Testing of the YC-14 Flight Control System Software

D. L. Martin* and D. Gangsaas*
The Boeing Company, Seattle, Wash.

The YC-14 tactical transport aircraft uses three channels of digital flight control electronics (FCE) to provide fail-operational, fail-safe augmentation of the basic flight control characteristics. This paper discusses the system-level testing performed to verify the software prior to first flight and then prior to flights using modified versions of the software. Testing is performed in two phases: the FCE are connected to a simulation of the aircraft systems with no aircraft dynamics (open loop); and the FCE are connected to a simulation of the aircraft systems and the aircraft dynamics with simulated sensor inputs (closed loop). A semiautomatic testing facility used for data management and control of the tests is described.

Introduction

THE YC-14 is a tactical transport aircraft designed to demonstrate the feasibility of combining short-field takeoff and landing (STOL) performance with high-speed and high-altitude flight. The flight control system must provide good handling qualities with substantial margins of safety at all speeds. It must also compensate for the unusual flight characteristics typical of powered-lift STOL aircraft and allow conventional piloting techniques to be used at all flight conditions. The following functions met these objectives: command and stability augmentation in pitch, roll, and yaw axes; speed-hold and flight-path control; automatic aircraft configuration control; automatic engine-out control; altitude hold, heading hold, and refuel mode; in-flight selection of alternative control laws; fail-operational, fail-safe redundancy with automatic failure detection, annunciation, and redundancy management; and automatic preflight test. A redundant flight control electronics (FCE) configuration using three digital computers gave the prototype aircraft the necessary performance and design flexibility.

Initial development of the digital FCE involved separate testing of hardware and software. Special acceptance test software was furnished by the supplier to verify all hardware functions, such as data processing, data transmission, and analog signal conditioning. Intial verification was performed by the supplier while developing the first issue of the flight resident software. This testing was performed by processing modules of the code in a host computer, which simulated the FCE processor. As more modules were developed, this method was expanded to segment- and frame-level testing, in which groups of modules were processed together.

After hardware verification and initial software development were completed, the system-level testing was started. This involved testing the triplex flight computers loaded with the flight software and connected through associated interface units to a simulation of the aircraft sensors and control systems. Testing was performed prior to first flight and then prior to all flights with modified software.

Presented as Paper 77-1077 at the AIAA 1977 Guidance and Control Conference, Hollywood, Fla., Aug. 8-10, 1977; submitted Sept. 2, 1977; revision received March 15, 1978.

Index categories: Guidance and Control; Testing, Flight and Ground.

*Member of the C-14A Flight Controls Technology Staff. Member AIAA.

Electrical Flight Control System

Figure 1 describes the architecture of the YC-14 electrical flight control system. It is built around three general-purpose digital computers described in Table 1 and configured to provide fail-operational and fail-safe operation. Computer timing is divided into eight frames of 10 ms each. Computations are performed at the rates shown in Table 2.

Inputs to the computers are received via parallel digital transmission lines from interface units and via serial digital transmission lines directly from one inertial navigation system and three air data systems. Analog and discrete signals from aircraft-motion, control-feedback, pilot-control, and aircraft-configuration sensors are converted to the proper format in the interface units.

Outputs are transmitted as parallel digital data to the interface units, where the appropriate signal conversions are accomplished. Servo loops are closed in analog form through the interface unit. An exception is the throttle servo, in which position feedback is supplied by a digital encoder and loop closure is accomplished in software.

All cross-channel data required for redundant operation of the digital system are transmitted between the computer units via optical data links. A control and display panel (CDP) and a test and failure identification panel (TFIP) are the crew's interface with the system. The CDP has system-start, channel-engage/disengage, and flight-mode switches as well as status displays. The TFIP is used to run automated preflight tests of the system and displays the failures detected in flight.

The computers are programmed to accomplish aircraft control-law functions, system self-test, and rendundancy management. Alternative software can be selected by the pilot to aid control-law and system development. Figure 2 shows the memory allocation. Additional information on the YC-14 flight control system can be found in Refs. 1 and 2.

Software Testing

Software development was divided into three tasks: coding, testing, and control. Figure 3 illustrates the functional relationships between these tasks. Starting with the same software functional requirements, each task was performed by a separate team to provide redundancy in the interpretation of the software functional requirements. The coding team was responsible for establishing code requirements, supplying software documentation, and generating code. The testing team established testing requirements, predicted test results, and performed software testing. The control team approved the testing requirements, test results, and software documentation, and released new software issues. One software expert and two flight control

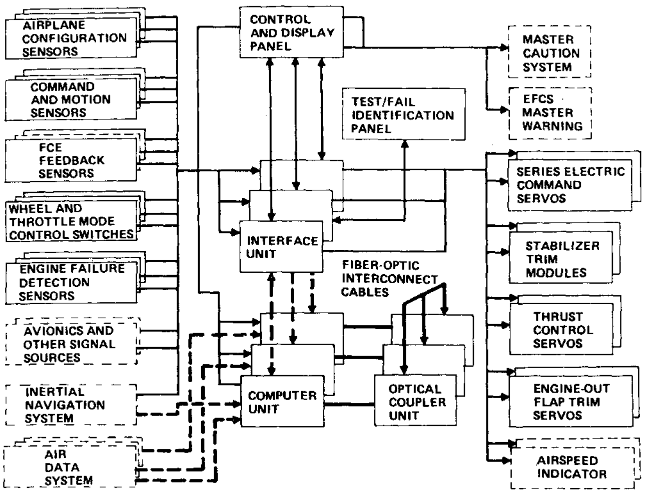

Fig. 1 YC-14 electrical flight control system schematic.

system designers with coding experience comprised the coding team. The testing team consisted of the three flight control engineers with detailed knowledge of the functional software specifications. The control team was composed of representatives from the Quality Assurance and Program Management organizations and included one member who was responsible for software configuration control.

The objective of the software testing was to verify the proper operation of all control-law and redundancy-management functions. Testing was accomplished in two phases, as shown in Fig. 4. Phase 1 consisted of software evaluation testing. Discrepancies between predicted and actual results were resolved by the coding and testing teams, followed by appropriate corrections to the code or the test prediction programs. After resolution of all discrepancies, the final software was submitted for formal testing. Phase 2 was closely monitored by the software control team, who approved the software for flight use following successful completion of this phase. Two methods of system-level testing were used, which were designated as open loop and closed loop.

Test Facility

Figure 5 is a schematic of the major components of the laboratory facility interfaced with the flight hardware. Cables

Table 1 Features of flight computers

Program memory	13,824 16-bit words
Constant store memory	1024 16-bit words
Scratch pad memory	1024 16-bit words
	1024 1-bit words
Clock rate	1.4 MHz
Frame synchronization interval	80 ms
Number of inputs per channel	
Continuous	43
Discrete	100
Number of outputs per channel	
Continuous	38
Discrete	88

Table 2 Computational rates

	Iterations per second
Control laws	12.5, 25, 50
Redundancy management	12.5, 25
Executive	12.5, 100

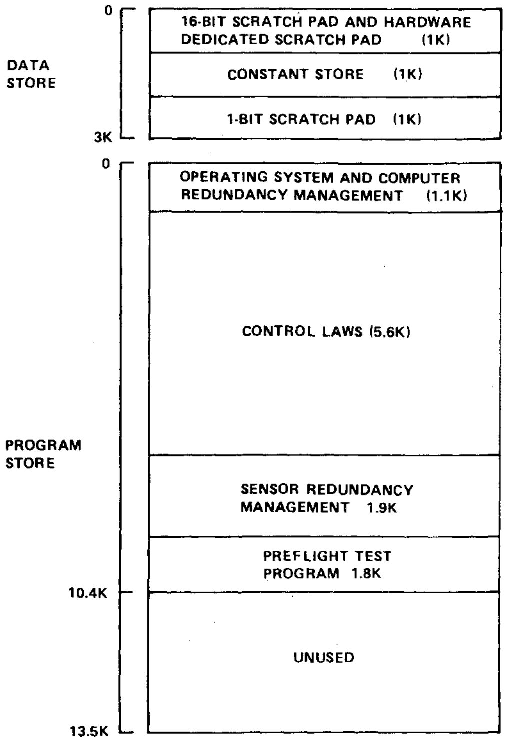

Fig. 2 YC-14 flight control computer software memory allocation.

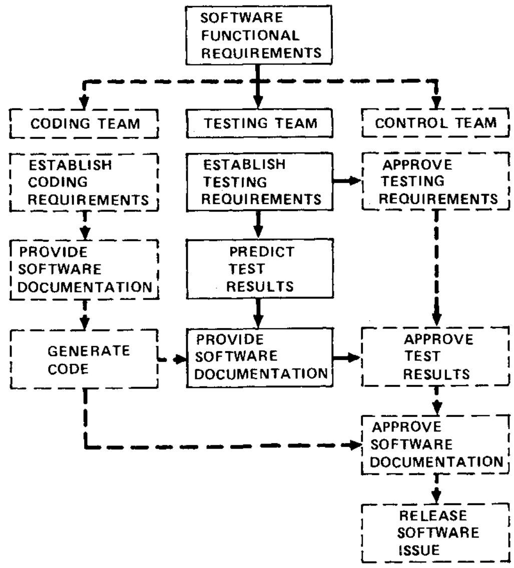

Fig. 3 Software development.

and patch panels necessary for carrying out both open-loop and closed-loop testing are included.

The flight hardware consists of three computer units, three interface units, a control and display panel, a test and failure identification panel, and three optical coupler units. Interconnection is by means of junction units associated with the patch panels. Either the flight EPROM memories or laboratory core memories can be used for testing. The latter can be programmed directly from a teletype.

The digital interface computing equipment (DICE) allows semiautomatic testing. It consists of a general-purpose digital computer combined with many input and output devices. It is connected to a teletype, a paper-tape reader, a line printer, and a magnetic tape unit. Programming is done in assembly language. In open-loop testing, the DICE read test data files from magnetic tape, printed-out operator instructions and test descriptions, applied inputs and sampled outputs, and

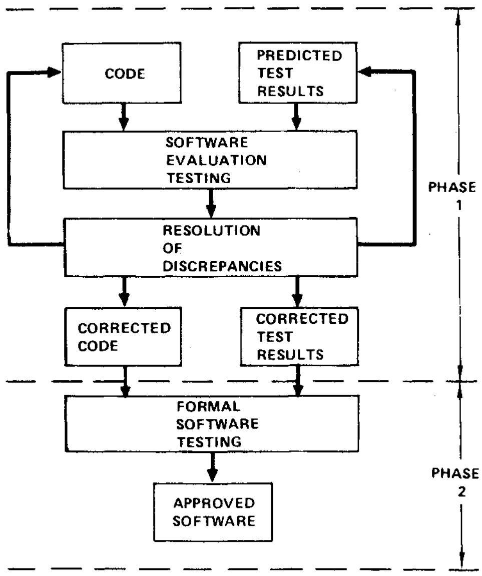

Fig. 4 Software testing procedure.

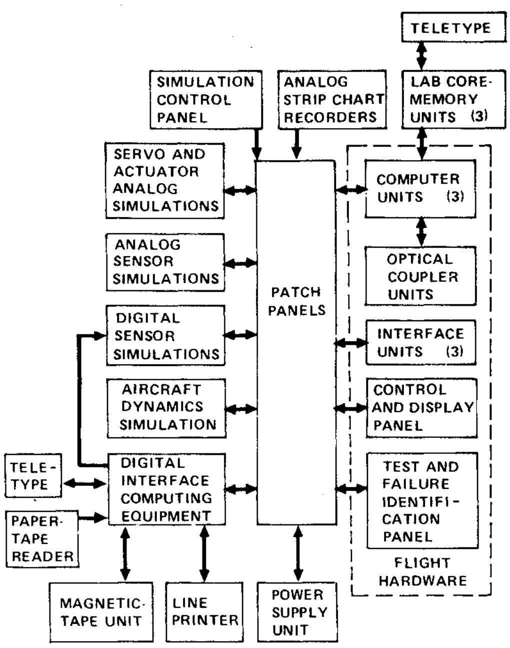

Fig. 5 YC-14 FCE software laboratory configuration.

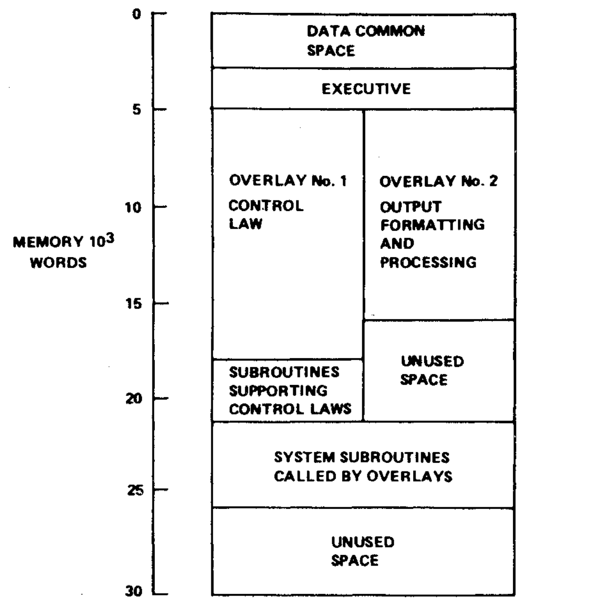

Fig. 6 Open-loop test simulation program software memory allocation.

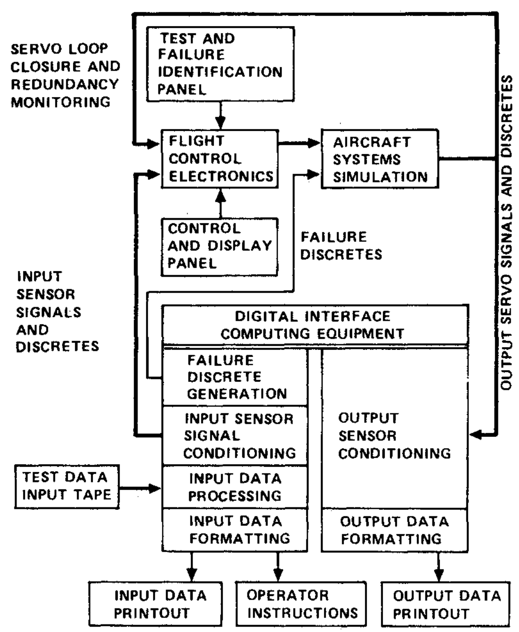

Fig. 7 Open-loop test configuration.

printed out actual and predicted test results. In closed-loop testing, it made all analog-to-digital data conversions.

Two SD-80 analog computers are used to simulate electrical commands, mechanical controls, control-surface actuation, and engine controls. In addition, switching functions allow isolation of servos following simulated failures. One EAI-581 analog computer is used for a six-degree-of-freedom simulation of the airplane dynamics.

Electronic buffer circuits are used for right scaling and format of all analog and two-state discrete inputs to the computer interface units. All digital sensors are simulated by the DICE, which transmits data to the flight hardware in the proper format and at the correct rate.

Simulation control panels and a teletype form the interface between the operator and test systems. Pilot action can be simulated from one control panel, which involves wheel, column, rudder pedals, and throttle lever inputs, as well as aircraft configuration selection. Others have manually operated switches that allow central control of the analog and digital computers and the simulation of failures. Test configuration changes are made by replacing prewired interconnect patch panels and by reprogramming the computers.

Test Result Predictions

A digital computer simulation was developed to predict open-loop test results. It was implemented on a general-purpose minicomputer. The memory allocation is shown in Fig. 6. The program performs open-loop test simulation and output formatting and processing.

Starting with the software functional requirements, the control laws were programmed in FORTRAN. This effort was independent of the flight software coding, except that the same computational sequences, data transfer delays, and iteration rates were maintained. Redundancy management functions and software provisions for system self-test were not included. Likewise, the effects of truncation and granularity associated with the fixed-point arithmetic used in flight computers were not represented. The modeled control-

Table 3 Test printout

Test Description	Input definitions	Predicted and actual outputs
Software issue number Date of test result predictions Test identification number Description of test purpose	Intial values for 21 sensor inputs Type and magnitude of dynamic input Output sample time State of six input discretes Aircraft system configuration Control mode selections Failure conditions	21 servo positions 20 command discretes 11 aircraft system configuration discretes

law software was combined with programs representing the characteristics of the flight computer interfaces and the associated elements of laboratory hardware and software. All essential open-loop test functions, as well as input and output routines, were included. All test simulation software was validated by inspection.

Approximately 800 test definitions and predicted test results were transferred by means of paper tape to the laboratory and stored on magnetic tape. A tabular listing of the data for each test was generated for reference during open-loop testing.

Closed-loop test results were obtained from the YC-14 flight simulator. The data consisted of time histories describing aircraft motions and control-surface positions following control inputs at seven flight conditions. Concurrently, linear models of the aircraft dynamics were supplied.

Redundancy management test results were obtained by inspecting the applicable functional requirements. In the case of open-loop testing, test data files were generated by manual input from the teletype. This allowed semiautomatic testing of these functions.

Open-Loop Testing

Figure 7 is a schematic of the laboratory configuration used for open-loop testing. This method was designed to allow software testing by systematic observations of input-to-output relationships. The FCE were connected to a simulation of associated aircraft systems. This simulation represented input and output functions in sufficient detail to permit verification of all control paths. Since the responses of the FCE to defined sets of inputs were being evaluated, the aircraft dynamics and the associated feedback signals were not simulated. This accounts for the term "open-loop testing." Feedbacks were provided for servo-loop closures and redundancy management monitoring.

Semiautomatic testing capability was furnished by a general-purpose digital computer, which was supported by many input and output devices, a magnetic tape unit, a teletype, and a line printer. Upon operator instruction, the computer obtained information stored on magnetic tape, provided inputs to the FCE, and received the outputs generated through the aircraft systems simulation.

The procedure allowed evaluation of both static and dynamic input-to-output relationships in a given test by applying inputs in two stages. First, an initial state of all inputs was established. After output transients subsided, one input was excited in the form of a step, ramp, or pulse. At a time ranging from 0.4 to 16 s later, the outputs were sampled. After each test, the information listed in Table 3 was printed. Referring to a set of previously defined tolerances, a particular test would be passed or failed after a visual inspection of predicted and actual test results.

The predicted test results were obtained from the test simulation, as shown in Fig. 8. It modeled the functions of the flight control software and all of the hardware involved in the

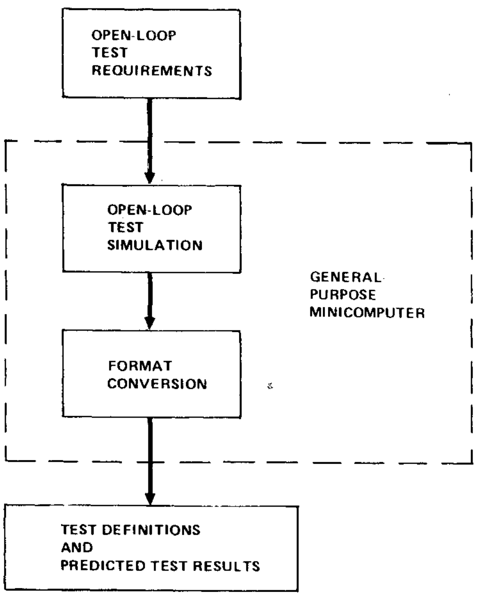

Fig. 8 Test result prediction for open-loop testing.

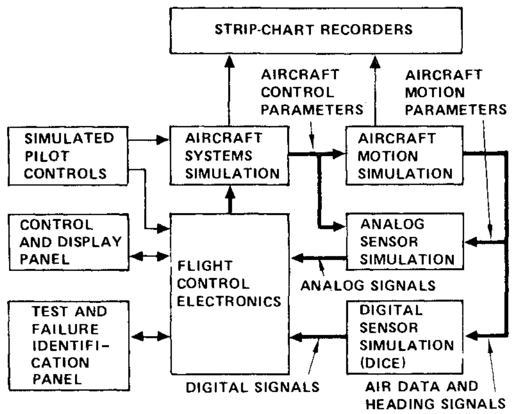

Fig. 9 Closed-loop test configuration.

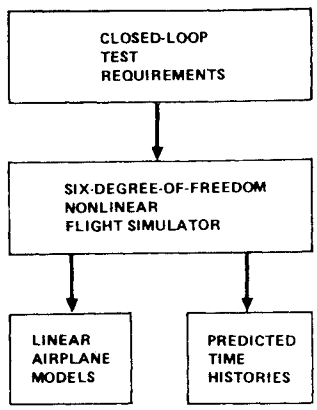

Fig. 10 Test result prediction for closed-loop testing.

laboratory testing. For the test conditions, inputs and predicted outputs were converted to a format compatible with the laboratory computer.

Some of the advantages offered by open-loop testing are 1) the total control of most inputs allowed methodical segment-by-segment testing of software, which was especially helpful in locating coding errors; 2) the relatively simple definition of input-to-output relationships in the results was suitable for formal software acceptance testing; 3) because of the lack of compensating airplane motion feedbacks, computational drift was detected more readily; and 4) compared to closed-loop testing, this method required less complex supporting laboratory equipment. Among the disadvantages encountered with open-loop testing are 1) some software functions could not be tested without the modeled airplane response; and 2) because of the lack of compensating airplane motion feedbacks, computational drift considered acceptable still would degrade accuracy.

Closed-Loop Testing

Figure 9 shows the laboratory test configuration used for closed-loop testing. The FCE drive the same aircraft system simulation used for the open-loop testing, but additional parts of the mechanical control system are included. These are not associated with the FCE but provide the mechanical paths for simulated pilot control. Inputs to the FCE are supplied by pilot-action, aircraft-system, and aircraft-motion simulations. Strip charts are used to record system operation.

The purpose of this method was to test the software in a closed-loop control capacity using the YC-14 flight simulator as a standard to measure its performance. The flight simulator was one of the main tools used to establish software functional requirements and represented the ultimate standard for comparisons. As shown in Fig. 10, linear models of the aircraft dynamics and small perturbation-time histories were obtained from the simulator.

Testing covered a wide range of aircraft configurations and seven flight conditions. Time histories of flight parameters and control inputs were obtained on strip charts and compared to those obtained from the flight simulator. Because of the inadequacy of a linear representation of nonlinear airframe dynamics, the evaluation of system performance was done in a qualitative, rather than quantitative, manner.

Advantages of closed-loop testing include locating major software errors relatively quickly and testing those software functions that required modeled airplane response. A disadvantage encountered with closed-loop testing is the complex simulations of aircraft dynamics and mechanical control systems. Also, unless a high degree of complexity is added to the laboratory equipment, only qualitative comparisons can be made to predicted test results derived from a fully nonlinear flight simulation. Thus, subtle but significant software errors may not be detected.

Test Experience

During the testing of the first issue of software, considerable effort was devoted to debugging the laboratory test facility. Although each component had been checked in great detail prior to the arrival of flight hardware, it was necessary to make the final verification of the test facility using the flight computers with operational software. Problems encountered during testing were largely associated with inaccuracies and failures of analog-to-digital interfaces. Additionally, limited digital computer capability constrained the development of automatic test procedures. As the program progressed, the testing efficiency increased significantly because of reduced facility maintenance and improvements in operations. This, plus fewer coding errors, accounts for the improving trend illustrated in Table 4.

The prototype nature of this program demanded frequent software updates as part of the flight control development. To minimize the impact on the aircraft flight testing, strict adherence to schedules was imposed on the software test program. A critical evaluation of testing requirements was needed to insure sufficient quality of the software within the constraints of time.

Table 4 Testing effort

Software issue	New or changed software modules, %	Man-months to complete testing
1	100	24
2	12	4
3	15	3
4	39	6

The initial control-law software verification was most time-consuming. The testing and coding teams worked together closely to identify and correct errors. Initially, all testing was performed with three channels but, during the flight test program, a single channel of flight hardware was used for the first phase of software validation. This required additional complexity in the laboratory to compensate for the two missing channels but had the advantage that only one set of hardware was removed from flight status. Subsequently, testing of redundancy management software and the formal software qualification testing were done using a full complement of three channels.

Errors in the control-law software, such as missing control paths, incorrect gain schedules, numerical overflows, and major errors in logic, were readily detected during the early parts of the verification. Computational anomalies causing drift of integrators and subtle control-law logic errors normally were discovered during a second round of testing.

The redundancy management scheme used for the YC-14 flight control system was unsophisticated. Three-channel median selection and two-channel averaging of sensor inputs after first failure were used. Servos were either triplex or dual with one model channel. The channel associated with a failed input sensor or servo would be isolated following first failures. After a second failure, the remaining two channels would be taken off line. Computations were identical in all channels. Each signal selector was tested for proper three- and two-channel operation, and signal monitors were tested for correct thresholds and time delays. Command failures, both open-feedback and command hardover, were simulated for each servo. Proper failure warnings, failure identification, and system reconfiguration were verified for first and second failures.

After four major software revisions, only one significant software error was detected during flight. The error, which caused mistracking of the control-law computation in the three channels, was the result of incorrect use of cross-channel data for one parameter. Each synchro input was multiplied in software by a factor equal to the ratio of the nominal reference voltage to the actual reference voltage. Both the synchro outputs and the reference voltages were transmitted between channels, and the three inputs would be compensated in each channel prior to signal selection. However, because of an error in timing, each channel was using the current correction factor for its own sensor, whereas the correction factors for the other two sensors were from the previous frame. Thus, each channel performed signal selections on a different set of values, resulting in different selected input data for the three channels. Although the discrepancies were small, the effect of threshold detectors and integrators led to large mistracking between channels during flight. In the laboratory, the variations in the simulated synchro reference voltages were sufficiently small that this error would not be detected unless a bit-by-bit comparison between channels had been made.

Although control laws normally were verified on the flight simulator, the frequent software updates required some verification of their functions during the software testing. Redundancy management functions were not represented adequately on the flight simulator, and therefore, three-channel software testing represented the first opportunity to functionally verify redundancy management concepts. A deficiency in the requirements for servo monitoring was not detected until the failure modes were verified in flight. In the laboratory, servo hardovers were simulated by an input to the servo amplifier, whereas in flight a hardover was generated by driving the servo feedback. In the first case, the hardover would be detected, but in the latter it would not.

A few minor errors were found by inspection of the code after the software had been approved. These did not affect the function of the software sufficiently to be evident during software or flight testing.

Conclusions

The system-level approach to software testing used in the YC-14 program has been successful. Relatively error-free software has been produced while adhering to schedules imposed by the overall flight test program. This is largely attributed to organizing the software development into the three independent tasks of coding, testing, and control and the adoption of methodical semiautomatic test procedures.

The two methods of open-loop and closed-loop testing were complementary. However, because of reduced complexity, higher accuracy, and greater suitability for formal qualification testing, the open-loop method was preferred. The closed-loop method allowed quick scanning of the software for major errors, but subtle discrepancies were difficult to detect.

Both the speed and quality of the open-loop testing using the existing test facilities would be improved by measuring the input-to-output relationships based on digital data at the computer interface units, passing/failing individual tests automatically, and adding redundancy and preflight test functions in the simulation used to predict test results. Increased digital computing capability, including fast-access memory, would improve automation and speed, make it possible to generate input and output time histories rather than a single data point, and enhance aircraft systems simulation. Integration of flight computers with the flight simulator or development of a closed-loop test prediction simulation, including the linear aircraft model used in the laboratory, would increase the usefulness of closed-loop testing.

References

[1]Lee, A. H., "YC-14 Flight Control," AIAA Paper 75-1027, 1975.

[2]Kestek, R.E., "YC-14 Digital Flight Control Data Management," AIAA Paper 75-1087, 1975.