Operational Modeling of Automatic Flight Control System Software

Han Fengyan
Aviation Reliability Center
Beijing, China
fyhan@126.com

Zhang Han
College of Education
South-West University
Chongqing, China

Abstract—In order to model the operation of automatic flight control system software, based upon a set of new concepts including test profile, environmental profile recently presented by us, the technique of developing test profile from mission profile is put forward and illustrated with AFCS-1 automatic flight control system. This technique is presented by decomposing inputs, and assembling after analysis and modeling respectively. Compared with operational profile, this method of modeling, based on the mission of aircrafts, is not only more approximate to the actual instance of software operation, but also similar to the reliability test of hardware, and thus it provides a foundation for integrated reliability test of a system with hardware and software.

Keywords—operational modeling; test profile; environmental profile; automatic flight control system

I. INTRODUCTION

The operational modeling of software system is critical to the software reliability test. In 1992, Musa J.D. put forward the concept of operational profile[1,2]. In 1995, Ouabdesselam F. developed the method for constructing operational profiles of synchronous critical software[3]. In 2001, Sebastian Elbaum etc presented a refinement methodology for the generation of more accurate operational profiles [4]. In 2004, Rakesh S. etc put forth the method of educing the system operation profile of software component [5], and Mechelle G. etc extended the concept of operational profile by appending process profile, structural profile and data profile[6]. The research above is all based on operational profile. Because of the specialty of automatic flight control system (AFCS) software, the operational profile based method is difficult to use in operational modeling of AFCS. Based upon the concept of test profile [7], this paper analyzes the operational characteristics of AFCS and the advantage of test profile, and presents the operational modeling method of AFCS software by an example of AFCS-1.

II. OPERATIONAL CHARACTERISTICS OF AFCS SOFTWARE

The operational characteristics of software are determined by the input characteristics basically, so that operational modeling of software implies to describe the input of software in practical usage. As typical hard real-time software with the characteristics such as real-time, embedded, and reactive, the input characteristics of AFCS software is specified as following:

a) Multiple parallel inputs. There are many devices connected to AFCS as well as the inputs are multiple and parallel and hence. Generally, there are more then 10 input parameters including commands from the pilot, input data from sensors, and data from other equipments connected with the AFCS.

b) Time-dependent. AFCS software is typical reactive software, whose state is dependent not only on the present inputs, but also on the history of the inputs.

c) Spatial correlation. As a typical hard real-time system, AFCS software has inputs associated with each others. Some of them are associated with defined function, and some are interacted by a constraint.

Because of the particularity specified above, it is very difficult to apply operational profile to AFCS software modeling.

III. TEST PROFILE AND ITS CHARACTERISTICS

A. Related Defines of Test Profile

The inputs of AFCS software are classified into modal parameter and environmental parameter, according to the effect of inputs.

Def.1. modal parameter. k is called modal parameter, if k is supported on \( \{x_i | i = 1, 2, ..., n\} \), and any variety of k will result in the system transfer from one state to another. The set which includes all modal parameters is called modal parameter set. The value of the modal parameter is called modal data. In practical system, the state of system is determined not by certain single parameter, but a combination of modal parameter. The state space of combination is denoted as \( S = \{s_i | i = 1, 2, ..., n\} \).

Def. 2. system mode. Define \( M = \{m_i | i = 1, 2, ..., n\} \) as a division of system functional state, if there is a bijective mapping relations between M and S, then \( m_i \) is called system mode, and M is called system mode set.

Def. 3. environmental parameter. For a parameter e supported on \( e \in \{x_i | i = 1, 2, ..., n\} \), if the system output will
vary with \( e \), but the system mode is independent of the value of \( e \), then the parameter \( e \) is called environmental parameter. The set which includes all environmental parameters is called environmental parameter set. The value of the environmental parameter is called environmental data.

**Def. 4. system mode profile.** Define \( M = \{m_i | i = 1,2,...,n\} \) as system mode set, then the set \( MP = \{(m_i, p_i) | i = 1,2,...,n\} \) is defined as system mode profile, where \( p_i \) is the occurrence probability of system mode \( m_i \). In other words, system mode profile is a set including specific system mode and its occurrence probability.

**Def. 5. environmental profile.** Define \( E = \{e_i | i = 1,2,...,n\} \) as system environmental parameter set, then the set \( EP = \{X[e_i(t)] | i = 1,2,...,n, t \in [0,T]\} \) is called environmental profile, where \( X[e_i(t)] \) is a stochastic process description of variation of environmental parameter during the mission profile \([0,T]\).

**Def. 6. test profile.** Define \( MP \) as system mode profile, and \( EP \) as environmental profile, then \( TP = \{MP, EP\} \) is called test profile.

Test profile is the integration of system mode profile and system environmental profile. It is a abstract description of actual usage of software system.

**B. Characteristics of Test Profile**

Software test profile use time series based framework to describe the usage of software from the perspective of system operation. Comparison between test profile and operation profile is shown in table 1.

### TABLE I. COMPARISON BETWEEN TEST PROFILE AND OPERATION PROFILE

<table>
<thead>
<tr>
<th>test profile</th>
<th>operation profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Educed from mission profile</td>
<td>Educed from function profile</td>
</tr>
<tr>
<td>Concentrate on software performance of completing mission in actual situation</td>
<td>Concentrate on software performance of executing specific functions with requirement of specification</td>
</tr>
<tr>
<td>Can be developed by user and tester. It does not need to have in-depth understanding of the software.</td>
<td>Developed by user, developer and test engineer. It needs in-depth understanding of the software.</td>
</tr>
<tr>
<td>Convenient for describing time-correlative data. It is based on objective statistical data and is easy to determine.</td>
<td>Convenient for describing isolated input points. Operation probability is somewhat subjective and Sometimes difficult to determine.</td>
</tr>
<tr>
<td>Be similar to hardware test profile, and it can be applied to software development.</td>
<td>It’s not compatible with hardware reliability test.</td>
</tr>
<tr>
<td>It’s useless in other works of software developing process.</td>
<td>Not only it can be used in test but also can be used in other works of software developing process.</td>
</tr>
</tbody>
</table>

**IV. OPERATIONAL MODELING PROCESS**

**A. Decomposition of Environmental Data**

One of the difficulties in AFCS operation modeling is that the environmental data is both certainty and randomicity. So we can suppose that environmental data could be decomposed to stochastic component and control (certainty) component.

Hyp. Environmental data sequence can be decomposed to mutual orthogonal components (control component and stochastic component).

\[
X_E = X_S + X_C
\]

Where \( X_E \) denotes environmental data sequence, \( X_S \), \( X_C \) denotes the stochastic component and the control (certainty) component respectively.

As environmental data can be decomposed to two components, we can establish models to describe stochastic component and certainty component respectively.

**B. Generation Process of Test Profile**

We can develop test profile according to two steps. First, generate system mode profile and environment profile; second, combine two profiles to get test profile. It’s easy to get system mode profile, so we mainly discuss the generation of environment profile.

1) **Determination of the mission profile**

Mission profile is the basis of environment profile for it determines environment condition of system during the mission. For certain aircraft, the mission profile is usually specified. But in the establishment of test profile, the mission profile may be tailored or complemented according to actual requirement.

2) **Determination of environmental parameter**

This process includes several works, such as analysis of system input environmental parameter, definition of parameter, determination of the type, range, and accuracy of parameters.

3) **The Analysis of Environmental Parameter Value**

We can identify the value of each environmental parameter in every system mode of each mission profile.

4) **The Relevant Analysis of Environmental Parameter**

The main purpose is to identify the relationship and constraint among the parameters. The space relevancy includes the relationship among the parameters, as well as the relationship between environmental parameter and mode parameters. The time relevancy mainly embodies in the control component of the environmental parameters.

5) **The Merge of Environmental Profile**

The result of the environmental parameter analysis can form an uniform environmental profile through engineering treatment. The process of the treatment can refer to the merge methods of the hardware reliability test profile.

6) **The Merge of Test Profile**

The merge of test profile is to merge system mode profile and environmental profile as a uniform test profile. Firstly,
embed the system mode profile into mission profile. Then, embed environmental profile into mode profile corresponding. Thirdly, merge multiple mission profiles which consist of mode profile and environmental profile.

V. CASE ANALYSIS

Now taking the AFCS-1 automatic flight control system software as an example, we particular describes the modeling process, that is the generation process of software test profile.

A. Mission Profile

The typical mission profile of the AFCS-1 is shown in Table 2.

<table>
<thead>
<tr>
<th>Mission</th>
<th>Time proportion</th>
<th>Altitude (km)</th>
<th>Speed (km/h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ground Start</td>
<td>3</td>
<td>0.15</td>
<td>0</td>
</tr>
<tr>
<td>Takeoff Climb</td>
<td>5</td>
<td>0.15-0.3</td>
<td>0-480</td>
</tr>
<tr>
<td>Cruise</td>
<td>60</td>
<td>0.3</td>
<td>440</td>
</tr>
<tr>
<td>Middle Altitude</td>
<td>5</td>
<td>5</td>
<td>465</td>
</tr>
<tr>
<td>Low Altitude</td>
<td>10</td>
<td>0.3</td>
<td>450</td>
</tr>
<tr>
<td>Bombing</td>
<td>8</td>
<td>5</td>
<td>465</td>
</tr>
<tr>
<td>Middle Altitude</td>
<td>4</td>
<td>0.3</td>
<td>455</td>
</tr>
<tr>
<td>Descent Landing</td>
<td>5</td>
<td>0.3-0.15</td>
<td>260</td>
</tr>
</tbody>
</table>

TABLE II. TYPICAL MISSION PROFILES

B. System Mode Profile

There are four operating conditions in AFCS-1, including stand-by state, operating state, self-check state and fault state.

Operating state can be divided into some concrete states. After analysis, we can form the system mode profile, such as Table 3.

C. Environmental Profile

There are 14 environmental parameters for AFCS-1 system, among which 11 signals are received from the aircraft bus and the sensors, and 3 signals operated by pilots.

The distribution of 3 pilots operated parameters, can be presented by the experience of the pilots. In order to get the distribution of other environmental parameters, we implemented flight test. Flight test covers all the operating condition of the mission profile.

D. Test Profile

From the result of the analysis above, we can conclude the software test profile of the AFCS-1. Test profile can be expressed as a complex table, such as figure 1. The test profile consists of 8 mission segments, including start-up, climb, cruise1, cruise2, and so on. Every mission segment consists of different system modes, every system mode is correspondence with 14 environmental parameters, and each parameter consists of control component and random component.

<table>
<thead>
<tr>
<th>State</th>
<th>Detail State</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-by State</td>
<td>Stand-by State</td>
<td>0.01</td>
</tr>
<tr>
<td>Operating State</td>
<td>Steady Mode</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Longitudinal Stability</td>
<td>0.12</td>
</tr>
<tr>
<td></td>
<td>Lateral Stability</td>
<td>0.08</td>
</tr>
<tr>
<td></td>
<td>Omnidiirectional Stability</td>
<td>0.42</td>
</tr>
<tr>
<td></td>
<td>Control Mode</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Longitudinal Control and Lateral Stability</td>
<td>0.07</td>
</tr>
<tr>
<td></td>
<td>Longitudinal Control and Lateral Cut off</td>
<td>0.05</td>
</tr>
<tr>
<td></td>
<td>Lateral Control and Longitudinal Stability</td>
<td>0.12</td>
</tr>
<tr>
<td></td>
<td>Lateral Control and Lateral Cut off</td>
<td>0.05</td>
</tr>
<tr>
<td></td>
<td>Lateral and Longitudinal Control</td>
<td>0.06</td>
</tr>
<tr>
<td>Self-check</td>
<td>Self-check before flying</td>
<td>0.02</td>
</tr>
</tbody>
</table>

Figure 1. the Structure of AFCS-1 Test Profile
VI. CONCLUSION AND DISCUSSION

The real analysis of the AFCS-1 system software shows that test profile is more suitable for the operational modeling of AFCS software.

a) Test profile and generation method of test profile based on mission profile, which can describe the operational process clearly and accurately.

b) The software reliability test technique based on test profile, which is similar to hardware reliability test, establishes the foundation for the reliability test of integrated software and hardware.

c) Although the basic concepts including test profile, environmental profile, as well as the test profile developing technique based on mission profile, are put forward for AFCS software, they are also suitable for other software which has the same properties.

REFERENCES