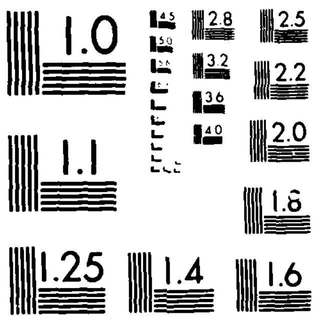
AD-A165 082 AN APPLICATION OF ARTIFICIAL INTELLIGENCE THEORY TO RECONFIGURABLE FLIGHT..(U) PRINCETON UNIV NJ DEPT OF MECHANICAL AND AEROSPACE ENGINEERIN.. D A HANDELMAN 1/1

UNCLASSIFIED 30 JUN 84 MAE-1665 ARO-20155.2-MA F/G 6/4 NL

END

FILMED

DTIC



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS 1963-A

20155.2-MA

1

1665-MAE

AN APPLICATION OF ARTIFICIAL INTELLIGENCE THEORY TO RECONFIGURABLE FLIGHT CONTROL

David A. Handelman

June 30, 1984

Interim Technical Report

U.S. ARMY RESEARCH OFFICE

Contract No. DAAG29-84-K-0048

PRINCETON UNIVERSITY

Department of Mechanical and Aerospace Engineering

Princeton, NJ 08544

Approved for Public Release:

Distribution Unlimited

ABSTRACT

A knowledge-based flight control system capable of detecting, identifying, and reconfiguring for a wide range of aircraft failures has been designed. Combining modern control theory, statistical hypothesis testing, and artificial intelligence techniques, this research addresses the question of whether or not an "intelligent" computer could assist a pilot during a failure. Analytical redundancy techniques, including a Generalized Likelihood test, are used for failure detection. Failure diagnosis is performed by an expert system. Utilizing knowledge of cause-and-effect relationships between all aircraft components and the statistical results of a Multiple-Model algorithm, the expert system decides which aircraft component has failed and how to reconfigure for the failure. Preliminary tests on an 8-bit microprocessor system were conducted and are summarized, and plans to expand to a 16-bit multi-microprocessor system are outlined.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

As machines become more complicated, the possibility of system failures increases. Component failures can have disasterous effects on the operation of any system, but in the case of aircraft, the consequences of such an event can include failure to complete the mission, massive property damage, and loss of life. The goal of this research effort is to identify control system architectures that will enhance the ability of present and future aircraft to accomodate failures. What separates this research from previous efforts in aircraft failure detection, identification, and reconfiguration (FDIR) is the desire to accomodate an expanded range of system failures, including not only control system components but elements of the aircraft itself. Techniques of artificial intelligence theory are employed along with statistical hypothesis testing and modern control theory in accomplishing this task.

2. TRENDS IN AIRCRAFT FAILURE ACCOMODATION

Failure accomodation has long been a primary concern of aircraft designers. Consequently, trends in aircraft FDIR techniques closely follow trends in aircraft design. The Wright brothers designed their aircraft with the belief that the pilot should have maximum authority over its motion. Unfortunately, this high degree of maneuverability resulted in marginal static stability requiring exececssive pilot effort. In later years, increased stability and autopilots helped reduce pilot workload. In order to reduce the likelihood of catastrophic failures, aircraft were designed with large safety margins. It was hoped that the aircraft would degrade gracefully and "limp home" in the event of a failure. However, failure accomodation became more complicated when the desire to carry large payloads emerged. The introduction of hydraulics into the control loop marked the first step in separating the pilot from the control surfaces, and this required redundancies to avoid crippling failures of the primary flight control system.

Recent design developments affect both the operational cost and maneuverability of modern aircraft. Active controls permit smaller tail surface area, resulting in lower fuel consumption. Fly-by-wire control systems help integrate stability augmentation, command augmentation, autopilot, and trim subsystems. The combination of the two allow increased maneuverability and the possibility of reconfiguration. However, the cost and weight penalities incurred by the required redundancies (i.e. duplex, triplex, or even quadruplex sensors or actuators) often prohibit the use of such systems.

The cost and weight savings that can be afforded by a reduction of these redundancies point to the need for reliable failure detection and identification (FDI) methods. Figure 1 shows some of the trade-offs involved in sensor FDI. The easiest way to detect and identify a sensor failure is to compare three sensors which measure the same quantity. Such a triplex system can be very expensive, however. In the less expensive duplex system, a failure is easy to detect but hard to identify. Additionally, functional redundancy between unique sensors can be exploited to further reduce costs. For example, a rate gyro and an accelerometer can each provide pitch rate information; therefore, the signals can be compared to detect a failure in one of the two components.

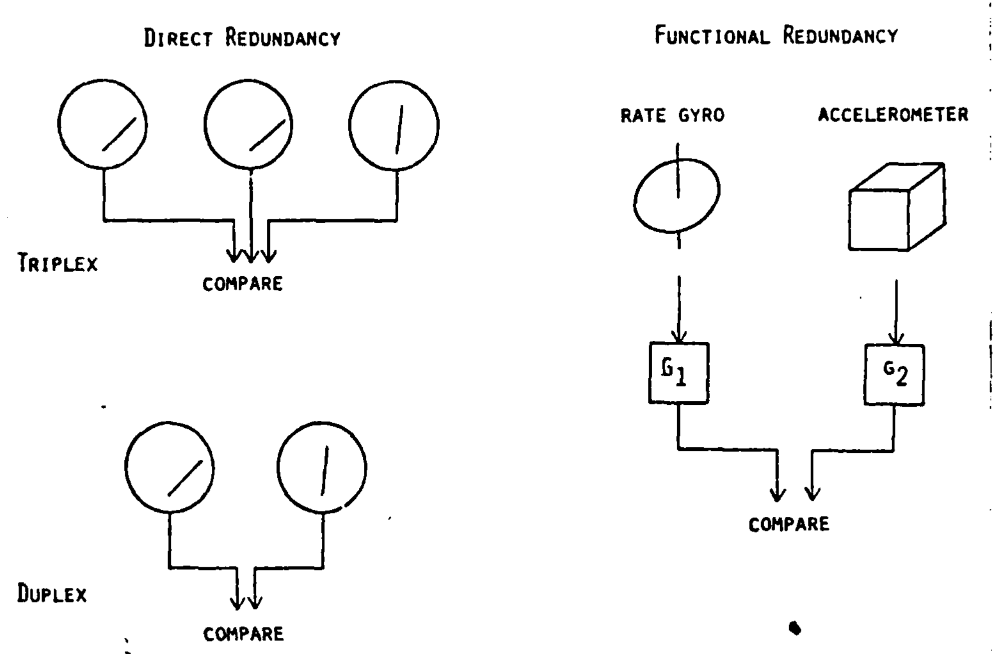


Figure 1: Failure Detection and Identification Problems

Although seemingly straightforward, these FDI techniques can run into problems. Consider a triplex system where two of the sensors are powered from one electrical source and the third sensor from a different source. If the triplex FDI scheme identified a failure by singling out the one sensor which differed from the other two, a power failure to the first two sensors would be misconstrued as a failure of the third. This brings up the need for the incorporation of intelligence in the failure diagnosis process, an intelligence that will recognize when such "higher-order" relations among different elements of the aircraft exist.

How have these basic techniques been used in existing aircraft? Table 1 gives a sampling of aircraft (two military, one research, and one civil) which depend on these techniques to handle sensor and actuator failures, as described in {1,2}.

TABLE 1

Trends in Aircraft Failure Accomodation

AIRCRAFT	FDIR METHOD: SENSORS	FDIR METHOD: ACTUATORS
• SAAB VIGGEN JA37 1st MILITARY AIRCRAFT IN SERIES PRODUCTION AND FIELD SERVICE WITH DIGITAL AUTOMATIC FCS	DUPLEX/ COMPARISON	SIMPLEX/ ANALYTICAL RED. (SERVO MODEL) WITH REVERSION TO TRIM POSITION
• GENERAL DYNAMICS F-16 ANALOG FLY-BY-WIRE	QUADRUPLEX/ COMPARISON	QUADRUPLEX
• NASA F-8 DFBW FULL AUTHORITY DFBW CONTROL WITHOUT MECHANICAL REVERSION (ANALOG BACKUP)	TRIPLEX/ COMPARISON DUPLEX/ ANALYTICAL RED. (STATISTICAL TESTING)	TRIPLEX
• SPACE SHUTTLE ORBITER FULL AUTHORITY DFBW CONTROL WITHOUT MECHANICAL REVERSION	HARDWARE AND ANALYTICAL RED.	HARDWARE AND ANALYTICAL RED.

3. MOTIVATION FOR RESEARCH

The sampling of fatal air carrier accidents shown in Table 2 provides an indication of failure types that must be considered. As illustrated by the three groups, many accidents appear to be the result of a misuse of information, knowledge, or capability. For instance, a pilot depends on instruments for accurate aircraft status information. Inaccurate or partial information deprives the pilot of the resources necessary to safely operate the aircraft, leading to misuse of information. Similarly, negligence or inexperience on the part of the pilot represents a misuse of knowledge. Finally, modern jet aircraft have highly redundant control effectors. As a result it, may be possible to counterbalance the effect of a failed primary control effector, such as an aileron, with a secondary control effector, such as a trailing edge flap. If an aircraft is controllable following a failure, but through a lack of information, knowledge, or ability the pilot fails to control it, this represents a misuse of capability. Nowhere is this fact as pronounced as in the circumstances surrounding the non-fatal 1977 Delta Flight 1080, in which the left elevator jammed up 19 degrees {3}, or the 1979 American Airlines DC-10 crash, in which a separated engine pylon disabled slat disagreement and stall warning systems {4}.

It is felt that these aircraft were still controllable following their failures, but the pilots could neither recognize nor react to the failures fast enough. The present research addresses the question of whether or not an "intelligent" computer could assist a pilot in such a situation.

TABLE 2

Fatal Accidents of U.S. Scheduled Air Carriers, 1961-1979

<u>Information</u>

Reverse Thrust Warning Light Malfunction
Landing Gear Warning Light Malfunction
Loss of Electrical System to Attitude Instruments

<u>Knowledge</u>

Turbulence, Airframe Failure in Flight
Hydraulic Pressure Loss Uncorrected by Pilot

<u>Capability</u>

Hydraulic System Degradation
Rudder Support Material Failure
Rudder Control System Malfunction
Flight Control System Failure
Engine Pylon Failure

4. RESEARCH OBJECTIVES

The objective of this research it to use artificial intelligence techniques, along with statistical hypothesis testing and modern control theory, to help the pilot utilize information, knowledge, and capability in the event of one or more failures. An "intelligent" flight control system that uses knowledge of cause-and-effect relationships between all aircraft components will be developed. It will screen the information available to the pilot to aid in its interpretation, to supplement the pilot's knowledge, and most importantly, to utilize the remaining flight capability of the aircraft following a failure through reconfiguration.

The types and modes of failures that the system will be expected to handle include those in Table 3. Failure type corresponds to the type of aircraft element that has failed. Types of elements include sensors, controls, actuators, effectors, and supports. Additionally, structural failures are classified according to where they occur on the airframe. Failure modes pertain to how the failure affects the given element. A sensor, for example, can suffer from bias shift, sticking at a certain position, increased noise, and intermittent operation. By identifying the specific mode, a decision can be made as to whether or not reliable information can still be retrieved from that sensor following the failure. Similarly, an intelligent flight control system can decide whether or not an actuator or effector can be relied upon to command a certain control response following a failure.

TABLE 3

Failure Types and Modes

Failure type	Examples
sensor	pitot-static tube
control	stick, rudder pedals, throttle
actuator	aileron actuator, elevator actuator
effector	aileron, elevator
support	batteries, electrical wires, hydraulic lines
structural	wing damage

Failure mode	Examples
bias	sensor bias jump
stuck	null, hard-over
noise	sensor noise increase, turbulence or microburst encounter
intermittent	broken wire

5. INITIAL ASSUMPTIONS

In order to adapt to significant failure-induced changes in the configuration of the aircraft, the control system must have a variable structure. A fly-by-wire flight control system can be reconfigured by supplying new mathematical models and gains to the computer; thus, a control system (with no computer-related failures) of this form is assumed. Note that the pilot flies the aircraft via the flight computer and has no direct link to the control surfaces. It is essential, therefore, that the flight computer have the mathematical model and gains corresponding to the actual aircraft configuration. Assuming that a failure will significantly change the configuration, it will be the job of the Knowledge-based Reconfigurable Flight Control System (KRFCS) to replace the pre-failure model with the correct model. The assumption of the presence of a fly-by-wire system is not unreasonable for the reasons stated above, and in fact it is hoped that the results of this effort will help speed the acceptance of such systems into more types of aircraft.

Previous work in aircraft FDIR {5,6,7,8,9} has centered around sensor and actuator failures; this effort will encompass more complicated failures, such as those due to structural damage. The FDIR scheme will have to extend the idea of functional redundancy to that of analytical redundancy containing relationships between all aircraft components. It also must contain information about the effects of failures on aircraft behavior. The starting point will be the aircraft nonlinear equations of motion shown in Appendix A and described in {10}.

Although these equations can be solved by computer, no closed-form solutions exist; therefore, linearization in state space form is performed to make the problem analytically tractable. This results in a perturbation equation describing aircraft motion about a nominal trajectory. With the assumptions of a fixed sampling interval and a piecewise constant input, a discrete-time sampled data system is produced. In order to include the effects of failures in the model, deterministic biases and zero-mean, Gaussian, white noise sequences are included. The resultant stochastic dynamic system, which is an approximation to the actual nonlinear aircraft dynamics, is shown in block diagram form in Appendix A.

Next, a controller and estimator are designed for the aircraft using the linear discrete-time model. Note that the model corresponds to the nominal, non-failed aircraft. With no failures, the closed-loop system shown in Fig. 2 will provide the pilot with the resources necessary to safely operate the aircraft. If a failure occurs and the aircraft configuration changes significantly, the controller will have the wrong gains for the present configuration, and the estimator will be trying to predict the behavior of an aircraft configuration which no longer exists. At this point, the KRFCS must provide the controller and estimator with the correct numbers to keep the aircraft flying.

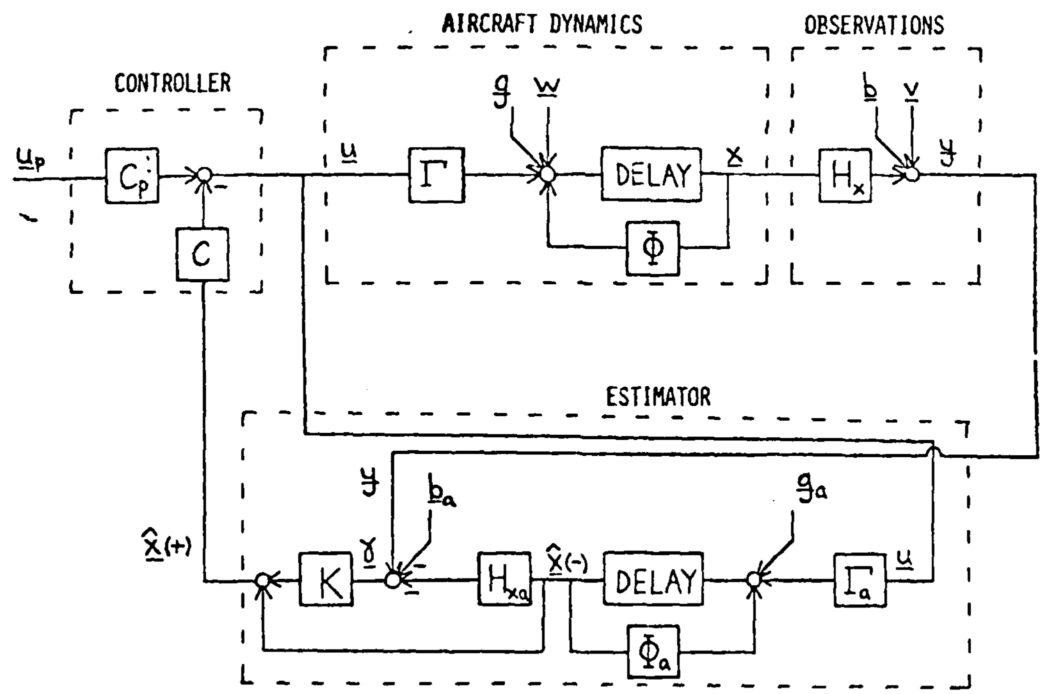


Figure 2: Closed-Loop Dynamic System

6. ISSUES OF FDIR

The next step in the design procedure is to identify an intelligent algorithm which accomplishes failure detection, identification, and reconfiguration with the given control structure. One method of dealing with the problem is to automate the procedure a human observer would follow if given enough time. The KRFCS supervises aircraft behavior until some abnormality occurs, at which time a failure flag is raised. The system then allocates its resources to best serve the problem-solving process. This will be important if implementation requires a multi-microprocessor environment. Next, the system tries to diagnose exactly what has failed. Concurrently, immediate and temporary measures are taken to help reduce the effect of the failure during diagnosis. An example of such compensation would be the deflection of a flap to offset a sudden, unexplained roll. When the failure is identified, the best control configuration given the present circumstances is chosen and reconfiguration begins. Finally, the new control scheme is implemented.

In the present system, reconfiguration will be the easiest task the KRFCS must perform. When a failure has been detected, the output of the ensuing diagnosis will be the name of the failure which is most likely to have occurred (given the background knowledge and failure-time information available to the system). Reconfiguration will involve looking in memory for the pre-calculated model and gains corresponding to that failure. This implies that the system contains in memory a vast array of models and gains, calculated off-line, corresponding to every conceivable failure, or at least to all the failures the system will be expected to handle. Questions of how many failures to include and what resolution is necessary to ensure reasonable handling qualities (aileron failed in increments of 4 degrees, or 2 degrees, or 0.5 degrees, ...) will be addressed in the future. This FDIR scheme is depicted in Fig. 3.

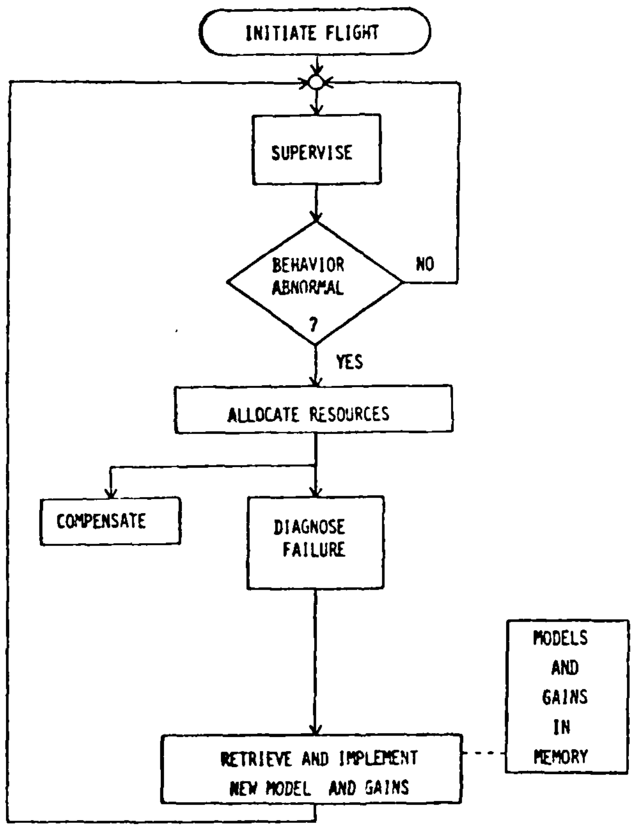


Figure 3: FDIR - Flow Chart

A more aesthetically pleasing solution to the problem of aircraft fault tolerance would be a restructurable control system which could perform on-line parameter estimation and on-line gain calculation; however, parameter estimation schemes with the required accuracy and speed remain to be defined. Additionally, a restructurable system stresses processor speed while the reconfigurable one presented stresses memory. With implementation a prime goal, it appears that the "brute force" method of pre-calculated failure models and gains in memory is more feasible at the present time. More comparisons between reconfigurable and restructurable control can be found in {11}.

The problems of failure detection and diagnosis remain. When the attempt is made to detect and diagnose all types of failures (not simply sensor failures, as described earlier) it is necessary to use all the analytical redundancy available, such as that contained in the mathematical model of the aircraft. The Generalized Likelihood Ratio (GLR) Method and the Multiple-Model (MM) Method {7}, both summarized in Fig. 4, are two algorithms that use this redundancy to choose from a finite set of alternatives the model which best predicts the actual aircraft behavior. In FDI, the set of alternatives would be the set of failures one hopes to detect and identify.

The GLR method uses the innovations process from a single nominal Kalman filter to calculate the likelihood that a given system bias jump has occurred. For this reason, the computational load is low, and the method quickly detects failures. Because only additive effects can be modeled, failures that produce parametric model changes can not be easily diagnosed. Consequently, the GLR method is useful principally for detection in our application.

FDI WITH THE GENERALIZED LIKELIHOOD RATIO (GLR) METHOD

SYSTEM WITH BIAS JUMP DUE TO FAILURE

$$\underline{x}(k+1) = \Phi\underline{x}(k) + \Gamma\underline{u}(k) + \underline{w}(k) + \underline{f}_i(k,\theta)\nu$$

$$\underline{y}(k) = H_x\,\underline{x}(k) + \underline{v}(k) + \underline{g}_i(k,\theta)\nu$$

WITH UNKNOWN BIAS INNOVATIONS WILL NO LONGER BE WHITE

$$\underline{\gamma}(k) = \underline{\gamma}_N(k) + \rho_i(k,\theta)$$

OFF-LINE CALCULATIONS

- RECURSIVE EQUATIONS FOR $\rho_i(k)$
- $a_i(k,\theta) = \sum_{j=\theta}^{k} \rho_i^T(j,\theta)\, V^{-1} \rho_i(j,\theta)$

ON-LINE CALCULATIONS

- $d_i(k,\theta) = \sum_{j=\theta}^{k} \rho_i^T(j,\theta)\, V^{-1} \gamma(j)$
- LIKELIHOOD $\ell_i(k) = \max_\theta \dfrac{d_i^2(k,\theta)}{a_i(k,\theta)} = \max_\theta \left[2\nu\, d_i(k,\theta) - \nu^2 a_i(k,\theta)\right]$

FDI WITH THE MULTIPLE MODEL (MM) ALGORITHM

EACH HYPOTHESIZED FAILURE HAS A KALMAN FILTER ASSOCIATED WITH IT

$$\hat{\underline{x}}_i(k+1|k) = \Phi_i\hat{\underline{x}}_i(k|k) + \Gamma_i\underline{u}(k) + \underline{g}_i(k)$$

$$\hat{\underline{x}}_i(k+1|k+1) = \hat{\underline{x}}_i(k+1|k) + K\,\underline{\gamma}_i(k+1) \qquad \underline{\gamma}_i(k+1) = \underline{y}(k+1) - H_{x_i}\,\hat{\underline{x}}_i(k+1|k) - \underline{b}_i(k)$$

IF HYPOTHESIS i IS TRUE $(1 \leq i \leq N)$ THEN $\underline{\gamma}_i$ IS A ZERO-MEAN GAUSSIAN SEQUENCE

COVARIANCE: $E\left[\gamma(k)\,\gamma^T(k)\right] = V_i(k)$

PROBABILITY DENSITY: $F_i\left[\underline{\gamma}_i(k)\right] = \dfrac{\exp\left[-\frac{1}{2}\gamma_i^T(k)\, V_i^{-1}\, \underline{\gamma}_i(k)\right]}{\left[(2\pi)^m \det V\right]^{1/2}}$

BAYES RULE GIVES RECURSIVE FORMULA FOR CONDITIONAL PROBABILITIES

$$p_i(k+1) = \frac{F_i\left[\underline{\gamma}_i(k+1)\right]\cdot p_i(k)}{\sum_{j=1}^{N} F_j\left[\underline{\gamma}_j(k+1)\right]\cdot p_j(k)}$$

Figure 4: Summary of GLR and MM Algorithms

The MM method, on the other hand, runs a Kalman filter for each of the failure hypotheses, so it can accomodate parametric as well as additive failures. It also is tolerant of non-Gaussian noise, which may occur in practice. Using Baye's rule, the algorithm points to the model which best predicts aircraft behavior, which is the one the controller and estimator should use following a failure. The method is slow at detecting failure-induced model switches; therefore, it is used principally for failure identification. One way to accomplish FDI would be to first detect a failure with the GLR test, then run the MM algorithm to choose the proper model from the set of all possible failure models.

The observations used by the estimator are derived from signals provided by the Flight Sensors, which are but a subset of all the sensors on the aircraft. Auxiliary Sensors, which measure quantities such as battery voltages and hydraulic line pressures, can convey important failure-time information and should be included in the FDIR scheme. In addition to the GLR test, the KRFCS should look at auxiliary sensor signal levels to see if warning thresholds have been exceeded, and at transition rates to see if the signal has jumped an unreasonable amount in a given amount of time. The flow chart for such a system is shown in Fig. 5.

The KRFCS will be expected to handle many types of failures. Each failure will change the aircraft configuration in a unique way and will, therefore, have a unique model associated with it. If the previously-mentioned FDIR scheme is employed, the MM algorithm will be required to choose among thousands of models. Although this may be a theoretically feasible solution, it will require an immense amount of computing

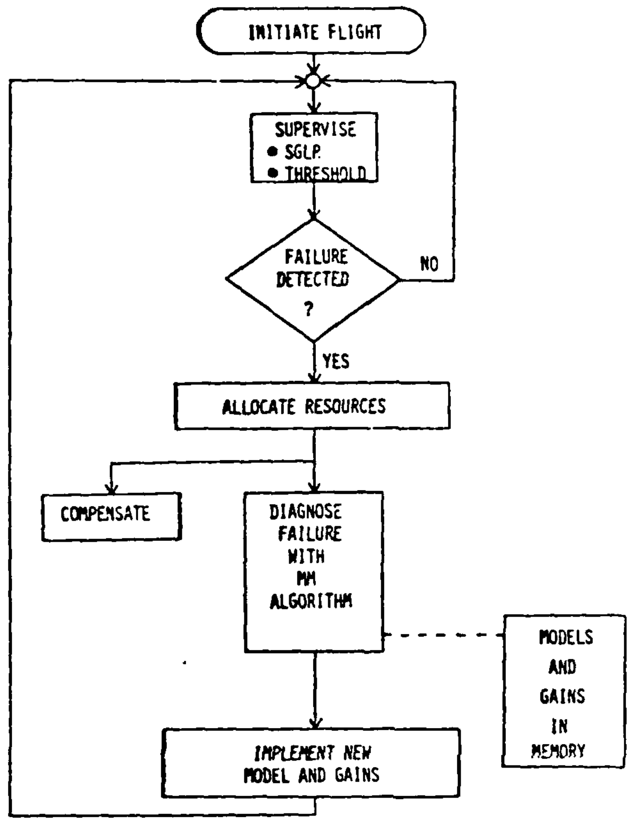
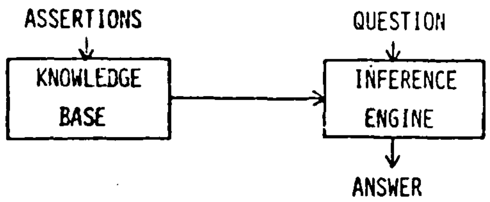


Figure 5: FDIR with Analytical Redundancy - Flow Chart

power. Our goals include eventual implementation of the control system, and computer resources must be kept to a minimum. If there was a way to let the MM algorithm test only those models corresponding to failures that are most likely under the circumstances, the required computer speed could be drastically reduced. What is needed is a "black box" that contains enough knowledge and "intelligence" to be able to perform failure diagnosis. It should take as input the failure-time status of all aircraft sensors and GLR test results and give as output a "failure candidate list" naming the most probable failures. It is assumed that given ample time, a human expert with years of experience in analyzing aircraft failures would be able to provide such a service; however, aircraft failures are time-critical events. Recognizing that the speed and memory capabilities of modern digital computers could provide a solution, it is necessary to draw on the techniques of artificial intelligence (AI) theory.

7. THE ROLE OF ARTIFICIAL INTELLIGENCE

Artificial Intelligence research attempts to make computers perform tasks that require the emulation of human intelligence. Although this research includes work in areas such as natural language understanding and computer vision, it is the knowledge-based system (KBS) that best suits our purpose. The structure of a KBS is shown in Fig. 6, as seen in {12}. An inference engine, that acts as a reasoning control structure, combines facts, assumptions, definitions, and heuristics about the world to produce an answer to a specific question. If the reasoning mechanism encapsules the knowledge of a human expert, the KBS is called an expert system. The knowledge-based reconfigurable flight control system will contain an expert system that will answer the question, "What failures are most likely given the following information?"



- KNOWLEDGE BASE CONTAINS
  - FACTS
  - ASSUMPTIONS
  - DEFINITIONS
  - HEURISTICS
- INFERENCE ENGINE PERFORMS REASONING

Figure 6: Structure of a Knowledge-Based System

8. A KNOWLEDGE-BASED RECONFIGURABLE FLIGHT CONTROL SYSTEM

The job of the expert system is to narrow down to a reasonable number the list of possible failures to be tested by the MM algorithm. A failure is detected when a sensor value goes beyond a pre-specified warning level, or if it jumps too quickly, or if a state or observation bias jump is picked up by the GLR. This information is then passed on to the expert system. With knowledge of the cause-and-effect relationships among all aircraft components and failure diagnosis rules, the expert system decides which failures are most likely to have occurred. Accordingly, the system's knowledge base can be broken into two parts: the Global Data Base (GDB) and the Rules.

The global data base contains status information on aircraft components, called "value-data", and dependencies between different aircraft components, called "link-data". Examples of value-data include real-time sensor signal levels, sensor warning levels, estimator outputs, and component operational status. Link-data tells which sensors sense which components or states, which controls control which actuators, which actuators actuate which effectors, which effectors effect which forces and moments, and which forces and moments combine to influence which states. It also contains information on component location in the aircraft (which locations contain which components) and how each component is supported: electrically, hydraulically, or otherwise. The framework of the prototype KRFCS global data base, which includes only lateral-directional effects, is shown in Fig. 7.

The rules combine the facts, definitions, and assumptions contained in the GDB with heuristic reasoning to diagnose a failure. They are in the form of "IF ... THEN ..." productions which draw certain conclusions if certain conditions are met. The following example illustrates the type of rules the KRFCS expert system contains.

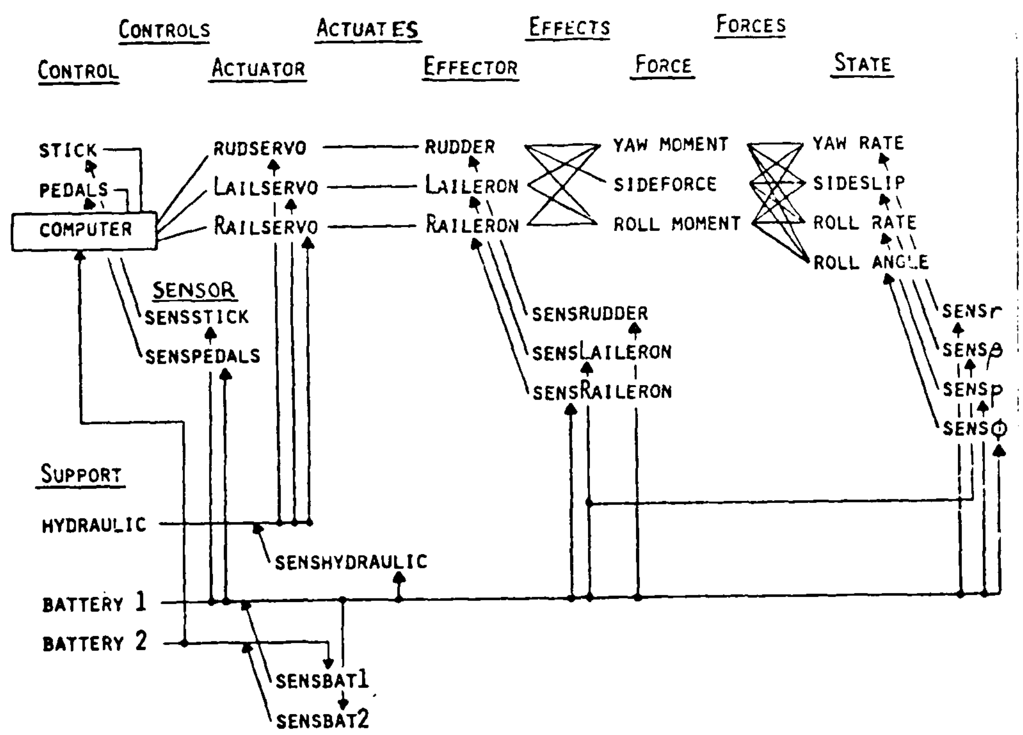


Figure 7: KRFCS Global Data Base

Rule 8:

IF a sensor (such as an aileron position sensor) has exceeded its expected value
AND that sensor senses an effector (such as an aileron)
AND no states (including roll rate) have exceeded their expected values

THEN a sensor failure is likely

Rule 10:

IF a sensor has exceeded its expected value
AND that sensor senses an effector
AND that effector strongly effects a state which has exceeded its expected value

THEN an effector failure is likely

These two rules show how the expert system can distinguish between a failed effector position sensor and a failed effector whose position is sensed. Both rules determine that a failure is likely under the given circumstances. This is called "diagnosis through validity". Other rules, labeled "diagnosis through contradiction", determine that a certain type or mode of failure is unlikely. The sixteen rules used in the earliest system experiments are contained in Appendix B.

Although the expert system may contain many rules, only a small number of them will be pertinent to a given failure at a given point in the diagnosis process. For example, if a failure is detected and no state bias jumps were observed by the GLR test, the expert system should not waste time testing rules that depend on the existence of a state bias jump in order to be true. The third part of the expert system, the "rule interpreters", provide the inference engine needed to select the appropriate rules to be tested.

Note that an advantage of this diagnostic technique is that the rules do not refer to individual aircraft components. When a failure is detected, the supervisor places the names of the offending sensors (as picked up by the threshold test) or bias jumps (as picked up by the GLR test) into special arrays. The rules and rule interpreters manipulate these arrays and special "scratch pad" memory stacks while carrying out the diagnosis. As the rules are executed, a running "scoreboard" keeps track of the most likely and unlikely failure types and modes, as well as the specific components involved. In this way the search for the failure can be kept at a high level of abstraction, and a small number of rules can provide a great deal of diagnostic power.

Rule 14, which deals with structural failures, is a good example of this (Appendix B).

The Multiple-Model algorithm processes a number of models and indicates which one best predicts aircraft behavior. However, the original list of candidate failures handed to the MM algorithm from the expert system may not contain the best model in all of memory. For this reason, the expert system must work with the MM algorithm. It must constantly update the failure candidate list until a model "close enough" to the actual failure is found. The knowledge-based reconfigurable flight control system, with expert system included, is shown in flow chart and block diagram forms in Fig. 8 and 9, respectively.

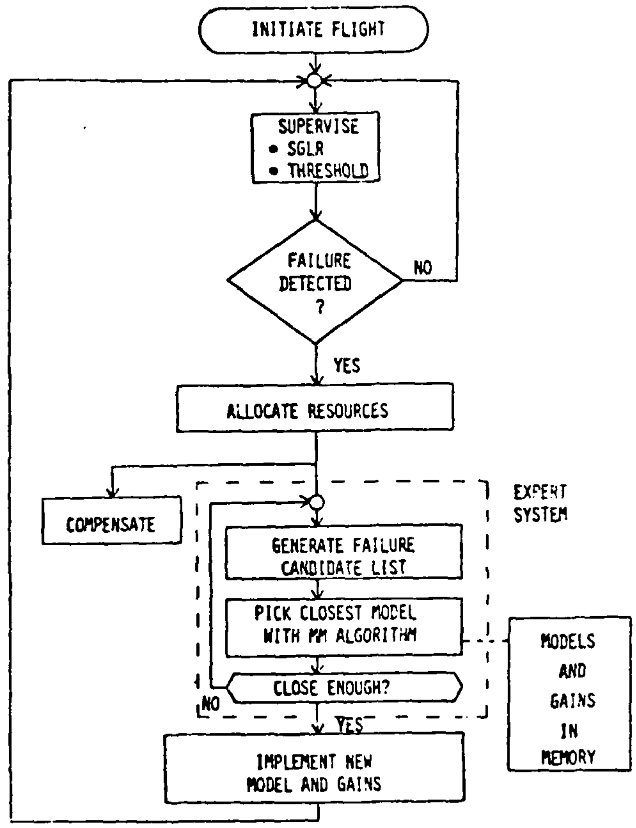


Figure 8: FDIR with Analytical Redundancy and AI - Flow Chart

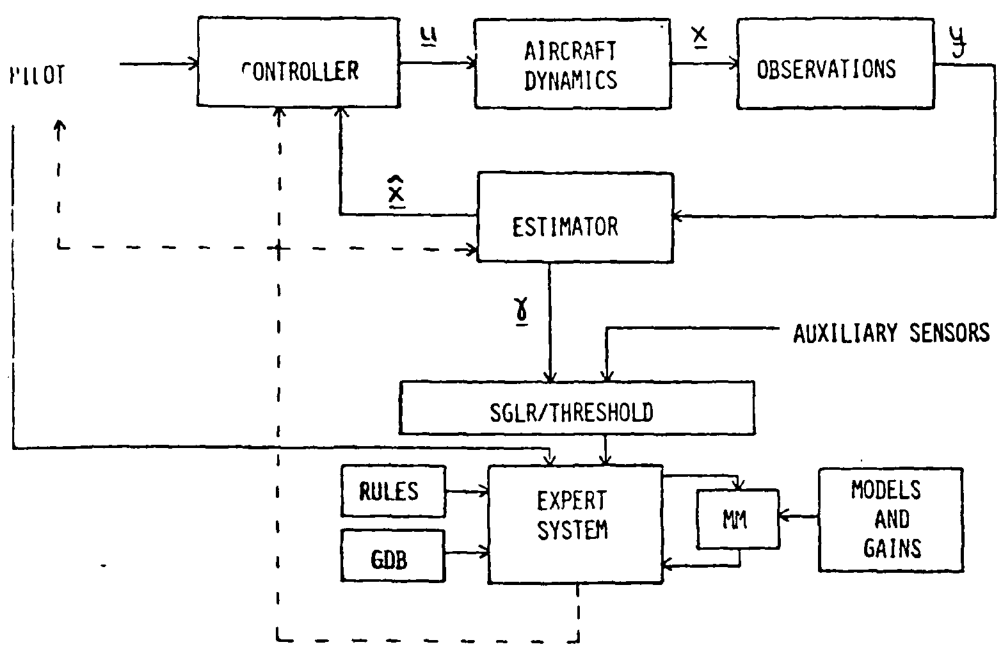


Figure 9: Knowledge-Based Reconfigurable Flight Control System

9. EXPERIMENTAL SETUP AND PRELIMINARY RESULTS

Initial development has been carried out using a microcomputer implementation of the KRFCS. A 4-MHz Z-80A 8-bit microprocessor, housed on a Multibus-compatible Monolithic Systems 8009 board, provided the processing power. Other features included an AMD9511 32-bit floating point math chip, a 64K CP/M operating system, and a dual 8-in flexible disk drive unit capable of holding 932 Kbytes of data. Pascal/MT+, a microprocessor version of Pascal, was chosen as the initial development language.

Preliminary system testing required software capable of performing 3 functional tasks: data preparation, flight simulation, and KRFCS operation. The separate software modules that were used to accomplish these tasks are shown in Fig. 10. After a nominal state-space model of the aircraft was derived, the user interactively generated models corresponding to distinct aircraft failures with the Failure Model Generator. The Gain Calculator then computed linear quadratic regulator and Kalman filter gains for each failure model. These were to be used by the KRFCS controller and estimator. Additionally, the rules and global data base were transformed into Pascal code by the Rule and GDB Encoders.

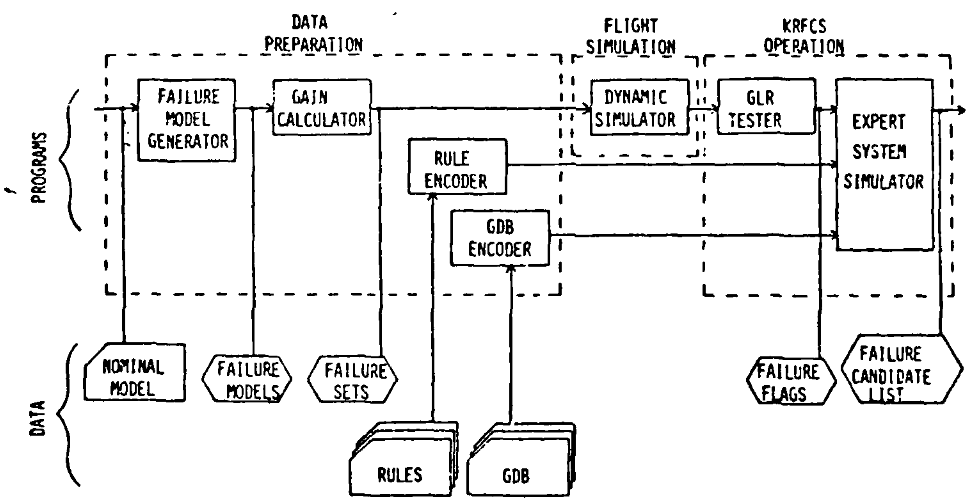


Figure 10: Experimental Tools

KRFCS operation was carried out with the Dynamic Simulator. This program received the nominal model and a failure model as input. It performed a deterministic linear simulation of the nominal aircraft model until the user induced the failure. When the GLR Tester picked up the model switch, it declared a failure and set failure flags to be read by the Expert System Simulator. With the failure flags, global data base, and rules, the Expert System Simulator showed the user each step of the diagnostic process. Its output was a Failure Candidate Scoreboard which gave failure type and component scores, indicating which failure models should have been tested by the multiple-model algorithm. The MM portion of the system had not yet been constructed.

These preliminary tests have provided useful information concerning the best performance to be expected from the GLR test, as well as its required computation time. It was found that in a deterministic setting the simplified generalized likelihood ratio test was very good at failure detection. Detection usually occurred in one sampling interval of 0.1 sec, even with failures involving parametric model changes for which the GLR test is not intended.

The simplified GLR test reduced computation time by assuming that specific bias jumps had occurred. However, the preliminary implementation was far too slow for real-time execution. A 3-sec moving window, which looked back in time for signs that a failure had occurred, required 30 secs computation time per 0.1 sec sampling interval. Although 300 times too slow with the present hardware, this detection approach will run much faster using newer microprocessors, parallel processing, and revised software.

10. CONCLUSIONS

A knowledge-based flight control system capable of detecting, identifying, and reconfiguring for a wide range of aircraft failures has been designed. Analytical redundancy techniques, including a Generalized Likelihood test, are used for failure detection. Failure diagnosis is performed by an expert system. Utilizing knowledge of cause-and-effect relationships between all aircraft components and the statistical results of a Multiple-Model algorithm, the expert system decides which aircraft component has failed and how to reconfigure for the failure.

Results of preliminary tests indicate that effective failure detection within a deterministic environment can be obtained with the analytical methods proposed. Additionally, the expert system can identify simple failures with its very limited knowledge base. However, many modifications to the expert system remain to be made. The system is more properly called a production system, in that when a failure is detected, the 16 diagnostic rules are fired in succession. No rule interpreters presently exist to provide a search control structure. Future versions will include this inference engine, thus giving the KRFCS a true expert system.

Most of the work performed to date has involved development of the basic idea behind the KRFCS and the utility routines needed to build it. Future software to be incorporated into the existing system includes coding of the Multiple-Model algorithm and the communication module linking it with the expert system. However, because memory requirements will soon outgrow the capabilities of an 8-bit machine, the next few research objectives are centered around the transition to 16-bit machines. The work schedule can be outlined chronologically as follows.

Adapt KRFCS to distributed processing environment

> Break the KRFCS into functional modules. Minimize the required amount of inter-module communication and standardize communication protocol. Modular design will permit the implementation of different languages, such as using LISP for the expert system.

Select implementation hardware and software language

> Determine number and type of boards needed for modular KRFCS. Choose operating system and run-time software package.

Taylor existing software to new machine

> Implement modular KRFCS architecture by modifying existing programs. Translate all system software into new run-time language.

Construct stochastic nonlinear simulator

> Simulation of aircraft dynamics including failures is to be performed on a machine physically independent of the KRFCS, e.g., an analog or general-purpose digital computer.

Hybrid simulations and software development

> Develop KRFCS components, including expert system GDB, rules, and rule interpreters, within the realistic environment provided by the nonlinear simulator.

Important issues need to be addressed in the near future. These include diagnosis of complicated failures, false alarm rate, overall speed, transient response of the aircraft, and varying nominal flight conditions. However, if these problems can be resolved, the KRFCS will become an intelligent and valuable mechanism capable of accomodating failures a pilot may not be able to handle alone.

REFERENCES

1. Folkesson, K., "Failure Management for the Saab Viggen JA37 Aircraft", in "Fault Tolerance Design and Redundancy Management Techniques", AGARD-LS-109, directed by Cunningham, T.B., Sept 1980.

2. Szalai, K. J., Larson, R. R., and Glover, R. D., "Flight Experience with Flight Control Redundancy Management", AGARD-LS-109, Sept 1980.

3. McMahan, J., "Delta Flight 1080 Story", Appendix A of "Restructurable Controls", NASA-CP-2277, compiled by Montoya, F.J., Howell, W.E., Bundick, W.T., Ostroff, A.J., Hueschen, R.M., and Belcastro, C.M., Sept 1982.

4. "American Airlines DC-10 Crash in Chicago", Excerpts from NTSB-AAR-79-17, Appendix B of NASA-CP-2277, Sept 1982.

5. Clark, R. N., Fosth, D. C., and Walton, V. M., "Detecting Instrument Malfunctions in Control Systems", IEEE Transactions on Aerospace and Electronic Systems, Vol AES-11, No 4, July 1975.

6. Friedland, B., "Maximum Likelihood Failure Detection of Aircraft Flight Control Sensors", J. Guidance, Vol 5, No 5, Sept-Oct 1982.

7. Willsky, A. S., "Failure Detection in Dynamic Systems", AGARD-LS-109, Sept 1980.

8. Folkesson, K., "Computer Based In-Flight Monitoring", AGARD-LS-109, Sept 1980.

9. Cunningham, T. B., "Failure Mangement Techniques for High Survivability", AGARD-LS-109, Sept 1980.

10. Stengel, R. F., and Berry, P.W., "Stability and Control of Maneuvering High-Performance Aircraft", NASA-CR-2788, April 1977.

11. Montoya, F.J., Howell, W.E., Bundick, W.T., Ostroff, A.J., Hueschen, R.M., and Belcastro, C.M., "Restructurable Controls", NASA-CP-2277, Sept 1982.

12. Hayes, J. E., and Michie, D., Intelligent Systems: The Unprecedented Opportunity, Halsted Press, New York, 1983.

Appendix A

STATE SPACE FORMULATION

The controller and estimator of the fly-by-wire flight control system contain gains derived from a linear mathematical model of the aircraft. The Generalized Likelihood Ratio and Multiple-Model algorithms of the KRFCS utilize linear models as well. Linearization begins with a nonlinear model representing the aircraft kinematics and dynamics as shown below. Note that the matrix notation uses subscripts and superscripts to relate inertial axes and body axes where appropriate.

NONLINEAR EQUATIONS OF MOTION

TRANSLATIONAL KINEMATICS	$\dot{\underline{x}}_I = H_B^I \underline{v}_B$
ROTATIONAL KINEMATICS	$\dot{\underline{\nu}}_B = L_B^{-1} \underline{\omega}_B^I$
TRANSLATIONAL DYNAMICS	$\dot{\underline{v}}_B = (\underline{F}_B + \underline{T}_B)/m + H_I^B \, g_I - \tilde{\omega}_B^I \underline{v}_B$
ROTATIONAL DYNAMICS	$\dot{\underline{\omega}}_B^I = I_B^{-1} (\underline{M}_B + \underline{G}_B) - I_B^{-1} \tilde{\omega}_B^I I_B \underline{\omega}_B^I$

INERTIAL POSITION VECTOR

$$\underline{x}_I = \begin{bmatrix} x_I \\ y_I \\ z_I \end{bmatrix}$$

EULER ANGLE VECTOR

$$\underline{\nu}_B = \begin{bmatrix} \Phi \\ \Theta \\ \psi \end{bmatrix}$$

BODY-AXIS TRANSLATIONAL RATE VECTOR

$$\underline{v}_B = \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

BODY ANGULAR RATE VECTOR

$$\underline{\omega}_B^I = \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

STATE EQUATION FORM

$$\dot{\underline{x}} = f(\underline{x}, \underline{u}) \qquad \underline{x} = \begin{bmatrix} \underline{x}_I \\ \underline{\nu}_B \\ \underline{v}_B \\ \underline{\omega}_B^I \end{bmatrix}$$

The nonlinear nominal and linear perturbation equations are obtained by expanding the state equation in a Taylor series about some nominal trajectory. Because the flight control computer can move the control surfaces only at discrete instances in time, a zero-order hold with fixed sampling interval is assumed. A linear discrete-time state equation results.

LINEAR EQUATIONS OF MOTION

$$\dot{\underline{x}} = \dot{\underline{x}}_0 + \Delta\dot{\underline{x}} = \underline{f}(\underline{x}_0, \underline{u}_0) + \frac{\partial f}{\partial \underline{x}}\bigg|_{\substack{\underline{x}=\underline{x}_0 \\ \underline{u}=\underline{u}_0}} \Delta\underline{x} + \frac{\partial f}{\partial \underline{u}}\bigg|_{\substack{\underline{x}=\underline{x}_0 \\ \underline{u}=\underline{u}_0}} \Delta\underline{u} + \text{H.O.T.}$$

NOMINAL TRAJECTORY

$$\dot{\underline{x}}_0 = \underline{f}(\underline{x}_0, \underline{u}_0)$$

PERTURBATIONS ABOUT NOMINAL TRAJECTORY

$$\Delta\dot{\underline{x}} = F\,\Delta\underline{x} + G\Delta\underline{u}$$

SAMPLED DATA SYSTEM

ASSUME FIXED SAMPLING INTERVAL Δ AND PIECEWISE CONSTANT INPUT:

$$\underline{u}(t) = \underline{u}(k),\ k\Delta \le t \le (k+1)\Delta$$

DISCRETE-TIME STATE EQUATION

$$\underline{x}(k+1) = \Phi\underline{x}(k) + \Gamma\underline{u}(k)$$

$$\Phi = \text{EXP}\,(A\Delta)$$

$$\Gamma = \int_0^{\Delta} [\text{EXP}(A\tau)]\,d\tau \cdot G$$

The state-space model is composed of the state equation and observations provided by flight sensors. In order to closely represent the dynamics of an aircraft, however, deterministic biases and noise sequences must be added. Noise enters the state equation through disturbances such as turbulence, and biases enter through failures that cause parametric model changes. Similarly, sensor failures cause noise and bias changes in the observation equation.

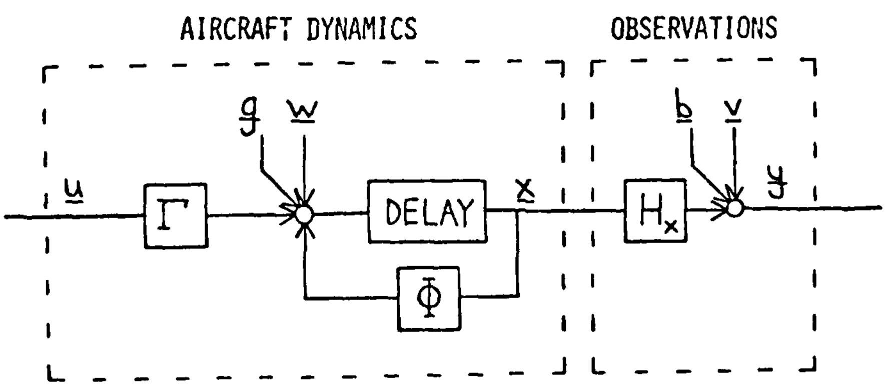
STOCHASTIC DYNAMIC SYSTEM

STATE EQUATION

$$\underline{x}(k+1) = \Phi \underline{x}(k) + \Gamma \underline{u}(k) + \underline{g}(k) + \underline{w}(k)$$

OBSERVATION EQUATION

$$\underline{y}(k) = H_x \, \underline{x}(k) + \underline{b}(k) + \underline{v}(k)$$



The controller steady-state feedback gains can be derived by any pole placement technique, including linear quadratic regulator theory. Feedforward gains associated with pilot inputs can be computed from the desired non-zero set point and equilibrium response. The state estimator, on the other hand, is a Kalman Filter. It first predicts the present state by propagating the state equation to the next sampling interval. Then, it subtracts the measured observations from those that would result from the predicted state. Finally, the filter uses this new information to update the state prediction. In this way, the filter helps reduce the effect of modeling errors on state estimation accuracy.

CONTROLLER AND ESTIMATOR

CONTROLLER EQUATION

$$\underline{u}(k) = -C\,\hat{\underline{x}}(k|k) + C_p \underline{u}_p(k)$$

ESTIMATOR EQUATIONS (KALMAN FILTER)

PREDICTED ESTIMATE $\hat{\underline{x}}(k+1|k) = \Phi\,\hat{\underline{x}}(k|k) + \Gamma\,\underline{u}(k) + \underline{g}(k)$

INNOVATIONS PROCESS $\underline{\gamma}(k+1) = \underline{y}(k+1) - H_x\,\hat{\underline{x}}(k+1|k) - \underline{b}(k)$

FILTERED ESTIMATE $\hat{\underline{x}}(k+1|k+1) = \hat{\underline{x}}(k+1|k) + K\,\underline{\gamma}(k+1)$

ESTIMATOR GAINS CALCULATED OFF-LINE FROM

$$P(k+1|k) = \Phi P(k|k)\Phi^T + Q$$

$$V(k+1) = H_x P(k+1|k)\,H_x^T + R$$

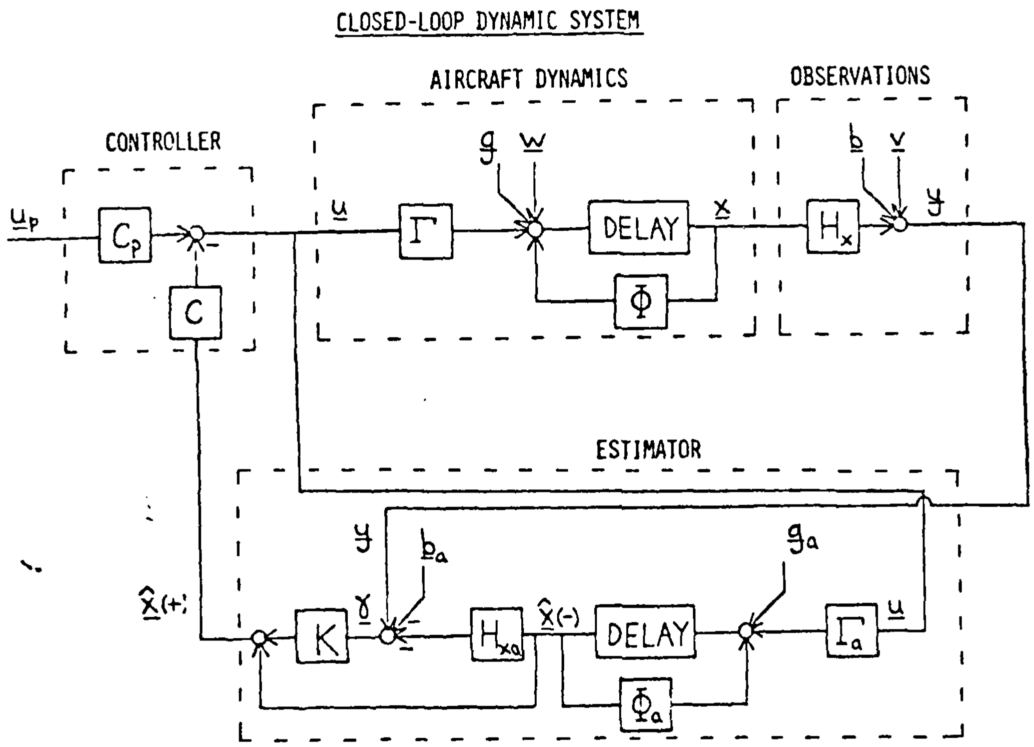
$$K(k+1) = P(k+1|k)\,H_x^T\,V^{-1}(k+1)$$

$$P(k+1|k+1) = P(k+1|k) + K(k+1)\,H_x\,P(k+1|k)$$

WHERE

$P(k|k)$ = COVARIANCE OF ERROR $\underline{x}(k) - \hat{\underline{x}}(k|k)$
$V(k)$ = COVARIANCE OF ZERO-MEAN, WHITE GAUSSIAN INNOVATIONS $\underline{\gamma}(k)$
K = ESTIMATOR GAINS

When the controller and estimator are combined with the original linear stochastic system, a closed-loop dynamic system is produced. This closed-loop system forms the basis for KRFCS development.



Appendix B

FAILURE DIAGNOSIS RULES

The KRFCS expert system uses rules and a global data base to diagnose aircraft failures. It narrows down to a reasonable number the candidate failures to be tested by the multiple-model algorithm. The following sixteen rules were used in preliminary expert system testing. In order to illustrate how the rules use "scratch pad" memory stacks and the SET and PROCEDURE features of Pascal during failure diagnosis, the source code for Rule 14 is included.

Diagnosis Through Validity

Rule 5: Flight Sensor Soft Validity via
Observations, Sensors, and States

IF an observation is flagged (from GLR test)
AND no flight sensors are flagged (from threshold test)
AND no auxiliary sensors are flagged (from threshold test)
AND no states are flagged (from the GLR test)

THEN a flight sensor soft failure (i.e. bias drift)
is likely

Rule 6: Flight Sensor Hard Validity via Observations and Sensors

IF an observation is flagged
AND a flight sensor used for that observation is flagged
AND all other sensors are not flagged

THEN a flight sensor hard failure is likely

Rule 7: Control Sensor Flag Status

IF an auxiliary sensor is flagged
AND that sensor senses a control

THEN an auxiliary sensor failure is likely
AND a support failure is likely

Rule 8: Effector Sensor Validity via States

IF an auxiliary sensor is flagged
AND that sensor senses an actuator/effector
AND no states are flagged

THEN an auxiliary sensor failure is likely
AND a support failure is likely

Rule 9: Structural and State Disturbance Validity via Sensors

IF a state is flagged
AND the flight sensors are OK
AND the auxiliary sensors are OK

THEN a state disturbance failure is likely
AND a structural failure is likely

Rule 10: Effector Validity via Sensors and States

IF a state is flagged
AND the flight sensors are OK
AND auxiliary sensors that sense actuators/effectors effecting the flagged state exist and are flagged

THEN an actuator/effector failure is likely

Rule 15: Common Support Validity via Sensors and Effectors

IF the support common to all the flagged sensors supports no non-flagged sensor
AND it either supports no actuator/effector
OR it supports an actuator that effects a flagged state

THEN a support failure is likely

Rule 16: Common Location Validity via Sensors and Effectors

IF the location common to all the flagged sensors contains no non-flagged sensor
AND it either contains no actuator/effector
OR it contains an actuator/effector that effects a flagged state

THEN a structural failure is likely

Diagnosis Through Contradiction

Rule 1: Flight Sensor Flag Status

IF no observations are flagged
AND no flight sensors are flagged

THEN a flight sensor failure is unlikely
AND the flight sensors are OK

ELSE the flight sensors are not OK

Rule 2: Auxiliary Sensor Flag Status

IF no auxiliary sensors are flagged

THEN an auxiliary sensor failure is unlikely
AND the auxiliary sensors are OK

ELSE the auxiliary sensors are not OK

Rule 3: State Disturbance Flag Status

IF no states are flagged

THEN a structural failure is unlikely
AND a state disturbance failure is unlikely
AND an actuator/effector failure is unlikely
AND the states are OK

ELSE the states are not OK

Rule 4: State Disturbance Contradiction via Sensors and Observations

IF a state is flagged
AND an observation is flagged
OR a flight sensor is flagged
OR an auxiliary sensor is flagged

THEN a state disturbance failure is unlikely

Rule 11: **Common Support Contradiction via Sensors**

IF the support common to all the flagged sensors supports a non-flagged sensor

THEN a support failure failure is unlikely

Rule 12: **Common Support Contradiction via Effectors**

IF the support common to all the flagged sensors supports an actuator/effector that effects a a non-flagged state

THEN a support failure is unlikely

Rule 13: **Common Location Contradiction via Sensors**

IF the location common to all the flagged sensors contains a non-flagged sensor

THEN a structural failure is unlikely

Rule 14: **Common Location Contradiction via Effectors**

IF the location common to all the flagged sensors contains an actuator/effector that effects a non-flagged state

THEN a structural failure is unlikely

```
procedure Rule14;

   begin
      if  ((anyFlagged(fltSensors) or anyFlagged(auxSensors))
            and common(locations,fltSensors,forwrd,0)
            and common(locations,auxSensors,forwrd,1)
            and areRelatedTo(actEffectors,locations,backwrd)
            and areRelatedTo(states,actEffectors,backwrd)
            and not(stackFlagged(states)))
          then
             begin
                ruletrue:= true;
                score(strucFail,-1)
             end
   end;   {  Rule 14  }
```

Forward chaining search (looking for IF part to be true)
checks if the "condition set" contains these set elements:

```
@fltSNotOK     { flight sensors not OK }
@auxSNotOK      { auxiliary sensors not OK }
```

Backward chaining search (trying to prove THEN part is true)
checks if the "action set" contains this set element:

```
@strucScore   { structural failure score affected }
```

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A165 682 N/A	3. RECIPIENT'S CATALOG NUMBER N/A
4. TITLE (and Subtitle) An Application of Artificial Intelligence Theory to Reconfigurable Flight Control.		5. TYPE OF REPORT & PERIOD COVERED Interim Technical Report Jan. 1/84 to June 30/84
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) David A. Handelman		8. CONTRACT OR GRANT NUMBER(s) DAAG29-84-K-0048
9. PERFORMING ORGANIZATION NAME AND ADDRESS Princeton University MAE Dept. D-202 Engineering Quadrangle Princeton, NJ 08544		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS N/A
11. CONTROLLING OFFICE NAME AND ADDRESS U. S. Army Research Office Post Office Box 12211 Research Triangle Park, NC 27709		12. REPORT DATE June 30, 1984
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

NA

18. SUPPLEMENTARY NOTES

The view, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Artificial Intelligence; Control Systems; Flight Control; Cybernetics; Computer Systems; Systems Analysis.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A knowledge-based flight control system capable of detecting, identifying, and reconfiguring for a wide range of aircraft failures has been designed. Combining modern control theory, statistical hypothesis testing, and artificial intelligence techniques, this research addresses the question of whether or not an "intelligent" computer could assist a pilot during a failure. Analytical Redundancy techniques, including a Generalized Likelihood test, are used for failure detection. Failure

DD FORM 1473 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. ABSTRACT CONTINUED

diagnosis is performed by an expert system. Utilizing knowledge of cause-and-effect relationships between all aircraft components and the statistical results of a Multiple-Model algorithm, the expert system decides which aircraft component has failed and how to reconfigure for the failure. Preliminary tests on a 8-bit microprocessor system were conducted and are summarized, and plans to expand to a 16-bit multi-microprocessor system are outlined.

END

FILMED

4-86

DTIC