for triplex sensor, because it would be involved less often. Note that the combination of self test and analytical redundancy are complementary, for they are suited for different types of fault isolation. Their combination is therefore natural and very beneficial.

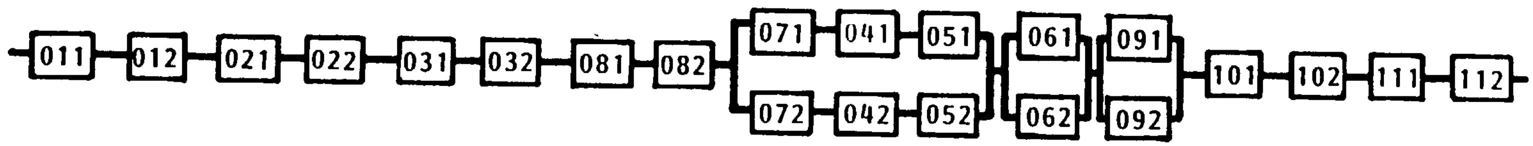
4.3 Simulation Investigation

Simulation time histories of observer states versus sensor measurements states are given in Figure 10. Even under multiple sensor signal losses, the observer outputs match well with those of the sensors. This is not surprising for such a short duration in which that airplane dynamics have been represented exactly, and where re-trimming or other airplane changes, e.g., fuel burn-off, have not occurred. In practice, such concerns can be handled using additional flight software, as well as normal sensor signal tolerances.

4.4 Observer Gust Filtering

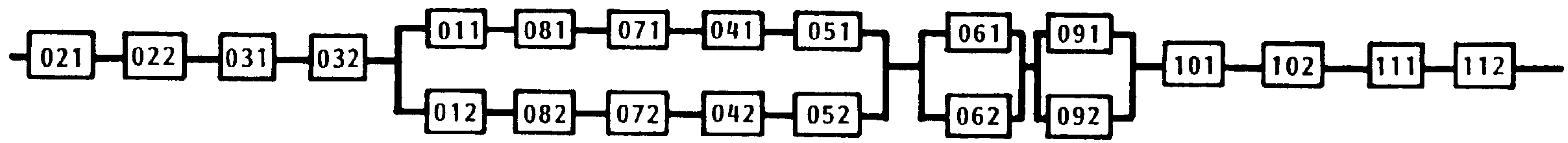
A scheme for using the observer algorithm for gust component filtering of the AOA signal is depicted in Figure 11. The wind forcing function vector is w, and it directly excites the airplane through Matrix B, but not the observer dynamics. Hence, the observer states do not reflect the wind forcing functions, in particularly with regard to the AOA signal. Hence, the observed AOA can be fed back into the DFCS less the atmospheric noise component present on the AOA vane output.

This effect is illustrated in the simulation time histories presented in Figure 12. AOA is essentially proportional to vertical velocity, or plunge, a signal that directly reflects vertical gusts. The corresponding observer signal is seen to be relatively free of the gust component, and hence it can be used advantageously for long term control functions like autothrottles. In certain cases however, like ride qualities control, the AOA gust component actually constitutes a meaningful signal, so filtering there would be inappropriate. In general, observers can be very useful because their filter characteristics are not strictly frequency dependent, a criteria that may not necessarily distinguish sensor signal content from the noise.



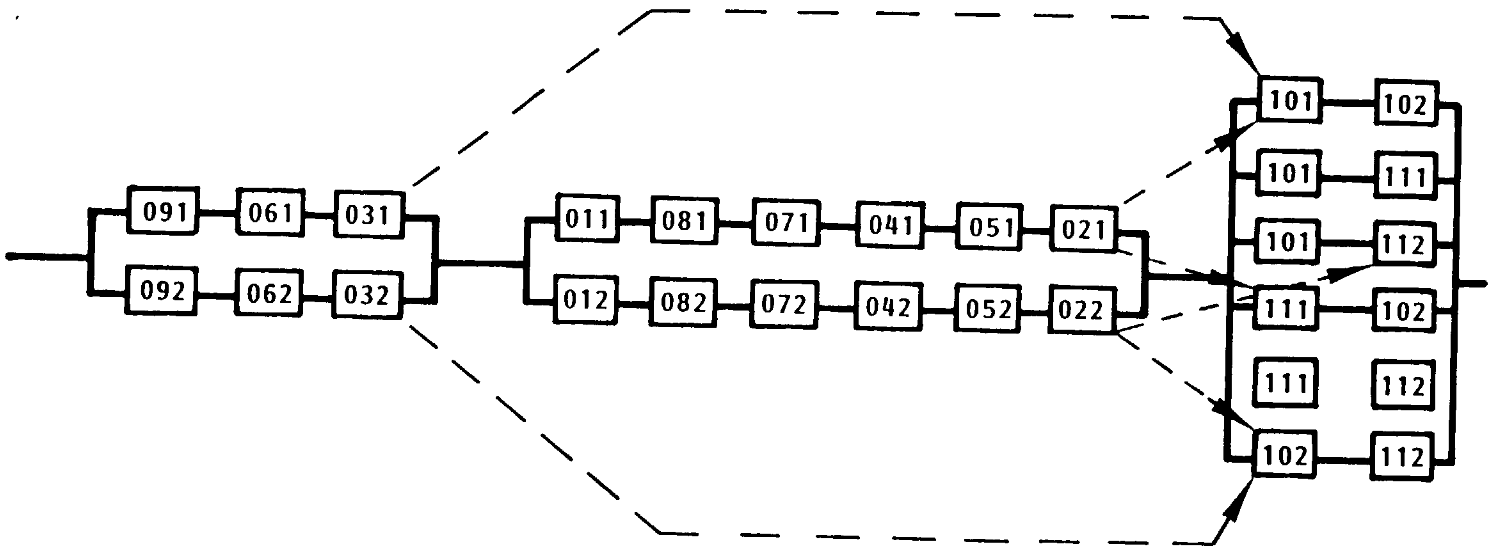
011 & 012	A500 POWER SUPPLIES	$\lambda = 21.0$	
021 & 022	A400 POWER SUPPLIES	$\lambda = 21.0$	
031 & 032	ACP201 POWER SUPPLIES	$\lambda = 29.0$	
041 & 042	HYDRAULIC SYSTEMS	$\lambda = 40.0$	
051 & 052	SERVO ACTUATORS	$\lambda = 15.0$	
061 & 062	STABILIZER POSITION LVDT'S	$\lambda = 0.0002$	
071 & 072	SERVO ENGAGE FUNCTIONS	$\lambda = 7.36$	
081 & 082	PROCESSORS	$\lambda = 8.8$	
091 & 092	CONTROL WHEEL SENSORS	$\lambda = 31.67$	ANALOG TO DIGITAL FUNCTIONS INCLUDED IN SENSOR FAILURE RATES. ALL FAILURE RATES GIVEN IN FAILURES PER MILLION
101 & 102	PITCH RATE GYROS	$\lambda = 309.67$	
111 & 112	NORMAL ACCELEROMETER	$\lambda = 80.67$	

Figure 9. Reliability Assessment Configurations (Sheet 1 of 3 - No Self Test/No Analytical Redundancy)



011 & 012	A500 POWER SUPPLIES	$\lambda = 21.0$	
021 & 022	A400 POWER SUPPLIES	$\lambda = 21.0$	
031 & 032	ACP201 POWER SUPPLIES	$\lambda = 29.0$	
041 & 042	HYDRAULIC SYSTEMS	$\lambda = 40.0$	
051 & 052	SERVO ACTUATORS	$\lambda = 15.0$	
061 & 062	STABILIZER POSITION LVDT'S	$\lambda = 0.0002$	
071 & 072	SERVO ENGAGE FUNCTIONS	$\lambda = 7.36$	
081 & 082	PROCESSORS	$\lambda = 8.8$	
091 & 092	CONTROL WHEEL SENSORS	$\lambda = 31.67$	ANALOG TO DIGITAL FUNCTIONS INCLUDED IN SENSOR FAILURE RATES. ALL FAILURE RATES GIVEN IN FAILURES PER MILLION
101 & 102	PITCH RATE GYROS	$\lambda = 309.67$	
111 & 112	NORMAL ACCELEROMETER	$\lambda = 80.67$	

Figure 9. Reliability Assessment Configurations (Sheet 2 of 3 - Self Test/No Analytical Redundancy)



011 & 012	A500 POWER SUPPLIES	$\lambda = 21.0$	
021 & 022	A400 POWER SUPPLIES	$\lambda = 21.0$	
031 & 032	ACP201 POWER SUPPLIES	$\lambda = 29.0$	
041 & 042	HYDRAULIC SYSTEMS	$\lambda = 40.0$	
051 & 052	SERVO ACTUATORS	$\lambda = 15.0$	
061 & 062	STABILIZER POSITION LVDT'S	$\lambda = 0.0002$	
071 & 072	SERVO ENGAGE FUNCTIONS	$\lambda = 7.36$	
081 & 082	PROCESSORS	$\lambda = 8.8$	
091 & 092	CONTROL WHEEL SENSORS	$\lambda = 31.67$	ANALOG TO DIGITAL FUNCTIONS INCLUDED IN SENSOR FAILURE RATES. ALL FAILURE RATES GIVEN IN FAILURES PER MILLION
101 & 102	PITCH RATE GYROS	$\lambda = 309.67$	
111 & 112	NORMAL ACCELEROMETER	$\lambda = 80.67$	

Figure 9. Reliability Assessment Configurations (Sheet 3 of 3 - Self Test/ Analytical Redundancy)

MISSION TIME IN HOURS	NO SELF TEST OR ANALYTICAL REDUNDANCY	SELF TEST, BUT NO ANALYTICAL REDUNDANCY	SELF TEST AND ANALYTICAL REDUNDANCY
1.00	$.93988 \times 10^{-3}$	$.88032 \times 10^{-3}$	$.29802 \times 10^{-7}$
2.00	$.18788 \times 10^{-2}$	$.17599 \times 10^{-2}$	$.14901 \times 10^{-6}$
3.00	$.28169 \times 10^{-2}$	$.26386 \times 10^{-2}$	$.40233 \times 10^{-6}$
4.00	$.37541 \times 10^{-2}$	$.35167 \times 10^{-2}$	$.71256 \times 10^{-6}$
5.00	$.46905 \times 10^{-2}$	$.43940 \times 10^{-2}$	$.11027 \times 10^{-5}$
6.00	$.56240 \times 10^{-2}$	$.52705 \times 10^{-2}$	$.15795 \times 10^{-5}$
7.00	$.65606 \times 10^{-2}$	$.61463 \times 10^{-2}$	$.21309 \times 10^{-5}$
8.00	$.74944 \times 10^{-2}$	$.70213 \times 10^{-2}$	$.27716 \times 10^{-5}$
9.00	$.84272 \times 10^{-2}$	$.78956 \times 10^{-2}$	$.35316 \times 10^{-5}$
10.00	$.93593 \times 10^{-2}$	$.87691 \times 10^{-2}$	$.43660 \times 10^{-5}$

TABLE 2. RELIABILITY ASSESSMENT RESULTS

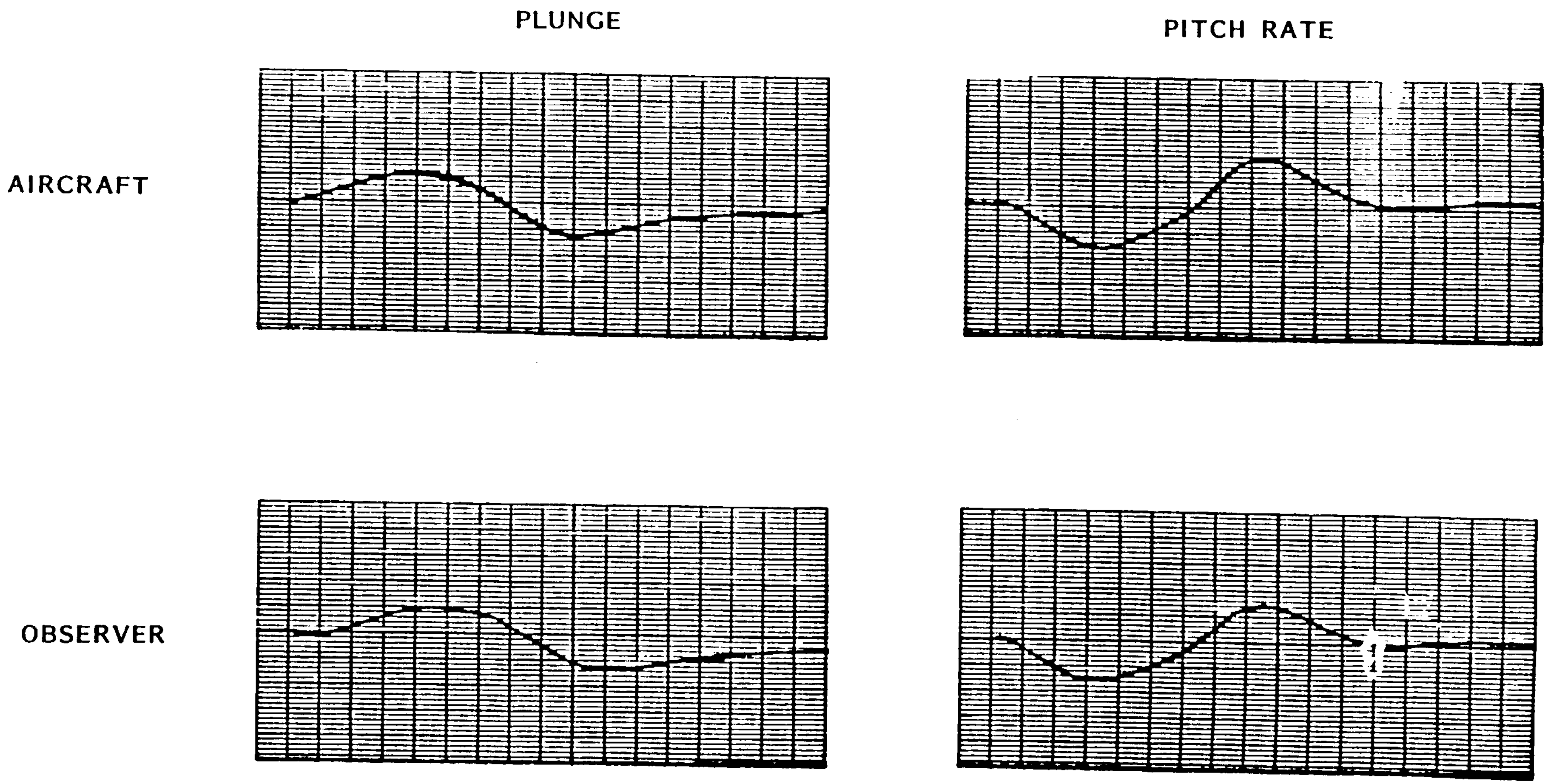


Figure 10. Observer Time History Response (Sheet 1 of 5 - Normal DFCS Sensors)

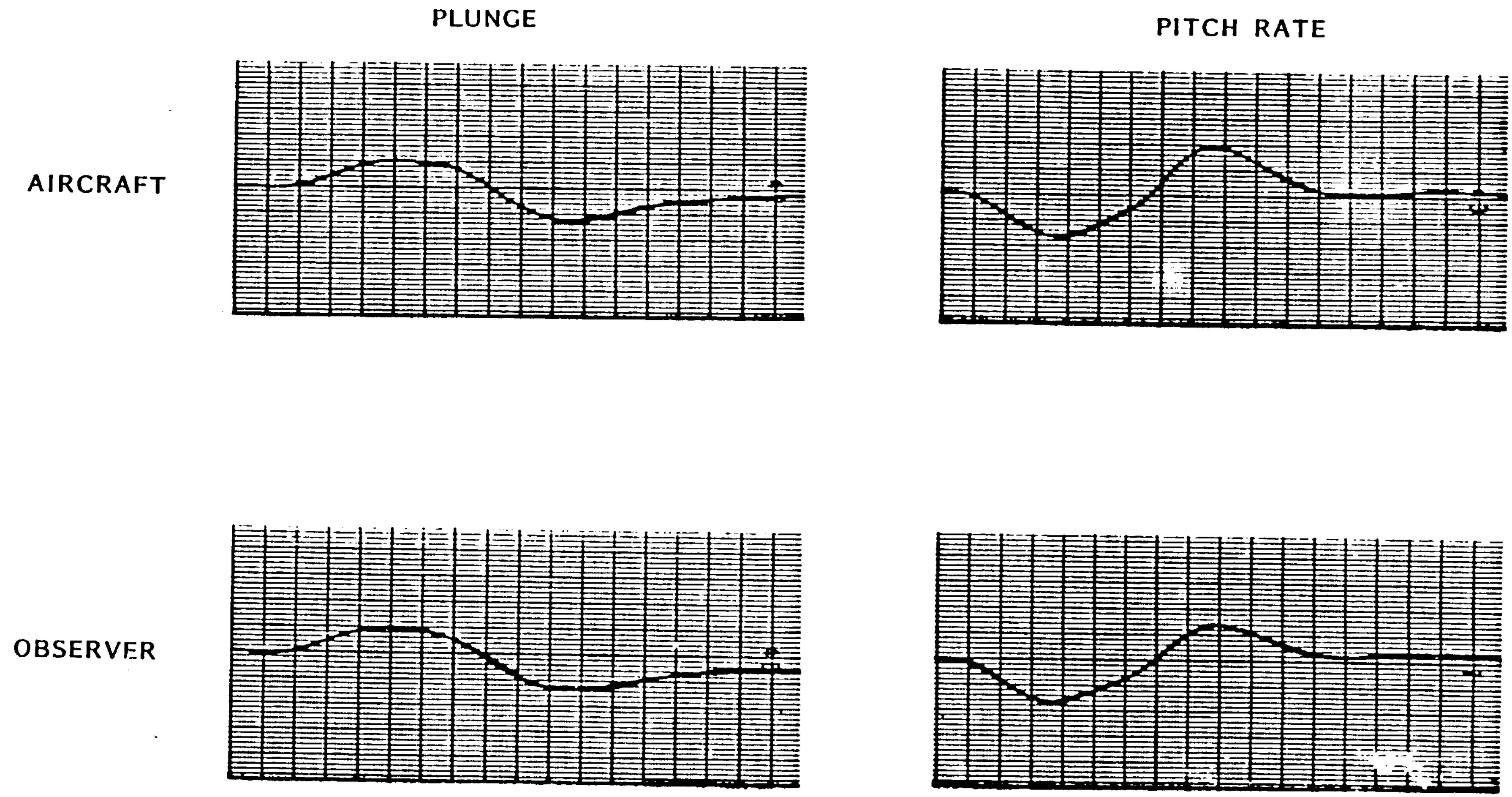


Figure 10. Observer Time History Response (Sheet 2 of 5 - Normal DFCS Sensors)

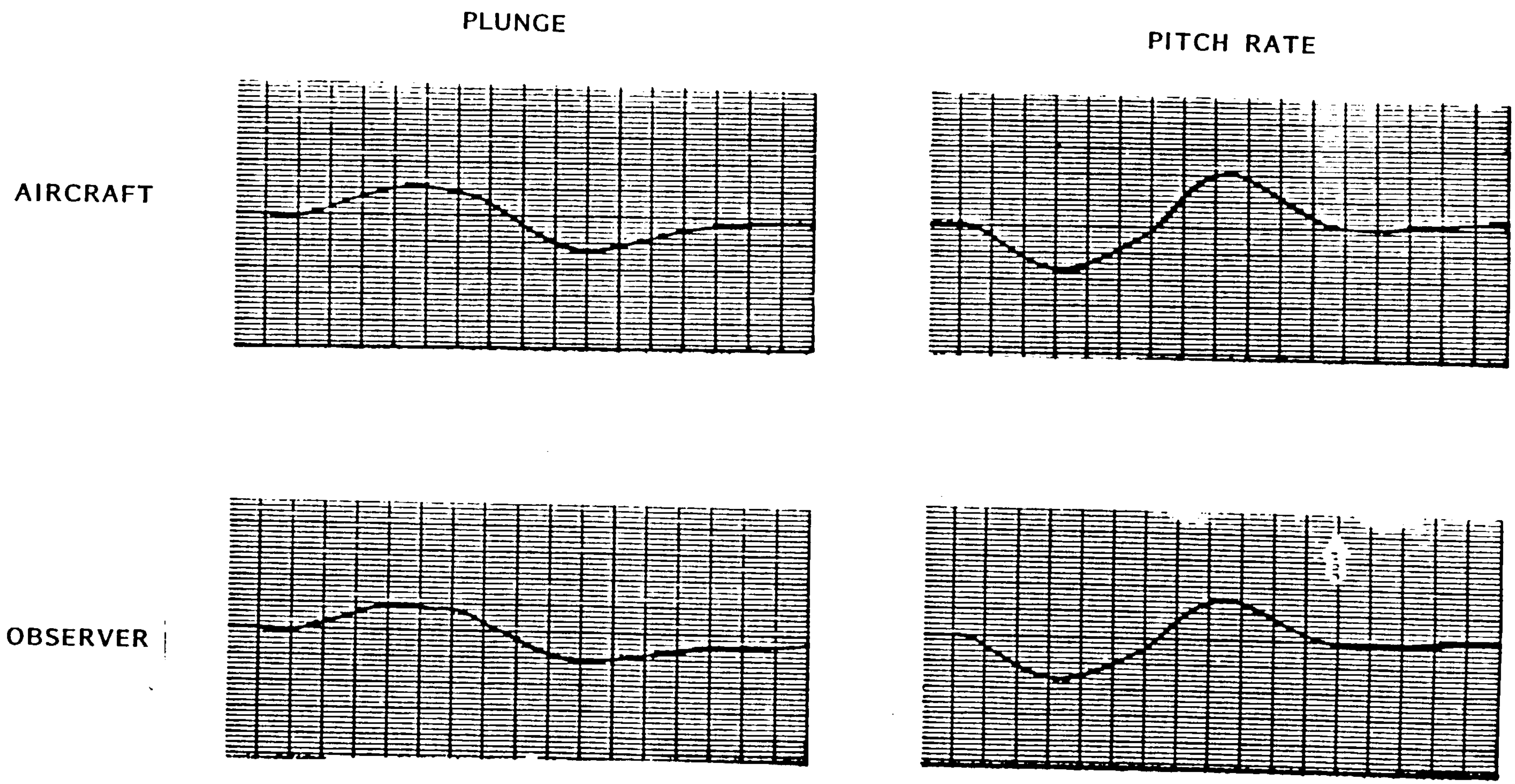


Figure 10. Observer Time History Response (Sheet 3 of 5 - Normal DFCS Sensors)

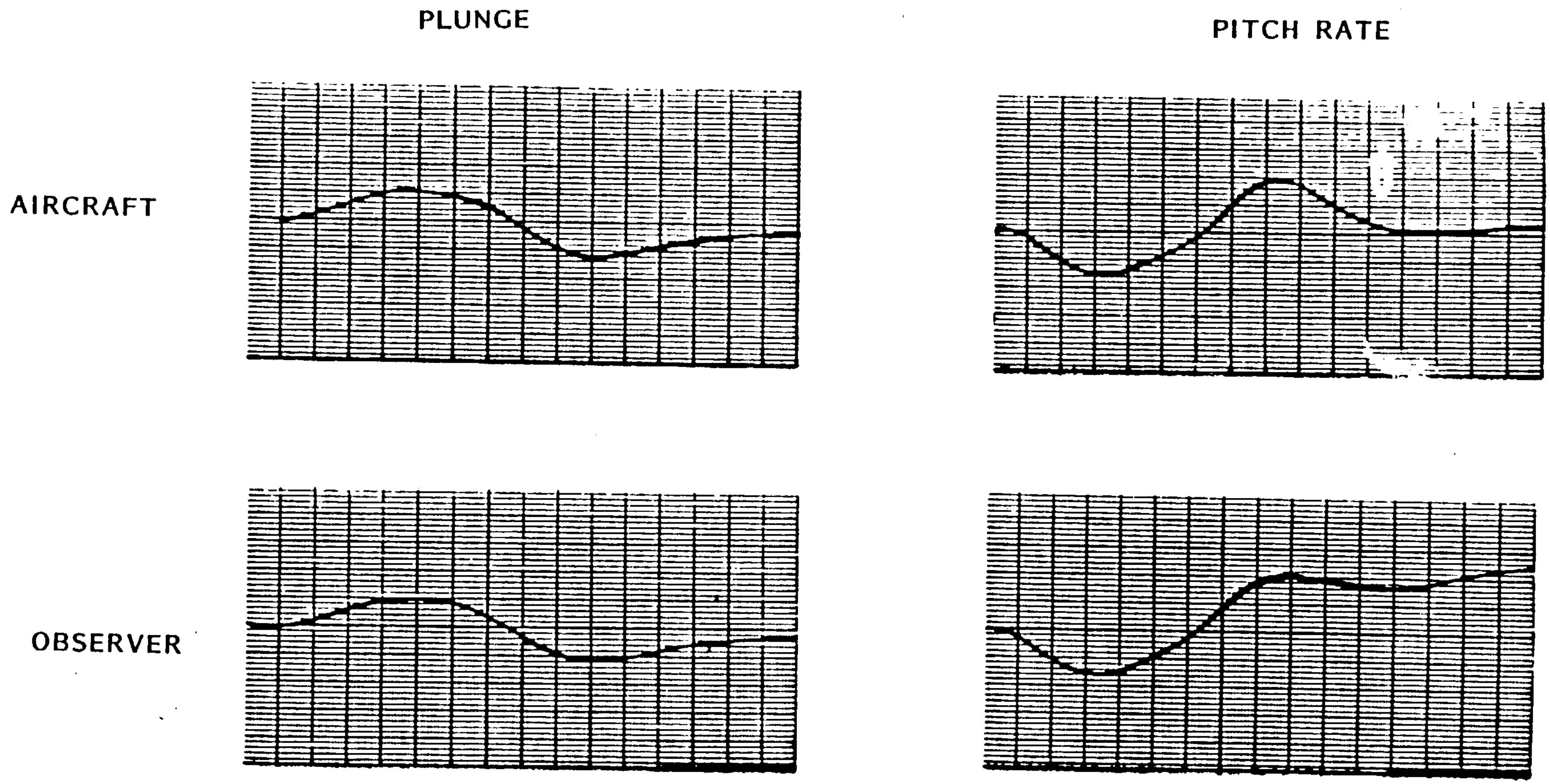


Figure 10. Observer Time History Response (Sheet 4 of 5 - Normal DFCS Sensors)

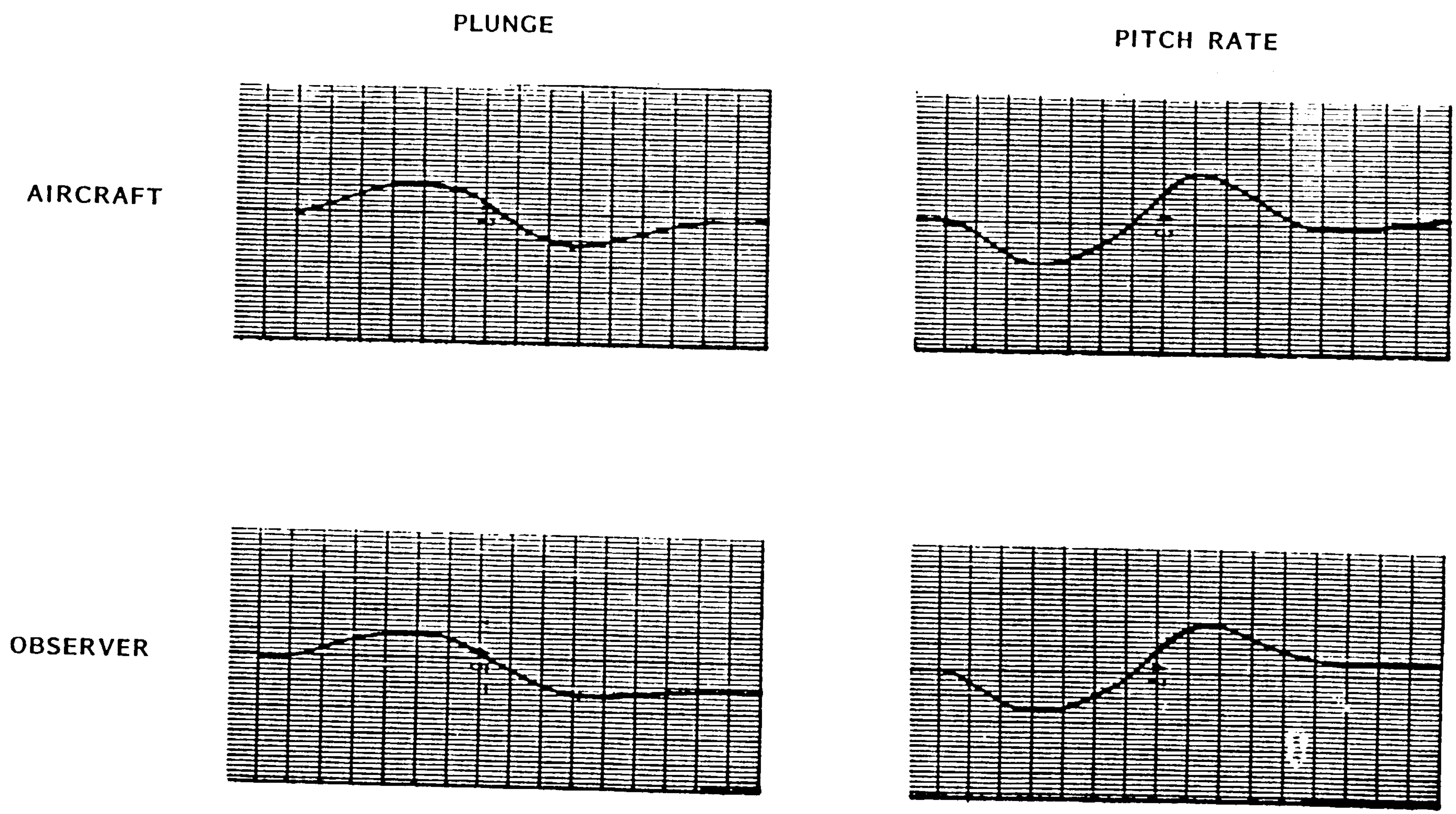


Figure 10. Observer Time History Response (Sheet 5 of 5 - Normal DFCS Sensors)

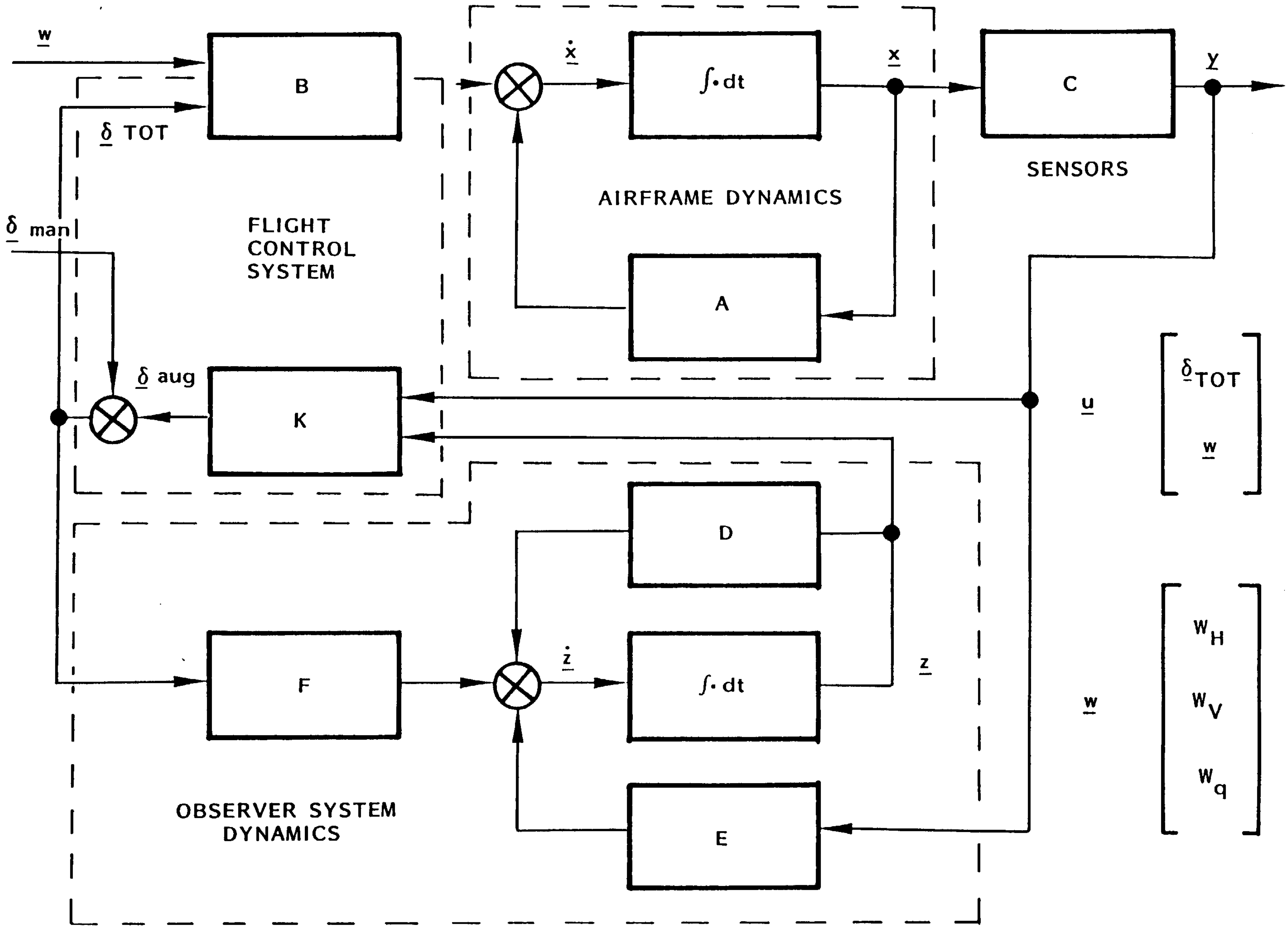


Figure 11. Observer Filtering Block Diagram

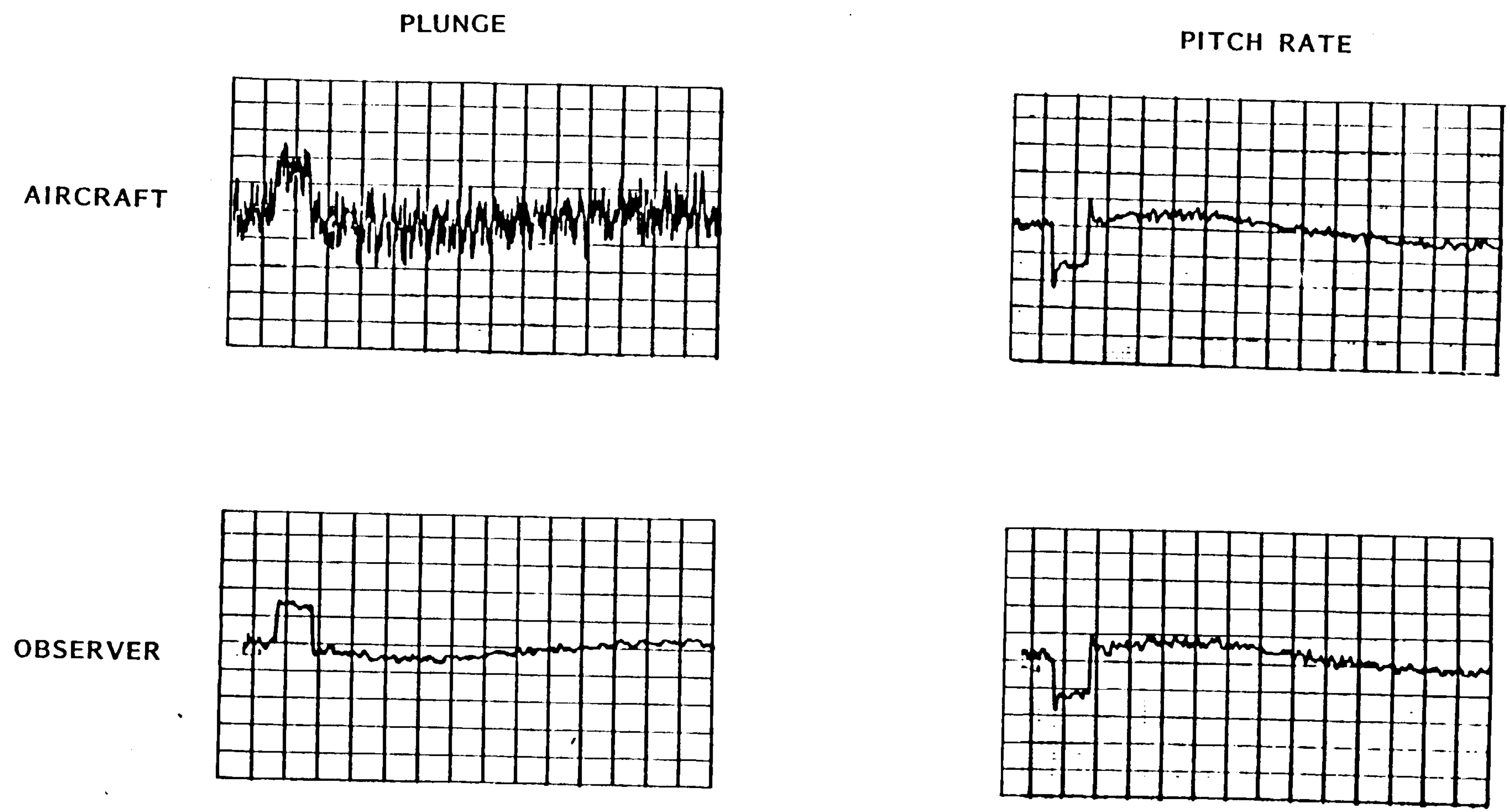


Figure 12. Observer Filtering Time History

5.0 SUMMARY

Basically, the technical feasibility and benefits of analytical sensor redundancy have been illustrated. The major benefit of extended likelihood of maintaining safe flying qualities has been demonstrated for a negative RSS transport airplane. Marginal reliability improvements have been calibrated, and the impact on validation/revalidation indicated.

Ultimately, the decision to employ analytical sensor redundancy is found to be primarily an economic one, for conventional sensor redundancy and reliability levels are quite adequate at present. Still, the tradeoff between reduced sensors and increased software overhead may in certain cases favor analytical redundancy, particularly where its use is also warranted by associated nonlinear filtering applications. There, DFCS performance benefits may prove to be vital.

References

1. Mulcare, D.B., L.E. Downing, and M.K. Smith: "Quadruplex Digital Flight Control System Assessment," DOT/FAA/CT-86/30, November 1987.

2. Luenberger, D.G.: "Observing the State of a Linear System," IEEE Transactions on Military Electronics, April 1964.

3. Luenberger, D.G.: "Observers for Multivariable Systems," IEEE Transactions on Automatic Control, April 1966.

4. Masreliez, J.C., and B.E. Bjurman: "Fault-Tolerant System Reliability Modeling/Analysis," Journal of Aircraft, August 1977.

5. Mulcare, D.B., and L.A. Barton: "N-Version Software Demonstration for Digital Flight Controls," DOT/FAA/CT-86/33, April 1987.

6. DeRusso, Roy, and Close: "State Variables for Engineers," Prentice-Hall, 1965.

7. Mulcare, D.B., and W.E. Larsen: "Real-Time Execution Monitors for Validating Fault Tolerance," National Aerospace Electronics conference, May 1988.