NASA Technical Memorandum 4242

Flight Control System Design Factors for Applying Automated Testing Techniques

Joel R. Sitz and Todd H. Vernon

OCTOBER 1990

NASA

NASA Technical Memorandum 4242

Flight Control System Design Factors for Applying Automated Testing Techniques

Joel R. Sitz
Ames Research Center
Dryden Flight Research Facility
Edwards, California

Todd H. Vernon
PRC Systems Service
Edwards, California

National Aeronautics and
Space Administration
Office of Management
Scientific and Technical
Information Division

1990

FLIGHT CONTROL SYSTEM DESIGN FACTORS FOR APPLYING AUTOMATED TESTING TECHNIQUES

Joel R. Sitz
NASA Ames Research Center
Dryden Flight Research Facility
P.O. Box 273
Edwards, CA 93523-0273
and
Todd H. Vernon
Planning Research Corporation
P.O. Box 273
Edwards, CA 93523-0273

Abstract

Automated validation of flight-critical embedded systems is being done at the National Aeronautics and Space Administration Ames Research Center Dryden Flight Research Facility. The automated testing techniques are being used to perform closed-loop validation of man-rated flight control systems. This paper discusses the principal design features and operational experiences of the X-29 forward-swept-wing aircraft and F-18 high alpha research vehicle (HARV) automated test systems. Operationally applying automated testing techniques has accentuated flight control system features that either help or hinder the application of these techniques. The paper also discusses flight control system features which foster the use of automated testing techniques.

Introduction

Ames Research Center Dryden Flight Research Facility (Ames-Dryden) is researching the application of automated testing techniques for the verification and validation of man-rated flight control systems (FCSs). Automated testing techniques were applied to the X-29 forward-swept-wing aircraft (Fig. 1) and the F-18 high alpha research vehicle (HARV) (Fig. 2) to reduce the time required to reliably validate the control system.

Automated testing techniques are being developed because of the increasing cost of flight qualifying embedded systems. Relaxed static stability, supermaneuverability, and the optimization of handling characteristics have resulted in complex FCSs. Complexities will increase as manned hypersonic vehicles require vehicle management systems using distributed processing techniques. These techniques will integrate vehicle controls, propulsion, thermal management, hydraulic management, electrical load management, and mission planning. The number of test cases necessary to prove system dependability will increase exponentially. For example, the triplex redundancy management logic for the X-29 aircraft has approximately 90 inputs. If test cases of normal, null, and extreme failures are run, then the number of test cases necessary to completely validate the redundancy management logic would be 3^{90}. Given an average of 15 min to configure and run a failure modes and effect test, the test cases would take approximately 2.49×10^{38} years to finish! Clearly, no system can be completely validated. New techniques must be developed to run more test cases in the same time. As the amount of test data generated increases, new methodologies for extracting information must also be developed. Embedded system features, which can act as catalysts for a more efficient validation process, can be used to complement these developments.

Government agencies and private industry are currently demonstrating automated testing techniques for all phases of the flight qualification process. Automated testing techniques are being used on the X-31 enhanced fighter maneuverability (EFM) program[1] as well as military advanced fighter demonstration and aircraft production programs. Ames-Dryden's traditional verification and validation techniques[2] have been developed from flight qualification experiences on several experimental flight research vehicles, including the F-8C digital fly-by-wire aircraft control system, the highly maneuverable aircraft technology (HiMAT) vehicle, the advanced fighter technology integration (AFTI) F-16 aircraft, the X-29 aircraft, and the F-18 HARV aircraft. This experience provides the cornerstone for developing advanced testing techniques. Ames-Dryden represents a unique environment for this type of research because of the diversity of embedded research systems which are flight qualified at the facility. The first digital fly-by-wire control system development and test effort used automated testing techniques. A software support package integrated with the F-8C aircraft simulation introduced automated testing techniques for the redundancy management logic.[3] Similar techniques were used on the HiMAT program during the late 1970's.[4] Ames-Dryden is currently developing automated testing techniques to reduce testing costs and increase the availability of an aircraft for flight. Ames-Dryden is constructing an integrated test facility (ITF)[5] to develop ad-

vanced flight qualification technology emphasizing aircraft-in-the-loop techniques.

Enhancements to the FCS's testability must accompany the developing automated testing techniques. With current computing technology, on-aircraft central processing unit (CPU) speeds of 20 million instructions/sec are possible using very large scale integration (VLSI) based 32-bit microprocessors. These speeds will offset the additional software overhead typically associated with higher order languages and will allow embedded features to improve target system and automated testing system integration.

Nomenclature

AC	alternating current
CPU	central processing unit
DC	direct current
FCC	flight control computer
FCS	flight control system
FS/CP	failure status/control panel
GUI	graphical user interface
HARV	high alpha research vehicle
ITF	integrated test facility
LED	light emitting diode
RFCS	research flight control system
RISC	reduced instruction set computer
SEU	system evaluation unit
SIH	simulation interface handler
STIL	system test interface language
UMN	universal memory network

The X-29 Forward-Swept-Wing Automated Testing System

The X-29 FCS, described in Ref. 6, used off-the-shelf hardware. The operational flight program was written in a processor-specific assembly language and consisted of 205 total modules, approximately 29,000 instructions, 220 variables, and 3,000 constants.[7] During aircraft ground testing for the second X-29 aircraft, automated testing techniques reduced the time required for aircraft FCS verification and validation tests from 4 weeks to 7 days. This was a labor savings of more than 8 man months and allowed the aircraft to be flown 3 weeks earlier than would have been possible with conventional test techniques.[8]

No modifications were made to the X-29 FCS software or hardware to assist in applying the automated testing techniques. Validation of control systems in a state ready for flight is desirable. Frequently performed verification and validation tests were the primary focus of automation, including time history, frequency response, and input/output control system checks. These tests were run for all FCS changes. System engineers and control engineers, not automated techniques, decided what should be tested.

A primary goal in developing automated testing techniques at Ames-Dryden was identifying open-systems technology that promotes a generic approach to closed-loop validation of uniquely configured embedded systems. Several different testing configurations are used for validating embedded systems at Ames-Dryden. The primary simulation configurations used for testing the X-29 FCS included modeled aircraft dynamics integrated with the actual FCS hardware.

Principal Design Features

The X-29 automated test system (Fig. 3) was designed to help verify and validate the FCS software. It was not designed to flight qualify aircraft systems using distributed processing techniques such as the F-18 HARV. The primary elements were a nonlinear real-time aircraft simulation with data recording capabilities, a Unix® workstation for simulation command file generation, the X-29 flight control computers (FCCs), and a hardware actuator model. The real actuators were used during aircraft-in-the-loop testing.

Minor software modifications to the X-29 aircraft simulation were required. When a change was made, it was designed to be easily incorporated into other simulations. Control of the simulations was automated by allowing the simulation executive to automatically read test command files instead of a tester manually typing simulation commands at a keyboard. The similarities in the command-line user interfaces inherent in all Ames-Dryden aircraft simulations made automated test system development for different aircraft possible. Ames-Dryden had adopted a standard command line interface to minimize development costs and training time when new simulations were required. The automated testing system benefited from this standardized user interface approach. The simulation commands set internal simulation variables. In some cases, commands used cumbersome and nondescriptive array variables internal to the simulation. Consequently, the test engineer needed to maintain an extensive working knowledge of the simulation software implementation.

An advantage of the X-29 automated test system was the ability to hide simulation software mechanisms by providing a higher order interface. This was accomplished with the system test interface language (STIL). Test procedures written in STIL were translated into simulation commands by the STIL translator. The STIL translator was written in the "C" programming language and was built around the Unix utility M4 C language macro preprocessor. The STIL translation was executed on the Unix workstation. The output of the translator was a file of valid simulation commands that

® Unix is a registered trademark of AT&T, Bell Laboratories, Murray Hills, New Jersey.

were transferred to the simulation computer and read by the simulation executive to run a test. The simulation software executive was optimized to increase the speed at which the new command files could be read.

Signal generation code was added to the X-29 aircraft simulation program to produce a variety of signal shapes. Frequency sweeps, square waves, biases, doublets, and null values could replace or be summed on to the simulation analog outputs to the FCCs. User-definable attributes such as signal amplitudes, pulse widths, pulse durations, and frequencies could be specified by simulation commands in the test command file. Test discrete outputs could be substituted for simulation discrete outputs. The flexibility on type of signal and when it was introduced to the FCS was useful.

Test results were interpreted primarily by comparison analysis to baselined data. The actual FCS was compared to an independently implemented nonlinear control system simulation. Automatically recording the test data results in real time, transfering the results to an appropriate analysis computer, and generating an overplot of actual and expected results were desired. The plotting process was optimized to produce several overplots with minimal user intervention. FORTRAN graphics routines were the basis for the plotting application. Plotting control files which governed the plotting process were automatically generated and used to drive the output plotting device. Other methods of interpreting the test results included capturing bit-packed control system status words, transfering the data to a Unix workstation, and automatically comparing them to expected results. A pass–fail message was generated for each test case to allow for quick scanning of the test results.

Development

The development process for the X-29 automated test system was to design a prototype system using a specific example of a typical verification test. Useful aspects of the prototype would then be carried forward to an operational phase. In May of 1987, a prototype automated test system was demonstrated by Ames-Dryden.

The design philosophy for the demonstration system was to provide a front end to the X-29 hardware-in-the-loop simulation (Fig. 4) with a Unix workstation. A relational database management system running on the workstation was the primary interface between the user and the automated test environment. The workstation provided a menu-driven user interface for test generation, control operation, results processing, and test documentation archiving. The relational database menu items were chosen with several combinations of keyboard control sequences. Communication between the Unix workstation and the real-time simulation was a standard RS-232 link. The goal was centralized, fully automated control of the entire test process.

The X-29 demonstration system was designed to automatically run an open-loop frequency response test across the flight control computers (FCCs) and display the results plotted against predetermined gain and phase margin limits. A Unix workstation process called the simulation interface handler (SIH) provided overall management of the test sequencing. Once a STIL test procedure was translated into the corresponding simulation commands, the SIH would send the command file to the simulation computer. The simulation computer would read the test command file to run the test, record the data, convert the data to an ASCII format, and transfer the data back to the Unix workstation. The SIH then initiated the data analysis routines and presented the user with a comparison of actual gain and phase data overlayed with the predetermined limits. All of this was controlled from a single menu-driven user interface, minimizing user interaction.

Operational Experiences

The automated testing technology demonstration for the X-29 aircraft highlighted the practical constraints of fully automating a process with equipment that was not designed for automated control. These practical constraints transformed the desired fully automated design into a highly interactive design. Several aspects of the X-29 automated test system did not represent a practical solution in an operational environment. For instance, the RS-232 connection between the real-time simulation and the Unix workstation was extremely slow for transferring large (1 Mbyte) data files. The RS-232 was the only option at the time of the demonstration.

Another constraint was using a relational database for a user interface. Selecting menu choices, traversing menu pages, and completing the database forms used for writing test procedures was accomplished with cumbersome control sequences typed on a keyboard. Graphical user interface (GUI) standards were not readily available. This lack of GUI standards prevented GUI development efforts, since the results would not be portable to emerging high-speed Unix workstations. Use of the database was not carried forward to an operational phase.

Practical constraints of the X-29 FCS also became apparent. These constraints included limitations associated with automating pilot switch actions on the failure status/control panel (FS/CP). The FS/CP (Fig. 5) was the pilot's interface to the FCCs. This panel was used to reset the digital computers, reset or arm the control system actuators, enter discrete flap positions, initiate built-in test sequences, and manually change control system modes. The interface between the FS/CP and the FCCs was a custom designed 1 MHz serial bus. Unfortunately, this bus had no provisions for external interfaces. Consequently, typical pilot actions using the FS/CP could not be completely automated. Automatically resetting and arming actuators and selecting flap positions with a thumbwheel were not possible and remained manual operations during testing. Since there were several discrete

flap position combinations associated with the X-29 control system, lengthy validation tests resulted.

Minimal hardware changes to the simulation were required to allow FCS mode changes to be automated. Changing control system modes from primary to backup and resetting the FCCs was successfully automated. The computer interface console (CIC), which interfaced the FCCs to the hardware-in-the-loop simulation test equipment, performed signal conversions such as DC to AC, DC to synchro, and low-voltage to the high-voltage discretes used on the aircraft by the FCCs. With simple hardware modifications to the CIC, the automated test system could drive relays emulating the physical FS/CP switch movement for mode changing and FCC resetting. A simulation command file could then be used to reset the FCCs. Because the FCCs cleared all internal random access memory (RAM) locations on startup, automated FCC reinitialization between consecutive test runs was possible.

The X-29 FCC CPUs were interfaced on the ground with a system evaluation unit (SEU) used to debug control system software. The FCC memory data and processor register values could be monitored in real time on the SEU front panel LED display. The data monitoring was limited to a single, user-selectable, internal FCC parameter and was not useful for analyzing relationships between parameters across different FCC channels. Hard copy results from the SEU interface of internal FCC variables were only available by capturing data dumps while the CPUs were stopped.

The primary method of viewing internal FCC variables in real time was a 64-word ARINC 429 bus. FCC software modifications were necessary to change which values were loaded on the bus. The 429 bus parameters were captured by an extended aircraft interrogation and display system (XAIDS)[9] and relayed to the simulation computer by a 1553 bus for display on strip charts.

Controling the simulation with command files was the most successful aspect of the X-29 automated testing development. The STIL interface preserved the manual user interface already in operation and increased the user's ability to write several similar tests quickly, accomplishing validation in a shorter period of time. The STIL, however, did not deliver all the capabilities normally associated with a programming language and had very limited mathematical capability with no looping features or other useful control constructs.

The X-29 automated test system acted in an open-loop fashion. There were no real-time feedbacks from the testing environment to provide closed-loop control of the test process. Consequently, no error recovery from test system hardware failures, or other erroneous situations, existed. The simulation computer executive would not halt the reading of a test command file once a test began executing but always attempted to finish executing a test command file regardless of the status of the testing environment.

A good example of the disadvantages of open-loop operation occurred during the X-29 aircraft-in-the-loop testing. A command file controlling the simulation commanded a change in flight condition. The simulation responded immediately and proceeded with commands which began exciting the FCS at the specified flight condition. However, manually controlled airdata test equipment connected to the aircraft's pitot-static system had not been adjusted to the desired flight condition before control system excitation began. The simulation computer started executing the test before the FCS had internally selected the appropriate gain set. Erroneous control system gains resulted in an aircraft limit cycle, and manual safety precautions were used to discontinue the test. This problem could have been avoided if the simulation executive had halted test execution until the FCS was properly initialized. To do this, feedbacks from the FCS confirming that desired flight conditions had been reached would be needed.

The F-18 High Alpha Research Vehicle Automated Test System

Principal Design Features

The F-18 HARV automated test system is meant to improve on the X-29 automated testing environment features. The development of a test language, closed-loop control of the testing environment, graphical user interfaces, and quick-look monitoring displays are being emphasized. The design will attempt to improve automation of the real-time aircraft simulation control and also to add features which automate the tester's decision-making capabilities. The F-18 HARV automated testing system is being partitioned into four general sections: test generation, test engine, test monitoring, and test data analysis.

Ames-Dryden is flying the F-18 HARV to perform high-angle-of-attack flight research. The basic FCS has been extensively modified. A research flight control system (RFCS), implemented with Ada® and interfaced to the basic FCS, will be used for aircraft thrust vectoring control. The F-18 HARV automated test system is currently being developed to help verify and validate both F-18 HARV control systems.

Open systems architectures and software standards are being followed when possible to insure portability to other computing platforms. The test concept for the F-18 HARV continues the integrated simulation and automated testing environment approach. An overview of the F-18 HARV hardware-in-the-loop simulation can be seen in Fig. 6. The X-29 automated test system did not address validation of concurrent processes such as those found in the F-18 HARV with the mission computer, FCC, and research FCC. The

® Ada is a registered trademark of the United States Government Department of Defense.

F-18 HARV testing methodologies will address the trends toward distributed processing vehicle management systems.

Initial Development

An interative development approach is being taken to the automated testing system. As useful features are designed and implemented, they are integrated with the F-18 HARV testing environment in a buildup fashion. This will insure usability and, more importantly, acceptance by the test engineers in an operational environment.

The integrated F-18 HARV automated test system and simulation (Fig. 7) consists of several computers interfaced to a high-bandwidth reflective memory network. Mainframe computers provide the functions of real-time input/output to the FCCs, data recording of all testing environment parameters, and the nonlinear aircraft simulation. The Unix RISC-based workstation provides test environment control and monitoring.

The reflective memory network or the universal memory network (UMN) connects computers such as mainframes and Unix workstations. It is a reflective memory system with composite rates of up to 40 Mbytes/sec. Processors connected to the UMN can effectively share a global memory partition with no special protocols or additional processor overhead. The UMN is currently operational in the F-18 HARV real-time simulation and is being used in conjunction with a high-speed data recording capability also developed for the ITF. The F-18 HARV automated testing environment will be designed around this high-speed memory network to overcome the data transfer, test monitoring, and test control deficiencies of the X-29 automated test systems.

Test generation is focused on improving the X-29 STIL concept of developing efficient methods of writing test procedures. The ability of the X-29 simulation executive to read command files was duplicated in the F-18 HARV simulation executive. Currently, a longer term solution to providing testers with a test language is being developed. The test language will support common higher order programming languages such as FORTRAN and C. For efficient use of the F-18 HARV testing environment, user libraries will provide the test engineer with access to automated testing features. These libraries, callable from common higher order languages, will hide the complexities of controlling the automated environment and will allow users to write test procedures to precisely control the validation process. This approach minimizes test language development time while providing a full set of programming control features. The test engine will compile the test procedure and produce the commands necessary to coordinate and control the automated test environment.

The test engine is a continuation of the X-29 simulation interface handler concept. The test engine will control real-time data recording, aircraft simulation, test monitoring, and any necessary real-time test data analysis processes. It is a high speed Unix workstation with a library of test functions used to obtain closed-loop control of the automated test environment. The test environment feedbacks will include real-time simulation values, FCS values, and operational status parameters from the automated testing environment. The test engine will also be used to provide the user interface. Figures 8(a) and 8(b) show the contrast between the manually controlled simulation and the closed-loop automated simulation control. The test engine hardware will be a RISC-based fileserver connected to a RISC-based Unix workstation providing the user interface. The test engine will be interfaced to the testing environment with the UMN.

The test monitoring and analysis functions will be driven by real-time current value tables (CVTs) located in the UMN reflected global memory partition. The design goal is to improve the quick-look capability at test data results and to provide several types of user-customizable displays in real time. This will provide the user with feedback to avoid rerunning lengthy tests because of set up errors or bad data. Various types of monitoring applications will be developed using commercial off-the-shelf applications. The F-18 HARV 1553 buses will be interfaced to the UMN to provide real-time information from the onboard FCSs and avionics. Because of the increases in Unix workstation computing performance over the last three years, implementing GUI standards is now feasible. The X-window-based applications are being chosen to provide a flexible multiwindow user interface.

The F-18 HARV test system is improving on the X-29 automated test system. Closed-loop control of the testing process will provide error recovery capability and give the necessary decision control to the test engine to better allow automation of failure modes and effects tests. The point and time when failures can be introduced can be controlled based on aircraft conditions. The automated test system signal generation modifications incorporated in the X-29 aircraft simulation program were designed to be easily transfered to the F-18 HARV simulation. While new F-18 HARV automated test system advancements are being developed, command file control of this signal generation code can be used in parallel with new developments.

Flight Control System Design Recommendations

Experiences with the X-29 automated testing capabilities have shown that more elegant approaches of combining the embedded system design and test requirements at earlier stages of development are needed. A major key to testability is participation of test personnel in the design process. If validation tools and techniques are identified during the initial stages of embedded system implementation, validation can be made easier. The target systems and the test systems must be considered one development effort. Well-structured top-down embedded system design with modu-

larity increases the ability to maintain the software. But validation is not made easier unless testability was considered during the design and coding stages. Higher order languages like Ada are an attempt to reduce software life cycle costs by increasing the readability and understandability of embedded code, but testability must also be addressed.

Nonreal-Time Considerations

Most features that would significantly increase testability are inexpensive to incorporate, but would require more discipline from the software engineering perspective. When validating a complex software system, online access to the information describing system implementation details and expected operations is needed. Often, design documentation is incomplete, easy to misunderstand, and difficult to piece together in a coherent fashion. Consistent, well-structured internal documentation of functional elements would allow validation tools to perform searches quickly to answer simple questions about how the embedded system should operate. For example, software off-the-shelf component data books should be established to complement the now emerging object-oriented programming (OOP) approaches. Data dictionaries are vital in the management of embedded system information. These dictionaries should describe all internal program variables, scale factors, maximum and minimum values, addresses, set and used information, bit packing descriptions, and a contextual comment on how the variable is used. Automated ways of updating the dictionary should be linked to program generation. A well-structured format allowing validation tools to parse the dictionary is required.

Real-Time Considerations

Validation normally adopts black box testing techniques. However, some validation tests require insight to the computing systems. The trend to segregate the redundancy management and mode logic techniques from the control law application software is continuing. System partitioning of this nature was proven valid with the X-29 FCS. The X-29 FCS used separate processors for input/output and control law execution. The F-18 HARV project will also demonstrate the validity and benefits of this type of embedded system partitioning with the RFCS. Redundancy management and mode logic functions account for a large percentage of validation test cases. These functions rarely change during the course of flight testing. Test cases for the redundancy management of embedded systems are typically generated using insight to the implementation of the redundancy management. The test case requirements are strongly influenced by knowledge of the internal software logic. To automatically test the input/output logic of an embedded system, monitoring and independently controlling all of the input/output signals in real time is vital. Control laws are traditionally tested with the black box approach. Time history and frequency response test case requirements are influenced more by the aircraft's envelope and operation than by internal software logic. The automatic testing of control laws is easier to achieve.

Real-time unobtrusive access to internal variables was needed several times during the X-29 validation and flight test process. In several cases, internal intermediate variables needed to be examined. For instance, the accuracy of an onboard analytical actuator monitor had to be verified during closed-loop dynamic maneuvers. This verification required software modifications to instrument the code. During flight test, surface command reasonability checks were tripping during the take-off roll, causing a down-mode to the analog backup control system. Analysis of how close the monitor was to tripping was needed as the aircraft taxied. The use of a temporary storage variable for multiple intermediate calculations should be avoided to facilitate real-time external monitoring. Tradeoffs in memory and timing constraints may be less critical with the advent of higher density memories and faster processors.

In most cases, modifications to the X-29 real-time software were made to output the required variables on an ARINC 429 bus. Sixty-four 16-bit words could be output. Four modules, each executing at 40 Hz, were used to load FCC output buffer registers for use by downlink instrumentation. In the X-29 validation lab, the ARINC 429 bus signals were relayed to the real-time simulation computer through a 1553 bus for display on strip charts. The primary use of the ARINC bus was to downlink vital signs of the FCS for in-flight monitoring. Methods allowing different 429 bus variables to be selected without requiring software modifications were considered, but never implemented.

As described in Ref. 10, high-performance experimental aircraft programs have traditionally relied on parameter estimation techniques to determine aircraft stability for safety of flight envelope expansion. During the X-29 envelope expansion, intermediate control system variables on the ARINC 429 bus were downlinked to perform near-real-time longitudinal frequency response measurements to assess vehicle stability characteristics. This capability required control system code instrumentation to capture the correct values for monitoring. Well-positioned, selectable software monitoring points would have avoided the need for control system changes and would have allowed for other uses of this type. The F-18 HARV FCS has a programmable feature by which 64 different variables can be requested from the basic FCS by the RFCS for output on the aircraft's 1553 avionics bus. However, to change the set of variables a recompilation of the RFCS software is necessary. Recompilation is not desirable for software under test.

Lengthy post-test data dumps are difficult to analyze. After a test is run, events may not be accurately remembered. Real-time data analysis offers more flexibility and would increase productivity in tracing real-time execution when applying troubleshooting techniques. Vital signs of the soft-

ware should be monitored in real time. Dynamically choosing real-time monitoring points and transporting the data to an engineering workstation is desirable.

Flight Control and Test System Synchronization

The aircraft simulation and flight control computers had to be synchronized for the X-29 hardware-in-the-loop configuration. Automated open-loop frequency response tests showed erroneous phase margins at higher frequencies because of unpredictable timing relationships between the simulation and FCCs. Fortunately, an 80-Hz synchronization output discrete was generated by the FCS and was used to drive the simulation real-time executive. This allowed for more predictable timing relationships and helped to correct the problem.

Timing relationships between the test system and embedded system can be critical. An automated test system must have the ability to introduce an error function at any point in the real-time cycle of the FCS software. During hardware-in-the-loop testing of the X-29, a high-frequency pulse to a canard position feedback was discovered to cause loss of control. This failure scenario only occurred 50 percent of the time. Loss of control was dependant on when the failed surface position input was sampled by the flight control system. Complex software modifications were required to use spare discrete and analog input signals for fault introduction. To completely automate failure modes and effect testing, these situations can be avoided if the embedded system and target system are interfaced correctly.

Embedded schemes, allowing spare input and output signals to be used by the automated test system, are desirable. For example, techniques are being developed to reserve externally controlled input and output discretes to automatically force miscomparisons of bit-for-bit voting planes. Other validation requirements are concerned with timing analysis. Appropriate hardware or software interfaces to allow analysis of standard case and worst case execution times are typically exercised during verification. System synchronization issues require visibility into the internal operation of the embedded system. The X-29 SEU could generate output discretes based on CPU program counter information to drive digital timers. To automatically address timing issues, appropriate hardware and software interfaces such as this are needed.

Software Instrumentation

The real-time characteristics of the onboard software inhibit reliably applying automatic code instrumentation after the software has been constructed. Automatic code instrumentors would need to know all subtle timing relationships. Code instrumentation should be done in conjunction with system design to assure these timing relationships are not disturbed. As a flight release goes through several changes in the operational phase of a program, software test points which have initially been added to the code may no longer be in the correct place to automatically test a change. To eliminate or reduce these adverse effects software test points should be included during the software development.

Concluding Remarks

Automated closed-loop validation of man-rated flight control systems is being done at the NASA Ames Research Center Dryden Flight Research Facility. Operational experiences in developing and using these automated testing techniques have highlighted the need for incorporating target system features to improve testability. Improved target system testability can be accomplished with the addition of nonreal-time and real-time features. Online access to target system implementation details, unobtrusive real-time access to internal user-selectable variables, and proper software instrumentation are all desirable features of the target system. Also, test system and target system design issues must be addressed during the early stages of the target system development. Processing speeds of up to 20 million instructions/sec and the development of high-bandwidth reflective memory systems have improved the ability to integrate the target system and test system for the application of automated testing techniques. New methods of designing testability into the target systems are required.

References

[1] Goossen, Emray, "Verification of Flight Critical Systems," Society of Automotive Engineers, Inc., Apr. 1990.

[2] Duke, Eugene L, "V&V of Flight and Mission-Critical Software," IEEE, May 1989.

[3] Szalai, Kenneth J., Felleman, P.G., Gera, J., and Glover, R.D., *Design and Test Experience With A Triply Redundant Digital Fly-By-Wire Control System*, AGARDograph, July 1976.

[4] Petersen, Kevin L., "Flight Control Systems Development of Highly Maneuverable Aircraft Technology (HiMAT) Vehicle," AIAA 79-1789, Aug. 1979.

[5] Mackall, Dale, McBride, David, and Cohen, Dorothea, *Overview of the NASA Ames-Dryden Integrated Test Facility*, NASA TM-101720,1988.

[6] Chacon, Vince, and McBride, David, *Operational Viewpoint of the X-29 Digital Flight Control System*, NASA TM-100434, 1988.

[7] Earls, Michael R., and Sitz, Joel R., *Initial Flight Qualification and Operational Maintenance of X-29 Flight Software*, NASA TM-101703, 1989.

[8] Office of Safety, Reliability, Maintainability and Quality Assurance NASA Quality and Productivity Improvement Programs, Quality and Productivity Awareness vol. 2 no. 3 "New Test Capability Developed at NASA Ames-Dryden," July 1989.

[9] Glover, R.D., *Design and Initial Application of the Extended Aircraft Interrogation and Display System. Multiprocessing Ground Support Equipment for Digital Flight Systems*, NASA TM-86740, 1987.

[10] Bosworth, John T., *Flight-Determined Longitudinal Stability Characteristics of the X-29 Airplane Using Frequency Response Techniques*, NASA TM-4122, 1989.

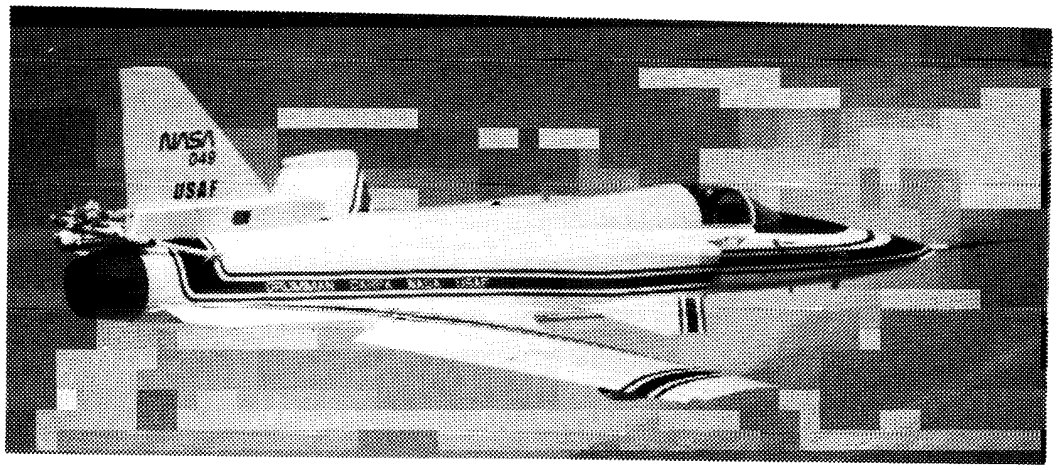


Fig. 1 The X-29 forward-swept-wing aircraft.

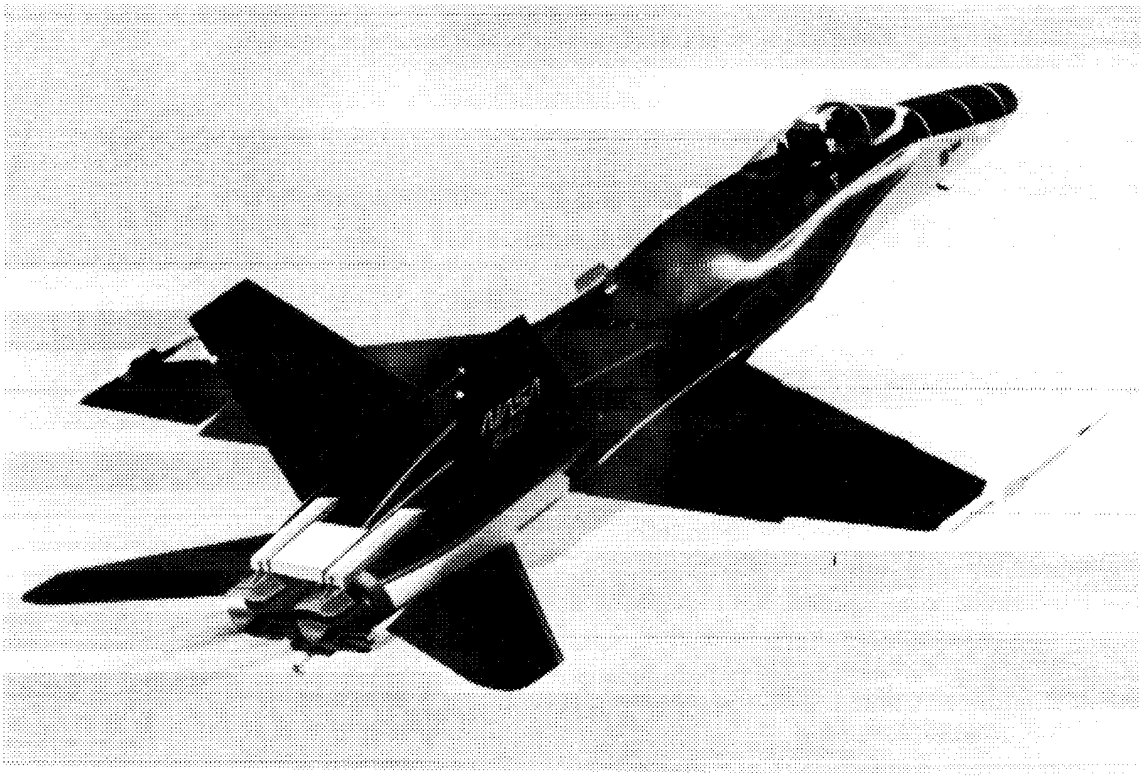


Fig. 2 The F-18 HARV aircraft.

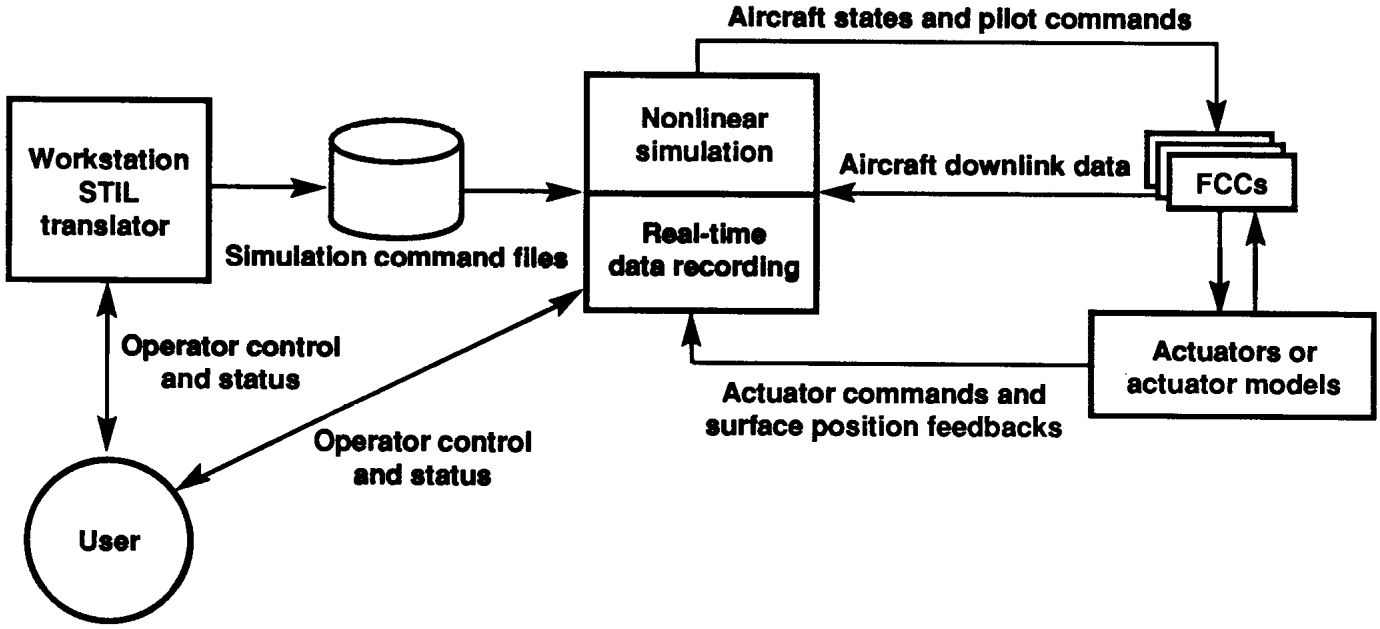


Fig. 3 The X-29 automated testing system functional diagram.

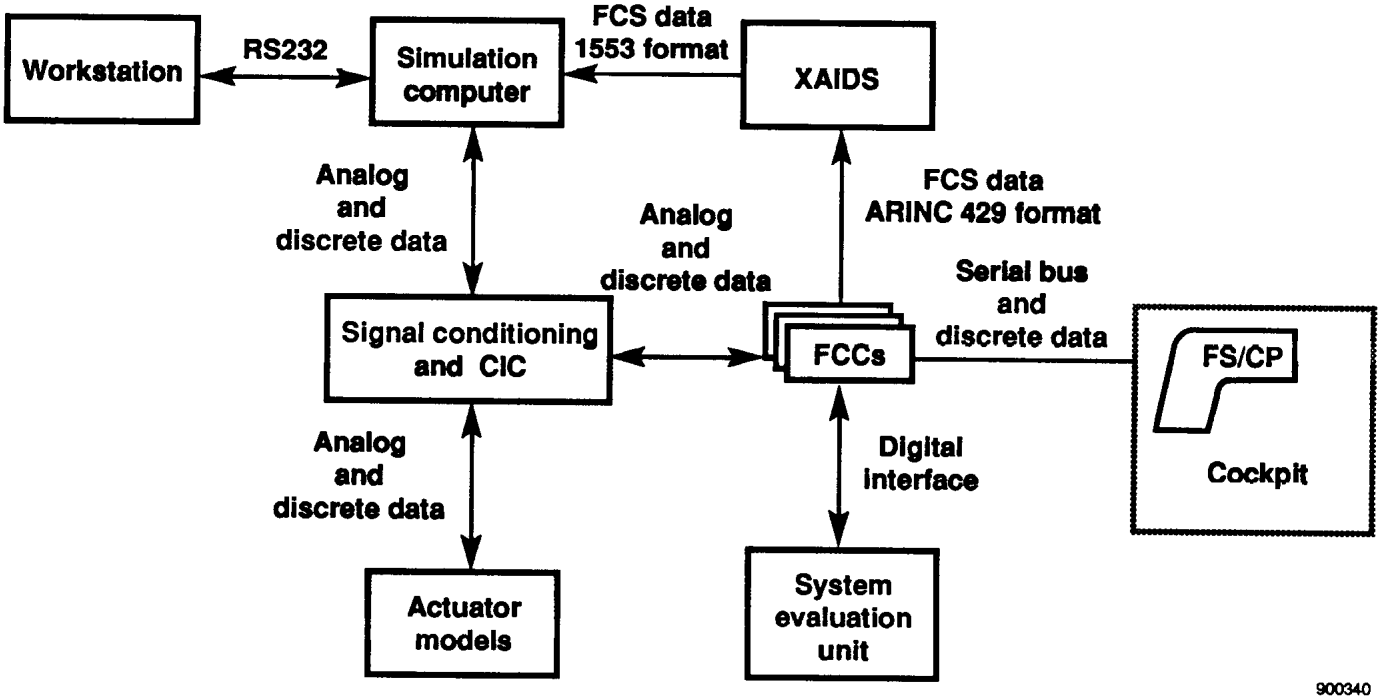


Fig. 4 The X-29 hardware-in-the-loop simulation.

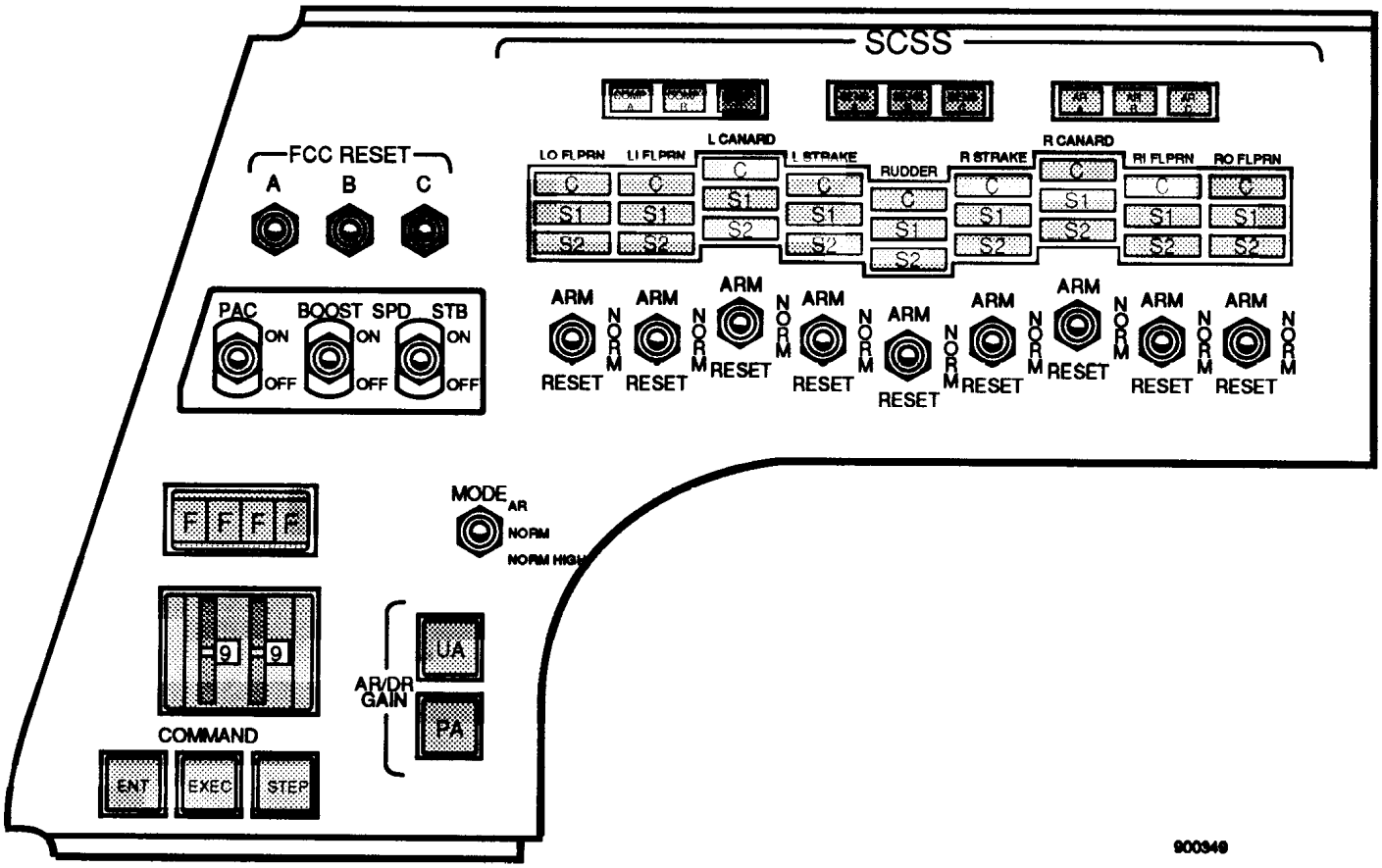


Fig. 5 The X-29 failure status/control panel.

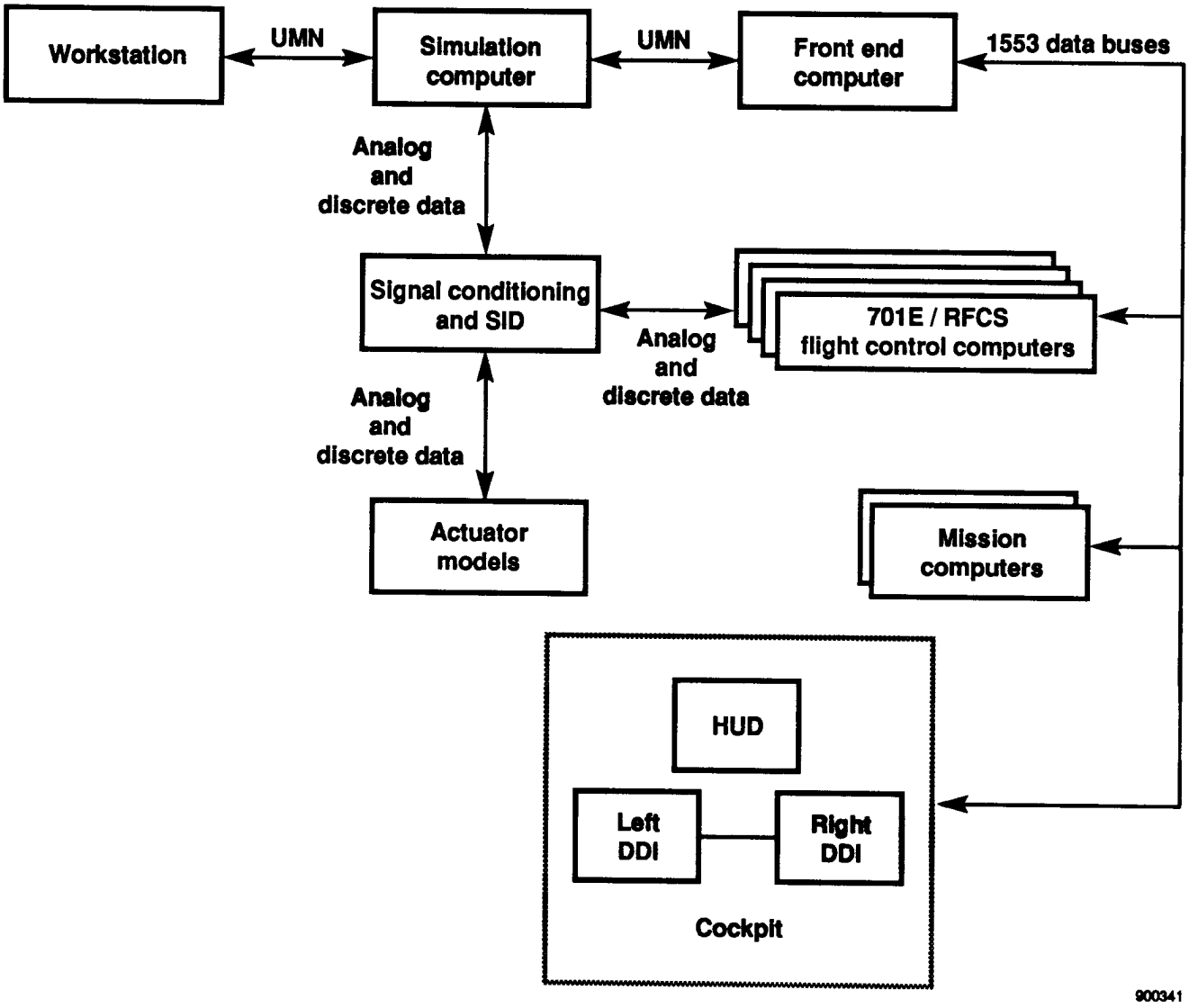


Fig. 6 Overview of the F-18 HARV hardware-in-the-loop simulation.

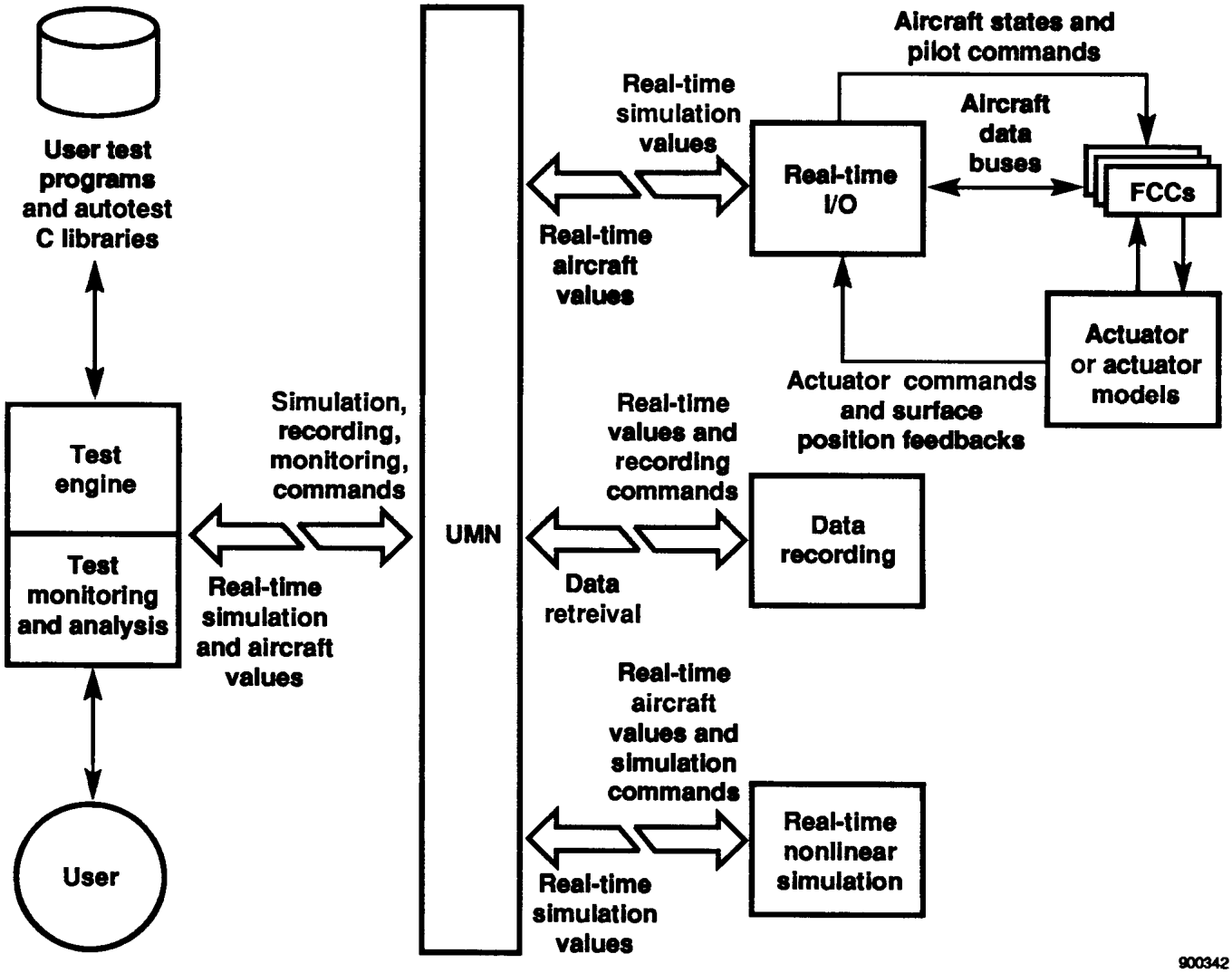
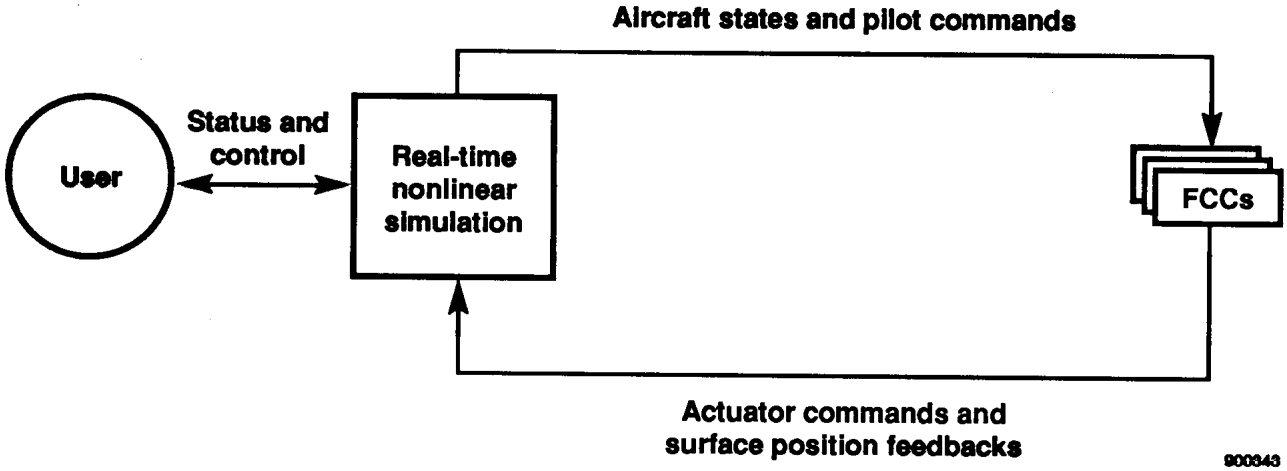
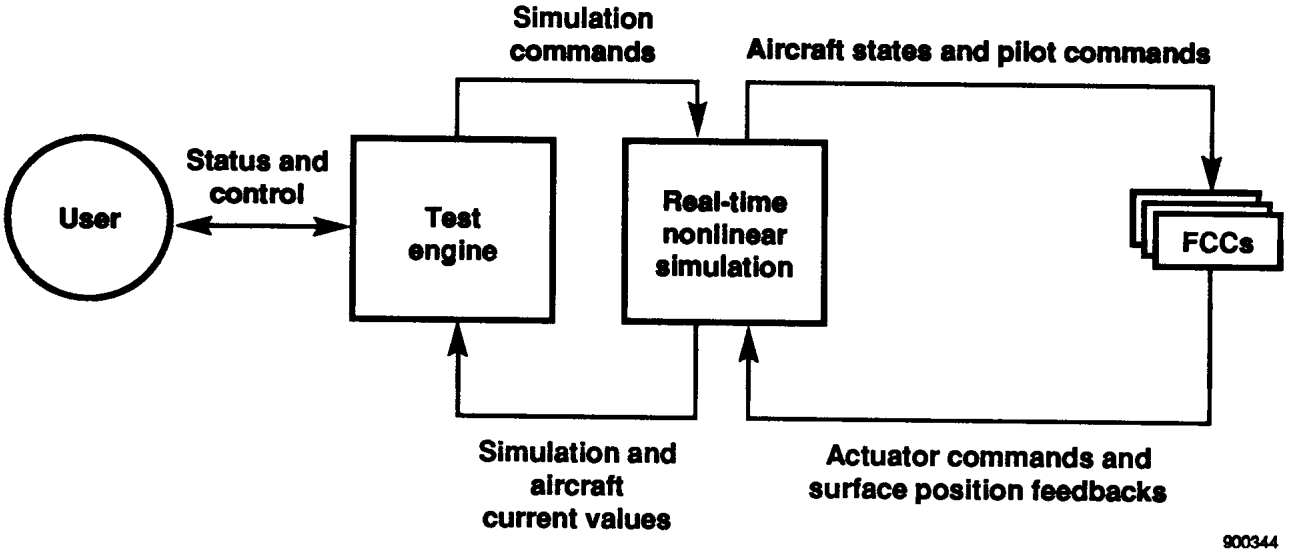


Fig. 7 The F-18 HARV automated test system functional diagram.



(a) Manually controlled simulation.



(b) Closed-loop automated simulation control.

Fig. 8 Simulation control comparison.

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM-4242		

4. Title and Subtitle
Flight Control System Design Factors for Applying Automated Testing Techniques

5. Report Date
October 1990

6. Performing Organization Code

7. Author(s)
Joel R. Sitz and Todd H. Vernon

8. Performing Organization Report No.
H-1631

10. Work Unit No.
533-02-51

9. Performing Organization Name and Address
NASA Ames Research Center
Dryden Flight Research Facility
P.O. Box 273, Edwards, California 93523-0273

11. Contract or Grant No.

13. Type of Report and Period Covered
Technical Memorandum

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Washington, DC 20546-3191

14. Sponsoring Agency Code

15. Supplementary Notes
Joel R. Sitz: Ames Research Center, Dryden Flight Research Facility, Edwards, California.
Todd H. Vernon: PRC Systems Service, Edwards, California.

Prepared for presentation at the IEEE 9th Annual Digital Avionics System Conference, Virginia Beach, Virginia, October 15–18, 1990

16. Abstract

Automated validation of flight-critical embedded systems is being done at the National Aeronautics and Space Administration Ames Research Center Dryden Flight Research Facility.

The automated testing techniques are being used to perform closed-loop validation of man-rated flight control systems. This paper discusses the principal design features and operational experiences of the X-29 forward-swept-wing aircraft and F-18 high alpha research vehicle (HARV) automated test systems. Operationally applying automated testing techniques has accentuated flight control system features that either help or hinder the application of these techniques.

The paper also discusses flight control system features which foster the use of automated testing techniques.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
X-29 aircraft F-18 aircraft Automated testing Embedded flight control systems	Unclassified-Unlimited Subject Category 05

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	16	A02

NASA FORM 1626 OCT 86 *For sale by the National Technical Information Service, Springfield, Virginia 22161-2171*

NASA-Langley, 1990