

doi: 10.3969/j.issn.1006-1576.2012.09.027

## 飞行控制软件测试用例生成技术

董成基, 齐杏林, 吕静, 赵志宁

(军械工程学院弹药工程系, 石家庄 050003)

**摘要:** 为进行飞行控制软件测试, 采用扩展有限状态机 (extended finite state machine, EFSM) 模型对弹药飞行控制软件测试需求建模和测试用例的生成展开研究。将时间约束作为变量处理和并发任务的平坦化分析进行研究, 通过分析生成了模拟软件完整执行过程的测试序列, 采用组合设计解决了各输入变量不同取值之间的组合问题, 并提出为变量增加时间属性来描述产生测试信号的时刻。以某弹药飞控软件为例, 进行建模和测试用例生成实践, 验证了方法的可行性。该研究为有限状态机模型在飞控软件和其它实时软件测试中的应用进行了探索。

**关键词:** 飞行控制软件; 测试用例; EFSM; 组合设计; 时间属性

**中图分类号:** TJ410.6 **文献标志码:** A

## Test Case Generation Technique for Flight Control Software

Dong Chengji, Qi Xinglin, Lu Jing, Zhao Zhining

(Dept. of Ammunition Engineering, Ordnance Engineering College, Shijiazhuang 050003, China)

**Abstract:** In order to test flight control software, use (extended finite state machine) EFSM model to research on the ammunition flight control software test requirement modeling and testing example generating. Denoted the time restriction as variables and flatten co-current missions are researched. Generate test sequence for simulated software execution procedure by analysis. Use combination design to solve the different values combination of each input variables and introduce the method that add time attribute for variables to describe the test single generation time. Taking flight control software of certain type ammunition as example, model and test case generation are carried out. It proves the method is available. It is a useful research for using EFSM model in flight control software and other real-time software testing.

**Key words:** flight control software; test cases; EFSM; composite design; time attribute

### 0 引言

基于需求建模生成测试用例的方法已被用于软件测试中<sup>[1]</sup>。在飞行控制软件测试方面, 刘久富<sup>[2]</sup>等采用虚拟原型技术 Statemate 工具对飞控软件进行建模, 但是未给出具体的测试用例生成算法。高猛提出使用统一建模语言 (unified modeling language, UML) 状态图对飞行控制软件进行需求建模<sup>[3]</sup>, 但只限于导弹发射升空前的测试, 对后续的控制等关键过程的测试并未深入。殷永峰等用基于 UML 实时扩展的嵌入式软件测试建模方法对航空电子系统嵌入式软件测试用例生成进行了介绍<sup>[4]</sup>, 但忽略了软件运行中的许多细节问题。

上述工作对测试需求建模做了很好的研究, 但都未深入到飞行控制软件的细节, 其测试用例不能模拟软件真实的运行环境, 对错误检测能力也未进行分析。因此, 笔者采用扩展有限状态机 (extended finite state machine, EFSM) 模型对弹药飞行控制软件测试需求建模和测试用例的生成展开研究, 着重实现对飞控软件功能的全面覆盖并提高测试的错误

检测能力。

### 1 基于有限状态模型的测试理论

有限状态机 (finite state machine, FSM) 与 EFSM 都是通过状态和状态之间的转换来对软件的行为进行建模。EFSM 是对 FSM 的扩展, 两者的基本原理是一样的。利用 EFSM 对 FSM 的扩展可以更准确地对软件建模, 而生成测试用例则需要将 EFSM 转换为 FSM, 进而利用 FSM 模型上的测试用例生成算法来生成测试用例。有限状态机模型可以用状态图直观地表示。在状态图中, 用节点表示状态, 用边表示转移, 边标识描述了对应转移的输入/输出情况。FSM 由输入符号串激活, 激活后从一个状态转换到下一个状态, 并产生相应的输出。对于测试而言, 输入串即为测试用例的输入。

FSM 中有 4 类故障模型: 操作错误、转换错误、冗余状态错误和缺失状态错误。测试用例必须保证能够检测出上述故障模型, 所以在测试一个从状态  $s_i$  到状态  $s_j$  转移时, 测试用例应该具备 3 个基本条件<sup>[5]</sup>:

收稿日期: 2012-04-23; 修回日期: 2012-05-07

作者简介: 董成基(1988—), 男, 四川人, 在读硕士研究生, 从事机电系统检测技术、嵌入式软件安全性研究。

- 1) FSM 能够被置到状态  $s_i$ ;
- 2) 对一个输入序列  $a_k$ , 输出与预期相符;
- 3) 确认转移后的新状态是预期的状态  $s_j$ 。

由于 EFSM 中引入了输入、输出、赋值和条件 4 种动作和全局变量, 因而状态的转换不仅依赖于输入, 而且可能依赖于全局变量, 因此测试要求还应增加一条: 在一条路径中的每一个有意义的变量都有一个确定的值<sup>[6]</sup>, 如此才能确定一个转移对应的输出和后继状态。

## 2 弹药飞行控制软件测试难点分析

某型弹药飞行控制软件的简化状态图见图 1。其主要状态包括起飞前状态、主动段飞行状态、末修段飞行状态, 其中主动段飞行状态和末修段飞行状态都是复合状态, 包含并发子状态。姿态控制是周期事件, 每 20 ms 读取惯性组合输出信号, 并进行姿态控制, 在进行姿态控制的同时, 定时中断到来的时候, 程序会转到中断处理程序发送时序命令, 处理完中断程序之后再回到主程序。中断处理的时间如果过长, 影响到稳定和制导控制就会发生偏差, 或时序紊乱等问题, 影响弹药实现其功能。

对飞行控制软件进行测试的难点有:

- 1) 软件的行为有严格的时序约束, 而 EFSM 模型中未提供对时间的描述机制;
- 2) 测试人员无法控制发送的时序命令和读取惯性组合的输出信号等行为的时间约束;
- 3) 在循环地进行姿态控制时, 还要发送时序命令, 只对循环测试一次, 不能模拟软件真实的运行, 从而漏掉测试项;
- 4) 注意中断子程序对主程序的影响。

## 3 测试用例生成

### 3.1 平坦化状态图

利用状态图生成测试序列, 首先要将状态图做平坦化处理, 使其不含嵌套状态。平坦化处理主要涉及对时间约束的描述和对并发子状态的分解。该飞控软件的时间约束主要体现在周期事件和定时中断上。用时间扩展机制对软件时间约束进行描述<sup>[4]</sup>要引入超时事件和周期事件等, 处理不方便; 因此, 笔者将时间作为一个变量来处理, 在前置条件中体现时间约束, 以简化处理过程。

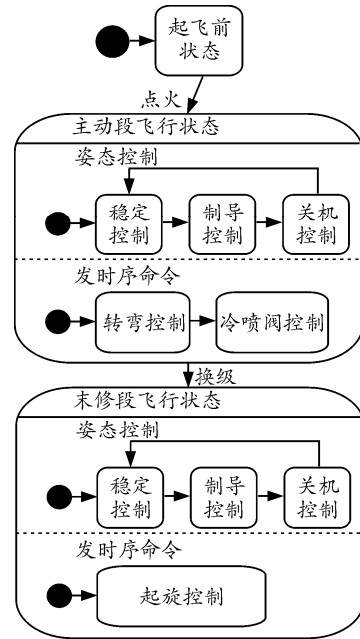


图 1 某型弹药飞控软件简化状态图

图 1 中的发时序命令和姿态控制 2 个并发子状态是在一个时间段内的并发, 实际上是一个中断处理程序, 通过时间上分段使用处理器来实现并发。平坦化的状态图如图 2 所示, 为了简化图形, 省去了末修段, 将状态图的边的完整标识用表 1 列出, 只在图中标出前置条件和触发事件。表 1 中的“ $t=t_1$ ”等即为对时间约束的描述。

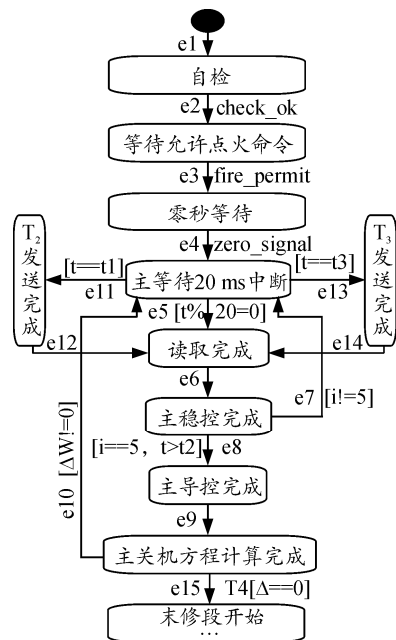


图 2 某飞行控制软件平坦化状态图

表 1 某飞行控制软件状态图的部分边标识

边	边标识(输入[监护条件]/输出)
e1	NULL [NULL] /NULL
e2	check_ok [NULL] /para_set
e3	fire_permt [NULL] /send(T1)
e4	zero_signal [NULL] /NULL
e5	tper_interr, a,w, [t%20==0]/output11(stab_contr), i++
e6	NULL [i!=5] /NULL
e7	[i==5, t>t2] / output12(navi_contr), i:=0
e9	NULL [Dw!=0]/NULL
e10	t1_interrupt [t==t1] /send(T2)
e14	NULL [Dw==0] /NULL
e23	NULL [h==h2] /output3(time_expl)
...	...

### 3.2 测试生成算法选择及测试序列生成

基于 FSM 的测试生成常用算法有转换路径法 (transition tour, TT)、特征序列法(也叫 W 法)、单一输入/输出串法 (unique input/output sequence, UIO)<sup>[7]</sup>, 飞控软件的 EFSM 模型中, 大部分转移都有前置条件约束, 即使是顺序转移, 也存在时间这一潜在的约束因素。这些约束使得 TT 法和 W 法都不适用。UIO 法为每一个转移构造一个遍历  $TE(t)=Phead(t)(S1) \cdot lable(t) \cdot UIO(tail(t))$  来作为测试用例, 其中  $Phead(t)(q1)$  为从初始状态  $S_1$  到转移  $t$  的最短路径。lable(t)是转移  $t$  的边标识,  $UIO(tail(t))$  是能将转移  $t$  的尾状态与其他状态区分开来的符号串。在 TT 法中, 如果不考虑路径最短, 按照实现功能的步骤一步一步转移, 就可避免与前置条件发生冲突, 另外时间约束的存在, 使得每个转移的标识即是自身的区分串。

根据上述分析及系统测试的需要, 笔者采用 UIO 法, 将其扩展为多输入/输出串 (multiple input/output sequence, MIO) 并限定软件的接收状态只有终态。在不考虑容错处理的情况下, 一个完整的测试序列即为按照时间约束从初态到终态的一条路径, 时间的约束使得该用例覆盖了软件的所有状态 and 所有迁移, 满足覆盖准则。

该飞控软件的一个完整的测试序列由每个转移中外部可见的输入/输出的连接组成:  $check\_ok/NULL \cdot trajectory, turn\_eigenvalue/NULL \cdot fire\_permt/T1 \cdot zero\_signal/NULL \cdot \{a,w/stab\_contr1\}^1 \cdot NULL/T2 \cdot \{\{a,w/stab\_contr1\}^i \cdot NULL/navi\_contr1\}^m \cdot \{a,w/stab\_contr1\}^4 \cdot NULL/T3 \cdot \{\{a,w/stab\_contr1\} \cdot NULL/navi\_contr1\} \cdot \{\{a,w/stab\_contr1\}^i \cdot NULL/navi\_contr1\}^n \cdot NULL/T4, T5 \cdot \{\{a,w/stab\_contr2\}^i \cdot NULL/navi\_contr2\}^x \cdot NULL/T6 \cdot \{\{a,w/stab\_contr2\}^i \cdot NULL/navi\_contr2\}^y \cdot NULL/t\_explode\_signal$ 。

其中由“{}”括起来的部分表示多个相同的输

入输出对的重复, 个数由括号右上方的上标表示; “NULL”表示输入或输出为空。

### 3.3 选取测试数据

测试序列确定了各输入变量的时序, 而测试用例不仅包括测试序列, 还包括测试数据。变量的合法取值往往不止一个, 对于连续变量, 其取值个数是无穷的, 对于这类变量的选取一般采用等价类划分和边界值分析来选取, 然而各个变量不同取值之间的组合又是一个难以处理的问题。组合测试 (combinatorial interaction testing, 也叫 t-way 测试) 的方法<sup>[8]</sup>可以生成输入变量之间的  $t$  元组合来替代笛卡尔乘积的组合, 并有实验表明最多需要 6 元组合即可发现程序中的所有组合故障。

除了变量取值外, 还需要考虑每个输入变量的时间约束。可以为每一个变量增加一个时间属性。该属性值可以准确地确定, 在测试活动中, 该在何时将该变量输入到被测系统中。若变量需在  $t_i$  时刻输入, 则通过在  $t_i$  上下浮动的时刻输入变量, 还可以测试软件的实时性。

## 4 测试用例分析

### 4.1 覆盖分析

根据 FSM 生成测试用例, 其覆盖准则有状态覆盖、转换覆盖、转换对覆盖准则和边界-内部覆盖<sup>[9]</sup>, 可以看出, 文中的测试序列是从初态到终态的一条模拟软件完整执行的路径, 满足上述 4 条覆盖准则, 也满足转换后状态确认的条件。

### 4.2 错误检测能力分析

在软件中有 2 类关键错误: 数据的修改错误和添加错误<sup>[10]</sup>, 其他的错误都可以由这 2 类错误概括。因为测试序列是从初态到终态的一条模拟软件完整执行的路径, 而且测试数据的选择考虑了用于计算使用变量的不同取值组合, 数据修改错误和添加错误都能被检测出来, 该方法生成的测试用例的故障检测能力也满足要求。

## 5 总结

笔者将时间视为一个变量, 解决了状态图的实时扩展问题, 并为输入变量添加时间属性, 解决了产生测试信号的时间问题。由于时间约束等限制, 使用有 EFSM 对飞控软件建模时, 不能直接使用基于 FSM 的算法直接生成测试用例, 无法发挥 FSM 在协议测试中强大的故障检测能力; 因此, EFSM

对于飞控软件测试的作用主要在于：是否准确描述软件的功能和行为，以及理清测试思路。

参考文献：

[1] Chu, P. Y. M., Liu, M. T. Global state graph reduction techniques for protocol validation in the EFSM model[C]. Eighth Annual International Phoenix Conference on Computers and Communications, 1989: 371-377.

[2] 李志宇, 刘久富, 杨忠, 等. 基于模型的飞行控制软件测试用例的生成[J]. 吉林大学学报: 工学版, 2006, 36(4): 543-548.

[3] 高猛. 实时嵌入式软件系统测试需求建模研究[J]. 航天控制, 2010, 28(5): 64-69.

[4] 殷永峰, 郑本焘, 陆民燕, 等. 基于 UML 实时扩展的嵌入式软件测试用例生成技术[J]. 系统工程与电子技术, 2011, 33(3): 694-699.

[5] Dahbura A. T., Sabnani K.K., Uyar M. U. Formal methods for generating protocol conformance test sequences[C].

Piscataway, New Jersey: Proceedings of the IEEE, 1990, 78(8): 1317-1326.

[6] Chang Jia Wang, Ming T. Liu. Axiomatic test sequence generation for extended finite state machines[C]. California: Proceedings of the 12th International Conference on Distributed Computing Systems, 1992: 252-259.

[7] A. P. Mathur. 软件测试基础教程[M]. 王峰, 等. 译. 北京: 机械工业出版社, 2010.

[8] Yu Lei, Raghu Kacker, Richard Kuhn D, et al. IPOG: A General Strategy for T-Way Software Testing[C]. Tucson, Arizona: 14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems, 2007: 549-556.

[9] 刘攀, 缪淮扣, 曾红卫, 等. 基于 FSM 的测试理论\_方法及评估[J]. 计算机学报, 2010, 34(6): 965-985.

[10] Tae Hyong Kim, Ik Soon Hwang, Min Seok Jang. Test case generation of a protocol by a fault coverage analysis[C]. Koganei, Tokyo, Japa, 1998: 690-695.

\*\*\*\*\*

(上接第 92 页)



图 7 数据评估显示图

表 1 统计结果

统计量	统计结果值	统计量	统计结果值
X/m	8.445 69	Vx/(m/s)	0.108 84
Y/m	11.341 32	Vy/(m/s)	0.370 21
z/m	3.670 97	Vz/(m/s)	0.168 14
经度误差/m	14.010 25	北速/(m/s)	0.088 08
纬度误差/m	5.005 70	天速/(m/s)	0.202 18
高度误差/m	6.632 22	东速/(m/s)	0.358 52

为验证评估的正确性，把数据文件 2011-06-02-16-57-31.txt 导入 OriginPro 软件，进行误差分析，得到的结果如表 1 所示。从图 5、图 6、

图 7、表 1 可以看出，软件平台显示的位置、速度信息和测试场景基本一致，统计结果正确，可见软件平台的实用性和正确性。

4 结论

抗干扰多模接收机人机交互平台整体结构紧凑，可靠性高，使用方便，满足了功能目标中提出的多样化、综合性、直观性的要求。该平台发挥了人机交互的优越性，为接收机的设计和定型起到重要作用。

参考文献：

[1] 袁建平, 罗建军, 岳晓奎. 卫星导航原理与应用[M]. 北京: 中国宇航出版社, 2004. 09.

[2] Lian Yuanfeng, Zhao Yan. The Design and Implementation of Performance Evaluation System for satellite navigation system[J]. Control&Automation, 2010, 26(4-1): 213-216.

[3] Luo Zhikun, Teng Zhaoshen. The Design and Implementation of Performance Evaluation System for satellite navigation system[J]. Chinese Journal of Scientific Instrument, 2009, 30(7): 1492-1497.

[4] 孙强, 马大玮, 张楠. 复杂电磁环境下 TDSCDMA 抗干扰性能仿真评估[J]. 四川兵工学报, 2010, 31(8): 76.

[5] 龚新文. 串口通信在 VS2008 中的实现与应用[J]. 电脑与电信, 2011, 3(27): 47-48.

[6] 梁林灿. 1553B 总线测试系统中上位机软件的设计与实现[D]. 广州: 中山大学硕士学位论文, 2007(5).