Study of UAV Flight Control Software
Test Method

Ting Sun, Jiasi Wang, Ying Zhang and Liang Xiang

ABSTRACT

UAV flight control software is the core of UAV flight control system. Sufficient test of the software is necessary to ensure safety and reliability of flight control system. According to the characteristics of some type of UAV flight control software, this paper proposed a test model based on multiple test environments and types, analyzed and applied test technique of coverage and fault injection, and developed a set of test techniques applicable to UAV flight control software.

INTRODUCTION

With the rapid development of UAVs in military applications, the safety and reliability of flight control software[1], as the core part of UAV, has become increasingly important for UAV flight control system. Once there is any problem with the software, it would affect normal control and operation of UAV system and result in serious harm to operational missions including reconnaissance and intelligence transmission. Thus, it is necessary to conduct strict test of flight control software and improve software safety and reliability in order to guarantee UAV quality.

CHARACTERISTICS OF UAV FLIGHT CONTROL SOFTWARE TEST

Some type of UAV flight control system is a flight control and navigation system specially designed for fixed wing UAVs. The system integrates flight control

Ting Sun, Jiasi Wang, Ying Zhang, Liang Xiang, Beijing Special Vehicles Institute, Beijing, China
computer and micro integrated navigation system (BD/INS) and is capable of realizing autonomous cruise in various forms according to the scheduled route. Its flight control software directly controls chips based on ARM kernel single-chip microcomputer. It is able to control ailerons, ascent and descent, direction, accelerator and task load of UAV and fulfill functions including manual/remote control, autonomous navigation, rudder calibration and reconnaissance. Also, it is provided with complete flight status monitoring and alarm functions and perfect emergency response system so as to ensure a safe operation of the system. The software is a typical real-time embedded software. The flight control software and its test have the following characteristics:

1) Real-time performance: With a large number of time-related algorithms, the software is required to complete all data processing in certain time. Thus, whether program execution duration satisfies the time requirements specified in system specifications shall be verified. During the period of test and design, detailed information of data input and output time must be included in test cases and random process shall be used to define time-related input variables.

2) High requirements for safety and reliability: Any error in flight control software design may directly cause failure of operational missions or even damage to UAV. Thus, there are high requirements for software safety and reliability.

3) Reliance on hardware: Normally, embedded software relies much on hardware. Once there is any fault with the system, it seems difficult to determine if it is caused by hardware or software. Additionally, as flight control software is developed on the host, it is quite hard to test software error caused by hardware difference between the host and the target environment where software runs. Thus, the real operating environment of the simulated flight control software must be taken into account during the construction of test environment.

TEST MODELS BASED ON MULTIPLE TEST ENVIRONMENTS AND TYPES

According to the characteristics of the above UAV flight control system, it is necessary to plan software test in terms of document review, code review, static analysis, configuration item test, system test, etc., formulate a reasonable test strategy, choose a proper test method and build a perfect test environment so as to guarantee the sufficiency of test[2].

This paper proposed a type of test model based on multiple test environments and types. Three types of environments including lab environment, semi physical simulation test environment and declared test environment were built. Static test was conducted under lab environment; dynamic test of configuration items under semi physical simulation test environment; and UAV flight control system test under declared test environment. A combination of static and dynamic test was adopted. Document review, code review and static analysis were adopted in static test. Black-box test was applied in dynamic test. Test items were classified by function
decomposition according to software requirement specifications. Functional test, performance test, interface test, safety test and other test types were analyzed. Static test was followed by dynamic test, and configuration test by system test. Dynamic test was conducted after all problems detected in document review, static analysis and code review had returned to zero.

The test model has the following characteristics:

(1) Wide coverage. Test may be divided into configuration item test and system test. Different test strategies and methods are adopted to test documents, source codes and software programs so as to eliminate software defects. Software requirements, correct design documents, standard codes and correct design or logic are guaranteed by document review, dynamic analysis and code review. Requirements for software functions, performance, interfaces and safety are checked in configuration item test. Software’s reliability in flight control system and the ability to collaborate with other UAV-carried integrated software are checked in system test.

(2) Clear hierarchy. During the phase of test demand analysis, document review is conducted first so as to ensure that the functional requirement in software requirement specifications can cover overall requirements for coverage development. Then, test items are formulated by function decomposition according to development assignment book and software requirement specifications. Width is prioritized in decomposition of test items. During the test design stage, design is conducted according to the priority of test items and test items with higher priority are tested in a focused manner. Regression test may adopt test cases of the previous version and new version is subject to a regression test. The results of the two tests are analyzed and compared to prove that there is no new defect caused by change. When there is any change to the function, test is conducted according to the test cases to be added or revised and the correctness of change is verified.

**UAV FLIGHT CONTROL SOFTWARE TEST STRATEGY**

**Static Test Strategy**

Objects of static test include documents and source codes of UAV flight control software. Document review, code review and static analysis are mainly adopted. Document review sheet and code review sheet are used to check the document standardization, consistency between code and design, code implementation standard, correctness of code logic expression, rationality of code structure and readability of codes. The Test bed static analysis tool is used to make control flow analysis, data flow analysis, interface analysis and expression analysis of software. Encoding rules are examined to obtain scale, cyclomatic complexity and annotation rate. Check if the software satisfies requirements specified in GJB/Z 102–1997 Guidelines on Software Reliability and Safety Design. Prior to the start of dynamic
software test, the error or consistency of documents and codes shall be effectively eliminated.

Static test is conducted under lab environment. Testers review codes and documents using test computer and the test tools installed on the computer.

**Configuration Item Dynamic Test Strategy**

Although document problems, code and design defects of some software can be effectively detected in static test, it is impossible to detect all structural and functional defects of some software. Thus, configuration item dynamic test shall be conducted so as to further test the structural and functional defects of software. Black-box test and function decomposition method was used in dynamic test. The methods adopted include equivalence partitioning, boundary value analysis, and test data generation and verification.

Functional test mainly focuses on autonomous flight, semi-autonomous flight, manual flight, task control and rudder calibration, etc. Test was conducted on normal and abnormal conditions of functions. Normal or abnormal equivalence data were used as test input, input and output coverage boundary values.

Performance test focuses on validity of threshold value, reconnaissance radius, target positioning precision, memory margin and other performance indicators in order to make flight control software protection control effective.

Interface test: Flight control telemetry data interface, flight control telemetry data interface, flight control load control instructions interface and flight control load status data interface are Serial Port 422, and flight control navigation data interface and flight control navigation query instruction interface are Serial Port TTL. The serial port commissioning tool is used to send normal and abnormal data, receive software feedback data, verify the consistency between data content and format and interface protocol as well as abnormal data processing capacity. External interfaces of the system include those with baroceptor, Beidou satellite, task load, A/D converter. Verification is conducted in system test.

Safety test: Safety test is used to find out if flight control software is able to protect flight safety of UAVs in case any abnormality occurs to software test during the process of UAV flight control. Also, the hidden demand, especially factors influencing software reliability, was analyzed according to the characteristics of UAV flight control software. Flight control software protection control measures, protection and verification towards key data and acting mechanism of watchdog were investigated.

Dynamic test of configuration items is conducted under semi physical simulation test environment, as shown in Figure 1.

The simulation computer simulates and sends combined navigation data. Installed with a serial port commissioning assistant, the test computer sends frame data and operational instructions to flight control software according to load status data protocol, receives load control instructions, load status data and inertia
navigation data from flight control software and checks the correctness of data format and content.

![Diagram of Semi Physical Simulation Test Environment](image)

Figure 1. Semi Physical Simulation Test Environment.

Oscilloscope is used to observe waveform pulse width of output signals of rudder, elevator, aileron driver and parachute opener interfaces and verify the correctness of function realization.

**System Test Strategy**

System test was conducted based on system specifications. Following configuration item test, flight control software was subject to UAV system test. System test aims to check if the software satisfied various requirements specified in system specifications and verify the comparability and consistency of related software operation, adaptability between hardware and software and capacity of software to jointly complete system tasks.

System test was conducted under declared environment. UAV sends instructions first. Then, ground measurement and control station, remote control handle and ground data chain, etc. are used to control and execute system tasks. Completion of procedural system tasks is regarded as the end. The status of devices for software, system execution action and HMI display data were used to evaluate the correctness of system characteristic status completed by the software and to check if software functions, performance and interfaces satisfied requirements during system task execution.
TEST TECHNIQUE BASED ON COVERAGE AND FAULT INJECTION

Coverage Technique

In software test, coverage technique is employed to evaluate the completeness of test. In the actual operation, coverage is normally divided into test coverage based on demands and based on codes.

1) Test coverage based on demands

The test coverage based on demands may be evaluated several times during the test process. Test coverage rate marks may be provided at each milestone of the whole life cycle (such as planned, implemented, executed and successful test coverage). During the process of test execution, two methods are normally applied in test. One is to determine the test coverage acquired through test execution; and the other is to identify successful test coverage (or test without failure in execution, for example, test in which no software problem is detected).

2) Test coverage based on codes

Test coverage based on codes refers to the quantity of codes that have been executed in test, as opposite to the quantity of remaining codes to be executed. Code coverage may be built based on control flow (statements, branches or paths) or data flow. Control flow coverage aims to test paths of code lines, branch conditions and codes or other elements of software control flow. And data flow coverage aims to test whether data status is valid according to software operation test, for example, whether data elements have been defined before use.

Fault Injection Technique

Fault injection technique represents an important verification means of software test. It can effectively test the fault-tolerant ability of software. However, in the actual test, due to excessive fault types and approaches causing fault, it is impossible to inject all possible faults. However, fault modes are limited and typical fault modes include task halt, task suspension, delay output and invalid output. Common fault injection methods are described as follows:

1) Code fault injection. Some fault modes are selected and injected into system software. Then, software is evaluated according to the status of software after fault injection so as to achieve the goal of testing. For example, directly modifying source codes of software to make software abnormal, checking the status of the tested software, etc.

2) Data fault injection. Error data is injected, including abnormal data format, content, length, etc. A lot of software faults such as CPU, memory and network faults may be simulated by change in memory elements.

3) Operation fault injection. Error operation instructions are injected and the rationality of software design is judged according to the status of software processing on error instructions, such as error operation, repeated operation or direct modification of operation instructions.
In the dynamic test of configuration items of some type of UAV flight control software, in addition to the normal functional test, reliability test shall also be conducted so as to prevent system collapse due to poor fault-tolerant mechanism and boundary and abnormal data processing. Thus, fault injection may be used to effectively verify the capacity of software to process abnormalities.

The following types of fault injection methods are adopted in configuration item test of some type of UAV flight control software.

1) Power-on self-check test: Inject flash faults. Check if UAV flight control software is able to test internal hardware faults in initial energizing and give corresponding alarm prompts.

2) Pre-flight self-check test: Inject external load faults including inertial navigation and altimeter. Check if software can test and locate relevant faults in case of test before launching.

3) Protection control test: Inject threshold values specified in the Nonconforming Protection Control Protocol, including height data beyond the scope specified in protocol, voltage and oil quantity not conforming to the provisions of protocol, motor speed being lower than the minimum threshold value, exceeding the maximum flight radius. Check the capacity of flight control software to guarantee flight safety of UAV according to protection control protocol.

4) External interface test: Inject fault communication data in combination with external interface communication protocol of UAV flight control software, such as modifying frame header, length and content of data and sending data overflow value. Test the fault-tolerant ability of error data and overflow data processing capacity.

5) Aerial resetting test: Inject sudden power loss of UAV in flight. Check if flight control software is able to save valid data before power loss. Also, reload these data after energizing recovery and maintain valid flight status before power loss.

6) Control instruction test: Inject abnormal control instruction not specified in the protocol and verify the processing capacity of flight control software on abnormal control instructions.

7) Watchdog test: Remove watchdog program codes by annotation and check if the watchdog can send alarm during software operation.

TEST RESULTS AND ANALYSIS

The test method proposed in this paper is based on multiple test models of different test environments and types. By combining coverage and fault injection technique, it is able to test UAV flight control software in a complete and effective manner. Test cases were comprehensively designed and the software problems detected covered a wide range. The software problems detected in different test types during the whole test are shown in Table I.

It can be seen from Table 1 that 60% of the software problems are detected in static test, including document review, static analysis and code review. The result
shows that static test shall be conducted before dynamic test, which will receive better test results. It can not only eliminate the description defects of requirements and design documents but also effectively improve the correctness and standardization of program codes and discover potential defects of programs in data processing and design.

Through coverage and fault injection under semi physical test environment, dynamic test of configuration items injects cross-boundary data, abnormal data and operation fault instructions.

<table>
<thead>
<tr>
<th>Test Type</th>
<th>Quantity of Problems</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static test</td>
<td>15</td>
<td>60%</td>
</tr>
<tr>
<td>Dynamic test of configuration items</td>
<td>8</td>
<td>32%</td>
</tr>
<tr>
<td>System test</td>
<td>2</td>
<td>8%</td>
</tr>
<tr>
<td>Total</td>
<td>25</td>
<td>100%</td>
</tr>
</tbody>
</table>

Effectively discovers the defects of flight control software in processing boundary, abnormal data and interface communication, further tests the software characteristics including safety and reliability and successfully discovers the problems of some software in function and performance.

System test is conducted based on dynamic test of configuration items after all problems detected in document review, static analysis and code review return to zero. The functions, performance, safety and reliability of UAV flight control software are evaluated as a whole. As complete and valid configuration item dynamic test was conducted on some type of UAV flight control software, the software problems detected during system test have been great reduced, only 8% of all problems detected. The result does not only demonstrate the necessity of configuration item dynamic test but also prove that only dynamic test of configuration items is not sufficient and software test shall be conducted at a higher system level.

CONCLUSION

This test method proposed in this paper was adopted to build a test model based on multiple test environments and types and design several test strategies. In combination with test technique of coverage and fault injection, the flight control software of some type of UAV was tested from different perspectives in order to
ensure the test sufficiency. Test results prove that strict test shall be conducted on UAV flight control software and it is an important means to effectively guarantee the quality and reliability of UAV.

REFERENCES

1. GJB/Z102-1997 Guidelines on Software Reliability [S].