文章编号:1671-4598(2024)02-0050-06 DOI:10.16526/j.cnki.11-4762/tp.2024.02.008 中图分类号:TP311.55 文献标识码:A

一种无人机飞行控制软件自动测试方法

刘玉军，赵创新，王振华，刘国强，彭 怡
（航空工业成都飞机工业（集团）有限责任公司，成都 610000）

摘要：针对机载飞行控制软件测试依赖硬件系统、测试周期长、手动测试方式效率低、重用性差、易出错和维护成本高等问题，研究了全数字仿真环境的关键技术，提出了一种机载飞行控制软件的自动测试方法；该测试方法在全数字仿真环境中进行测试，并对传统的测试数据生成算法进行改进，使用AETG-SA算法生成测试数据，将测试结果的反馈引入到算法中，动态调整算法参数，获得最优测试集合，提高了测试覆盖率；在全数字仿真环境中执行测试用例，减少了嵌入式软件测试过程中对硬件的依赖，对系统功能测试和故障模拟测试的覆盖更加全面；工程实践表明，基于全数字仿真环境的自动测试方法相较于传统全实物和半实物的测试方法测试充分性提升了4%，测试时间缩短了44%。

关键词：飞行控制软件；机载软件；全数字仿真环境；AETG-SA算法；自动测试方法

Automatic Test Method of UAV Flight Control Software

LIU Yujun，ZHAO Chuangxin，WANG Zhenhua，LIU Guoqiang，PENG Yi
（AVIC Chengdu Aircraft Industrial (Group) Co.，Ltd.，Chengdu 610000，China）

Abstract：For the problems of depending on hardware system，long test cycle，low efficiency of manual test，poor reuse，error-prone and high maintenance cost in airborne flight control management software test，the key technology of full digital simulation environment is researched，an automatic test method for airborne flight control management software is proposed. Based on the full digital simulation environment，this test method improves traditional test data generation algorithms，uses the automated expert test generator-simulated annealing (AETG-SA) algorithm to generate the test data，introduces the feedback from test results into the algorithm，dynamically adjusts the algorithm parameters，obtains the optimal test set，and improves the test coverage. Test cases are executed in full digital simulation environment，it reduces the hardware dependency of embedded software testing process，and higher coverage of system function test and fault simulation test，and improves the efficiency of the test cases greatly. Compared with the traditional full-physical and semi-physical test methods，the engineering practice shows that based on the full digital simulation environment，the sufficiency of the automated test method improves by 4%，and the testing time reduces by 44%.

Keywords：flight control software；airborne software；full digital simulation environment；AETG-SA algorithm；automatic test method

0 引言

近年来，软件成为航空装备的大脑和灵魂，机载软件信息化程度越来越高，需要完成大量数据采集、处理和信息融合，其复杂程度高，具有较强的实时性[1-2]，其质量不容有失，对软件测试的要求也在不断提高[3-5]。机载软件的规模、复杂程度呈增长趋势。装备训练贴近实战，机载软件质量缺陷不断暴露，软件的功能和性能需要持续优化，软件的变更和维护升级极为频繁[6]。

机载飞行控制管理系统为嵌入式系统，由飞控计算机和飞行控制管理软件（简称飞行控制软件）组成，飞行控制软件运行在飞控计算机中，实现对无人机的飞行控制、模式控制、应急处置、余度管理等功能。在传统的研制模式中，飞行控制管理系统的研制首先要进行系统设计，系统设计完成后，硬件团队开始进行硬件设计和调试样机的生产。软件团队要在硬件团队提交硬件设备后才能进行开发和测试。这种开发方式存在以下问题：1）研制周期较长，软件设计要等待硬件的选购、布线、制版、加工、调试，需要大量的时间，软件的开发人员和测试人员无法及时参与其中；2）由于进度、环境等原因，多配置项的系统集成和联试只能在主机联试时开展，同时不具备增量的、自动测试能力，导致软件无法响应需求的快速变更；3）机载硬件成本较高，在全实物系统的测试环境和半实物系统的测试环境中很多系统故障无法构造，如测试飞控计算机存储器故障下，飞行控制软件对数据的处理等。由测试环境导致的测试用例无法执行的数量越多，软件测试验证就越不充分，后期的无人机飞行就存在质量风险；4）软件的测试自动程度低，测试环境构建周期长、成本高[7]。

在测试用例的设计和执行方面，飞行控制管理软件的测试用例数量多、执行周期长，如何既能控制测试数据的数量，又可以满足语句覆盖率、分支覆盖率、MC/DC覆盖率等覆盖率的要求，一直是困扰测试人员的难题。

收稿日期:2023-06-14； **修回日期**:2023-07-19。
作者简介:刘玉军(1988-)，男，硕士，工程师。
引用格式:刘玉军，赵创新，王振华，等.一种无人机飞行控制软件自动测试方法[J].计算机测量与控制，2024，32(2)：50-55.

针对以上问题，本文介绍了一种机载飞行控制软件自动测试方法，测试数据的生成采用自研 AETG-SA 算法，并加入了反馈机制，提升了测试覆盖率。在全数字仿真平台的基础上搭建测试环境，测试用例自动执行并记录测试结果，使飞行控制软件的快速原型验证和测试不再过度依赖硬件，大大缩减了系统研制周期，同时提升了系统的可靠性与安全性。

1　全数字仿真环境构建

1.1　全数字仿真环境介绍

全数字仿真环境采用分布式网络架构和模块化设计，可以在通用平台（如 Windows）模拟真实的嵌入式硬件，包括芯片和外围器件[8]。嵌入式软件可运行在虚拟目标机上，实现软件设计的各项功能。全数字仿真环境可以对所有 CPU 寄存器、片上器件、内存的状态进行显示和修改，支持 ARM、DSP、PowerPC、龙芯、飞腾等[9]，具备单步、断点、变量监视等常用调试功能，系统的状态可以保存和恢复，提供外围设备控制器仿真库，具备可扩展能力[10]，支持各种仿真模型的扩展开发和接入，提供多种 API 接口。基于全数字仿真环境可以实现嵌入式系统的快速原型研制，系统故障注入，处理器被执行代码单元级测试等多项仿真、软件验证和测试功能。全数字仿真环境架构如图 1 所示。

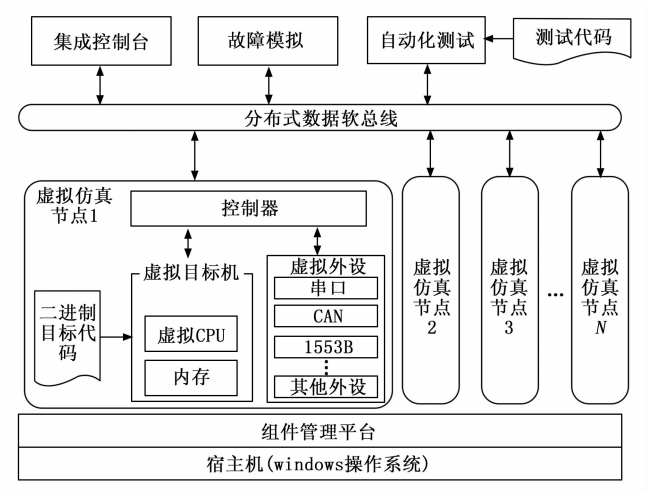

图 1　全数字仿真环境架构图

1.2　全数字仿真环境的关键技术

1）基于 AADL 的系统架构建模技术：AADL（架构分析设计语言）描述嵌入式实时系统的软件体系结构和执行平台体系结构的语言。AADL 为嵌入式实时系统（如航空电子等系统）的体系结构建模提供标准而精确的方法，可以对系统属性进行分析，并支持实现系统集成[11-12]。

2）基于 DML 的虚拟硬件建模技术：硬件虚拟化技术通过硬件模型和虚拟机两部分实现嵌入式软件的虚拟运行环境。DML（设备建模语言）提供一整套虚拟硬件模型构建接口和方法，实现嵌入式软件运行所依赖的硬件环境虚拟化。

3）基于 DDS 的网络通信技术：DDS（数据订阅/分发服务）是按照数据订阅分发服务标准实现数据订阅分发服务网络中间件 DDS 构建以数据为中心的数据订阅/分发网络，广泛应用于未来机载航电通信网络[13-14]，实现数据的稳定、高效、透明传输。

4）动态二进制翻译技术（Dynamic Binary Translation）：二进制翻译技术是一种即时编译技术，指的是从一种指令集翻译到另一指令集的技术。动态二进制翻译技术采用边运行边翻译的方式，将源机器码翻译成目标机器代码[15-16]它将源体系结构的二进制机器指令动态编译为可在目标体系结构上运行的代码[17]。全数字仿真环境中的底层翻译控制模块根据源机器码解码、翻译、优化编码，并将输出的机器代码送到执行模块，执行源机器代码，仿真目标系统及外围器件。

1.3　飞行控制系统全数字仿真环境构建

全数字仿真环境可以通过界面图形拖拽方式快速组建。利用全数字仿真环境可以有效地提高嵌入式软件的自动程度[18]，具体构建步骤如下：

1）全数字仿真平台可以直接虚拟出目标系统 CPU，按系统设计，首先选择 CPU 型号并进行配置，设置 ROM 和 RAM 的大小及其他参数。

2）搭建外设接口，将与飞行控制系统的外围接口如 CAN、RS422 等进行仿真，通过协同仿真时间同步保证系统运行时的时序一致。

3）搭建与飞行控制系统交联的设备或仿真软件，建立资源库，可快速建立复用场景。

1.4　飞行控制软件测试数据自动生成算法

机载软件对安全性、可靠性、稳定性的要求比普通软件严格得多。飞行控制软件的安全运行受很多因素的影响，软件测试人员对各个因素组合全部进行测试才能充分的发现软件问题，但这显然是不切实际的。因为测试用例的组合数量是非常巨大的，有些时候无法穷举并进行测试。根据航空无线电技术委员会发布的 DO-178C 标准，飞控软件的测试要达到 MC/DC 覆盖级别。有效的测试用例组合既可以最大限度地发现软件缺陷，又可以满足 MC/DC 覆盖率要求。

为提升测试效率，需要设计一种算法可以自动生成有效的测试数据，对被测软件进行测试。

解决测试用例自动生成的算法包含贪婪算法、随机方法、启发式搜索算法等[19]。在软件测试领域，郑燕妮等提出蚁群模拟退火算法约简测试用例[20]，仲晓敏等提出基于退火遗传算法的多路径测试用例生成[21]，傅博提出基于模拟退火遗传算法的软件测试数据自动生成[22]。以上算法都是软件测试用例的生成或约简，对飞行控制软件的适用性不强，且算法并未将生成的测试用例的执行结果进行反馈，从而动态调整算法参数，在软件测试过程中不能形成闭环。

以下对两种算法进行研究，并提出一种测试数据自动生成算法 AETG-SA（Automated Expert Test Generator-Simulated Annealing）算法，既可以满足 MC/DC 覆盖率要

求，又可以在测试过程中动态调整算法参数，从而获得最优测试集合。

贪婪算法是测试用例生成应用较为广泛的算法，其核心思想是使用局部最优解近似全局最优[23]，算法效率较高，生成速度快，但容易产生冗余，参数维度较高时不易产生理想的测试用例集。

一种贪婪算法 AETG 生成测试用例集合的基本框架如下：

输入：待测软件 FMS 参数及范围

输出：测试数据集合 T_s

a) S=Generate(FMS) //根据参数范围生成目标集合；

b) 随机选择参数 τ 的一个取值 P，使得 P 在 S 中出现的次数最多；

c) 生成一条测试数据 $\tau[k]=P_k$；

d) $P_i(i\neq k)$选择 P_i 的某个值来扩展测试数据 τ，最大限度地覆盖 S 中的组合；

e) 测试数据 τ 加入 T_s；

f) 以相同的方法扩展 P_{i+1}直至所有的参数取值完毕；

g) 返回 T_s。

贪婪算法只能遍历有限范围内的测试用例，收敛速度慢，选择的测试用例也并不一定是最优的。

模拟退火算法（SA，simulated annealing）是基于 Monte-Carlo 迭代求解策略的一种寻优算法，由高温到低温，遍历搜索空间，搜索过程中使用到 Metropolis 准则[24]。若系统在温度 T 时达到热平衡，系统在指定状态的概率 $P_T(S)$ 为：

$$P_T(S)=\frac{e^{-Es/KT}}{\sum_{\tau\in s}e^{-E_{s0}/KT}}$$

其中：ES 为 s 状态下的能量，K 为 Boltzmann 常系数，S 是所有可能状态的集合，Metropolis 准则给出新状态 SN 的接收准则为若 $E(S_N)\leqslant E(S_o)$ 则新状态自动接受，若 $E(S_N)>E(S_o)$，新状态的接收概率取决于以下概率函数：

$$P(AccS_N)=e^{-\frac{E(S_n)-E(S_0)}{KT}}$$

飞行控制软件参数多为实际意义的变量，主要以浮点数、整数为主，在逻辑判断时也会用到布尔类型[25]。根据飞行控制软件的数据特点，结合测试经验，若某条测试数据发现软件问题，相邻的测试数据也容易发生软件问题。此时，降低降温速率的值，生成更多相邻数据，尽可能找到引发软件问题的数据边界。在贪婪算法上引入模拟退火算法并进行进一步改进，设计一种测试用例自动生成的算法 AETG-SA。AETG-SA 将贪婪算法和模拟退火算法进行融合，融合的过程为：首先采用贪婪算法生成全部目标集合 S 和测试数据 τ_o，在生成下一组数据 τ_n 时采用模拟退火算法，根据降温速率和当前数据获取下一组测试数据，再依据 Metropolis 准则判断是否接受新解 τ_n，若接受则将 τ_n 纳入测试数据集合 Ts，生成初步的测试数据集合 Ts。测试进行的过程中根据测试数据是否发现软件缺陷动态调整模拟退火算法中的降温速率 $rate$，形成最终的测试数据集合。

算法基本流程框架如下：

设软件模块累加的 MC/DC 覆盖率为能量 E，集合 S 为初始温度，集合 T_s 为达到热平衡的温度，降温速率 $rate$。

输入：待测软件参数及范围、降温速率 $rate$、判断退出条件次数 m，覆盖率差值

输出：测试数据集合 T_s

a) $InitT_s$,$rate$,m, //初始化；

b) S=Generate(FMS) //根据参数范围生成目标集合；

c) τ_o=random(S)//在 S 中随机生成一组测试数据；

d) τ_n=Get(τ_o,$rate$)//根据降温速率和当前数据获取下一组测试数据；

e) if $E(\tau_n)>E(\tau_o)$ or $e^{-[E(\tau_n)-E(\tau_o)]/KT}>random[0,1)$;//判断是否接收新解；

f) $\tau_n\in T_s$//τ_n 纳入测试集合 T_s；

g) ErrorFlag=Test(τ_n) //执行测试数据；

h) if ErrorFlag=1 //测试发现软件缺陷；

i) $rate$=Lower($rate$)//降低降温速率并更新；

j) repeat $a\sim j$;//重复步骤 $a\sim j$；

k) while($E(\tau_n)-E(\tau_o)<\sigma$ or $E(\tau_n)==1$) //连续 m 次满足终止条件；

l) return T_s。

2 飞行控制软件自动测试设计与实现

飞行器控制管理软件是飞行器的核心和大脑，随着无人机技术的快速发展，飞行器控制管理软件的设计越来越复杂，模块之间交联众多。传统的飞行控制软件进行配置项测试需要搭建全套的软硬件测试环境。即使与飞控系统交联的设备采用模拟软件代替，飞行控制软件的测试用例依赖于系统状态[26]，需要等待软件运行到特定情况下，观察输出是否正确。测试用例需要经过精心的组合，才可能减少测试的时间和模拟飞行的次数，往往一轮完整的配置项测试需要数个星期的时间，费时费力。

飞行控制软件自动测试的设计与实现过程如图 2 所示，分为以下几个步骤：

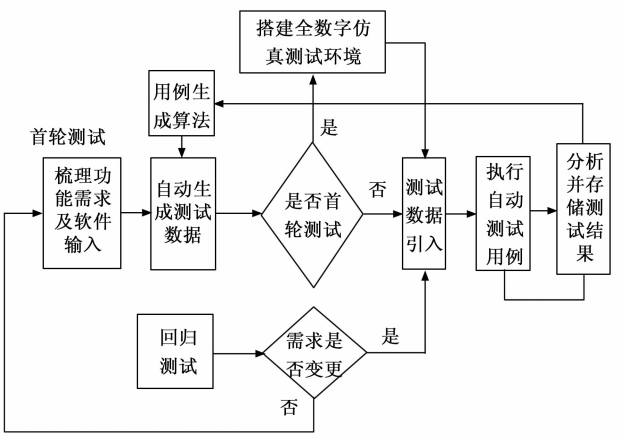

图 2 飞行控制软件自动测试设计与实现流程图

1）梳理功能需求及软件输入。将飞行控制软件的功能分为综检指令处理、遥控指令处理、应急处置管理、航线

管理、控制模式管理、飞行控制管理共 6 大模块。将模块进一步分解为功能点，明确每一个功能点的输入、输出及前提条件，给每个功能点编号。

2）自动生成测试用例。根据每个功能点的输入、输出及参数范围，使用 1.4 节设计的 AETG-SA 算法自动生成测试用例，覆盖正常场景和异常场景，测试用例的输入符合飞行控制软件与交联软件的接口协议。

3）编写自动测试驱动代码。首先搭建自动测试框架。采用代码隔离的思想，自动测试代码和飞行控制软件虽然都运行在全数字仿真环境中，但测试代码与飞行控制软件代码工程独立，减少了代码插桩导致后期装机时的代码冗余和因插桩引起的软件问题。将测试用例数据导入到自动测试框架中，执行测试用例，获取软件的输出结果并断言。

4）导入全数字仿真环境。基于全数字仿真环境搭建飞行控制软件的配置项测试环境，配置芯片和外围器件的参数，将编译后的代码导入到全数字仿真环境中运行。

5）记录并存储测试用例执行结果。将全数字仿真环境中的测试用例执行结果进行统计分析，生成测试报告并导出至 Excel 文件中。

首轮测试结束后，回归测试可以重用首轮测试设计的测试用例并自动执行，测试环境可以复用首轮搭建的全数字仿真环境，仅将修改后的代码加载到测试环境中，大大提升了测试效率。

3　某型无人机飞行控制软件自动测试应用实践

某型无人机飞行控制软件运行在飞控计算机中，实现无人机全过程的飞行控制与飞行管理功能，通过 RS422、CAN 等多种接口与机载系统进行数据交换，实现飞行管理、飞行制导、飞行控制、应急处置、数据处理、余度管理等多种功能。飞行控制软件软件是无人机的控制核心，对无人机的飞行安全至关重要。由于软件的复杂程度较高，软件测试过程耗时较长，为提升测试效率，采用基于全数字仿真环境的自动测试方法。飞行器管理系统与交联设备的连接如图 3 所示。

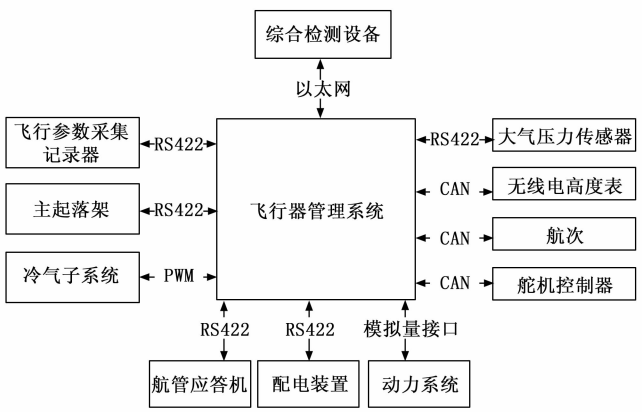

图 3　飞行器管理系统与交联设备连接图

对飞行控制软件进行功能分解，可以分解为多个功能点，测试用例即对多个功能点是否符合设计要求进行验证。以测试起飞滑跑时角速率内环的俯仰角保持控制功能为例，说明自动测试流程。角速率内环的俯仰角保持控制是无人机处于起飞滑跑、起飞拉起、着陆滑跑阶段对俯仰角进行控制的重要功能，测试过程中首先飞控软件判断无人机是否进入了起飞滑跑阶段，进入后生成俯仰角给定、俯仰角、俯仰角速率 3 个参数的测试数据，将测试数据引入，执行自动测试用例，计算升降舵偏度后判断测试是否通过，最后记录并存储测试结果。测试流程如图 4 所示。

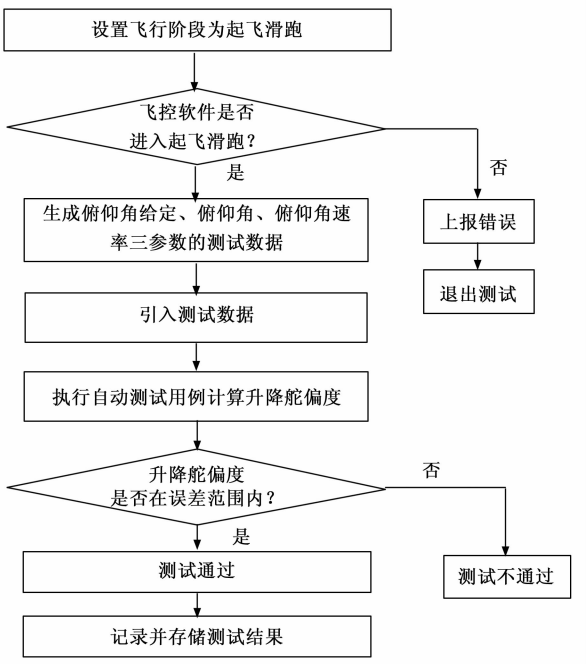

图 4　角速率内环俯仰角保持控制自动测试流程

自动测试伪代码如下：

```
Begin
SetModePitch_Hold_MODE
if FlightMode = Pitch_Hold_MODE
then
Generate Theta_Given,Theta,Theta_q
else
Report Set Pitch_Hold_MODEError
exit PitchHoldTest
Import Theta_Given,Theta,ThetaSpe
if abs() < 0.1(误差范围)
Display("PitchHoldTest Success")
Store PitchHoldTest Result
else
Display("PitchHoldTest Fail")
Store PitchHoldTest Result
End
```

将某型无人机飞行控制软件的功能模块进行梳理，在全数字仿真环境中自动执行测试用例，并将测试结果存储到本地。该软件测试数据对比见表 1。

表 1 某型飞行控制软件测试数据对比

测试环境	设计测试用例/个	执行测试用例/个	未执行测试用例/个	测试用例设计时间/h	测试执行耗时/h	发现软件缺陷/个	回归测试耗时/h
全实物系统手工测试	545	516	29	82.6	68.8	15	16.1
半实物系统手工测试	545	530	15	82.6	70.6	19	18.4
全数字仿真环境自动测试	545	545	0	93.4	0.3	20	0.1

本次测试共搭建 3 种测试环境：全实物系统测试环境、半实物系统测试环境，全数字仿真测试环境。其中全实物系统测试环境和半实物系统测试环境采用的是传统的人工设计并执行测试用例，全数字仿真测试环境中采用的是 1.4 节中的 AETG-SA 算法自动生成测试用例并执行。全实物系统测试环境中所有的设备均为实装设备，包括飞控计算机、大气压力传感器、起落架、动力系统、舵机控制器、配电装置等，搭建全套的测试环境非常费时费力，需要的资源的投入较大。半实物系统测试环境中，飞控计算机为真实设备，其余与飞控计算机交联的设备均使用仿真软件模拟，测试环境如图 5 所示。

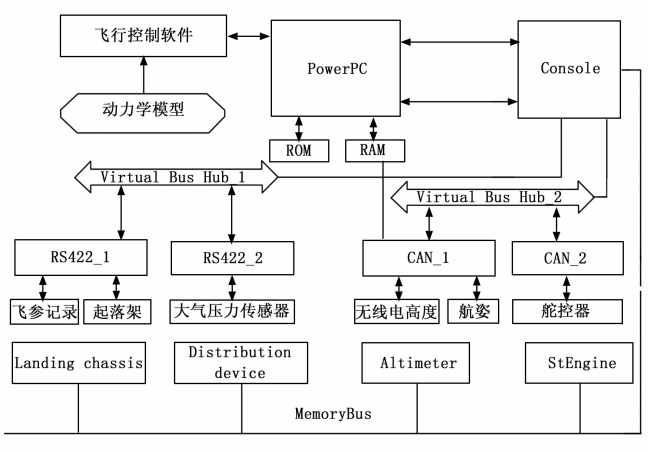

图 5 飞行控制系统全数字仿真环境图

相较于全实物系统测试环境在资源占用和时间花费上都有节约。而全数字仿真测试环境的飞控计算机和交联的测试设备均在仿真平台上搭建，平台构建的底层技术采用了 1.2 节中基于 AADL 的系统架构建模技术、基于 DML 的虚拟硬件建模技术、基于 DDS 的网络通信技术、动态二进制翻译技术。编译完成的飞行控制软件的二进制代码才可以运行到全数字仿真平台上，模拟真实的目标机实现相应的功能，接口类型和数据逻辑与真实环境一致，可以满足软件功能和接口的测试需求。

此次飞行控制软件共设计动态测试用例 545 个，设计相同数量的测试用例 545 个，在全实物系统测试环境中，受系统硬件和环境的影响，如故障构造、传感器数据未在飞行状态下只能产生部分数据等，只能执行 516 个测试用例，29 个测试用例未执行。如测试用例中设计的舵机控制器故障时，测试无人机飞控软件的故障处理，在全实物系统测试环境中，无法在不破坏硬件的情况下构造出此类故障，因此无法执行此测试用例。

半实物系统测试环境由于飞控计算机交联的设备采用仿真软件代替，增加了测试充分性，可执行 530 个用例，共 15 个用例未执行。半实物系统中虽然与飞控计算机交联的设备采用仿真设备代替，但飞控计算机本身的故障测试用例无法执行，如飞控计算机的存储器故障测试用例，一般测试为向飞控计算机存储器中写入测试数据以验证存储器工作是否正常，在半实物系统测试环境中无法执行。

全数字仿真测试环境由于不受系统硬件影响，以上举例的测试用例舵机控制器故障、飞控计算机存储器故障均可构造，设计的测试用例可全部执行，对软件的测试更充分，能更大程度地发现潜在的软件缺陷。

在测试耗时方面，全实物系统测试环境中人工执行 516 个测试用例，耗时 68.8 h，半实物系统测试环境中人工执行 530 个测试用例耗时 70.6 h，全数字仿真测试环境自动执行 545 个测试用例仅需 0.3 h。

分析上述数据不难发现，在全数字仿真环境下，使用 AETG-SA 算法自动生成测试数据，结合自动测试框架自动执行测试用例，测试充分性提升 5.3%。相较于半实物测试环境，全数字仿真环境测试手段执行充分性提升 2.8%。合并首轮测试用例的设计时间和回归测试耗时，相较于在全实物系统中进行测试，自动测试耗时缩短 44%，相较于半实物系统，自动测试耗时缩短 45%，大大提升了测试效率。

4 结束语

本文针对机载飞行控制软件功能测试耗时长、对硬件依赖程度高、自动化程度低、测试数据设计覆盖率低等问题，提出了一种基于全数字仿真环境的机载飞行控制软件自动测试方法，并在工程实践中进行应用。相较于全实物系统测试环境和半实物系统测试环境，全数字仿真测试环境进行功能自动测试极大的缩短了测试耗时，同时具备快速重建能力，提升了测试效率。但目前来说，该测试方法适用于同一型号软件及迭代版本的测试，不同型号软件之间的通用性有待提升。后续逐步对该测试方法进行优化，提高复用性。

参考文献：

[1] 孟小丰，杨娇茹，冉宏艳. 飞行控制软件复杂算法测试验证方法研究 [J]. 质量与可靠性，2022 (3)：44－45.

[2] 一种基于机载嵌入式软件的自动测试方法 [J]. 信息技术与信息化，2022 (2)：17－18.
[3] 邹小花，王　渊. 基于 DDS 的机载嵌入式软件仿真自测试方法 [J]. 计算机仿真，2022，39 (10)：50－51.
[4] 温晓玲，袁维波. 机载软件质量评价方法研究 [J]. 航空标准化与质量，2022，39 (10)：50－51.
[5] 肖思慧，刘　琦，黄澔鸿，等. 机载软件质量评价方法研究 [J]. 软件学报，2022，33 (8)：2851－2852.
[6] 任　然. 军用机载平台电子信息系统的发展趋势 [J]. 电讯技术，2017，57 (4)：486－487.
[7] 姜　磊，张天宏. 基于全数字仿真的航空机载软件验证技术研究 [J]. 航空制造技术，2014，6 (16)：109－110.
[8] 周　平，苏银科，沈　超. 基于 DDS 的分布式数字仿真系统设计与实现 [J]. 系统仿真学报，2014，26 (8)：1679－1680.
[9] ASAD K，MA W Q，CHRIS W. Multi-threaded simics systemC virtual platform [C] //Proceedings of 2015 IEEE/ACM International Conference on Computer-Aided Design (ICCAD). Austin：IEEE，2015：373－379.
[10] 肖前远. 航空嵌入式软件全数字仿真测试技术研究 [D]. 南京：南京航空航天大学，2010.
[11] 李铁颖，王科翔，戴苏榕. 一种基于 AADL 的航空电子系统仿真和验证技术 [J]. 信息技术与信息化，2019，50 (4)：23－24.
[12] 陆　寅，秦树东，郭　鹏，等. 软硬件综合 AADL 可靠性建模及分析方法 [J]. 软件学报，2022，33 (8)：2997－2998.
[13] 任棕诜，任雄伟. DDS 在分布式仿真中的应用探讨 [J]. 舰船电子工程，2015，35 (11)：106－108.
[14] 郝玲玲. 基于 DDS 的空战模拟仿真系统的设计与实现 [D]. 西安：西安工业大学，2020.
[15] 吴　晋. 动态二进制翻译系统优化技术与性能回归测试研究 [D]. 哈尔滨：哈尔滨工业大学，2022.
[16] 欧阳慈俨. 基于龙芯体系结构的二进制翻译系统环境优化 [D]. 北京：中国科学院大学，2022.
[17] 韦涌泉，张红军，董振辉，等. 基于动态二进制翻译技术的数管软件虚拟测试环境设计 [J]. 计算机测量与控制，2018，26 (4)：66－67.
[18] 王子豪. 基于需求模型的航天软件测试自动方法研究 [D]. 北京：中国运载火箭技术研究院，2019.
[19] 邱晓晗. 基于模型的飞控机载软件测试用例自动生成技术研究 [D]. 成都：电子科技大学，2019.
[20] 郑燕妮，李志蜀，李　奇. 蚁群模拟退火算法在测试用例约简中的应用 [J]. 计算机工程，2009，35 (2)：197－199.
[21] 仲晓敏，巫晓琳，赵雪峰. 蚁群模拟退火算法在测试用例约简中的应用 [J]. 计算机工程，2010，27 (12)：4544－4547.
[22] 傅　博. 基于模拟退火遗传算法的软件测试数据自动生成 [J]. 计算机工程与应用，2005，41 (12)：82－84.
[23] 韩　露，史贤俊，林　云，等. 测试不可靠条件下基于精华蚂蚁系统的诊断策略优化方法 [J]. 电子测量与仪器学报，2021，35 (3)：130－131.
[24] 王　博，王曙燕. 一种新的基于模拟退火的测试用例生成与约简算法 [J]. 计算机应用与软件，2013，30 (2)：78－79.
[25] 周　虎. 某自动飞控测试系统的设计与实现 [D]. 成都：电子科技大学，2014.
[26] 田　峰，丁文锐，李红光，等. 一种无人机飞行控制管理软件回归测试方法 [J]. 北京航空航天大学学报，2011 (5)：574－575.

(上接第 35 页)

[9] 童紫平，龙善丽，张　慧，等. 单片集成 MEMS 陀螺数字闭环接口 ASIC 电路设计 [J]. 中国惯性技术学报，2020，28 (4)：505－509.
[10] 郑天涌，裘安萍，夏国明，等. 硅微陀螺仪低功耗数字相敏解调 ASIC 设计 [J]. 半导体技术，2022，47 (3)：222－230.
[11] 王亚林，杨拥军，任　臣. MEMS 陀螺仪驱动接口 ASIC 设计 [J]. 微电子学，2019，49 (4)：529－533.
[12] CUI J，YAN G，ZHAO Q. Enhanced temperature stability of scale factor in MEMS gyroscope based on multi parametersfusioncompensation method [J]. Measurement，2019，148：1－12.
[13] 王晓雷，杨　成，李宏生. 硅微陀螺仪正交误差校正系统的分析与设计 [J]. 中国惯性技术学报，2013 (6)：822－827.
[14] 刘玉县，汤　丽，陈　婷，等. MEMS 陀螺仪驱动谐振频率的温度补偿方法 [J]. 传感器与微系统，2019，38 (5)：52－54.
[15] 卢新艳，徐淑静，任　臣，等. MEMS 陀螺仪零偏温度特性分析及性能优化 [J]. 微纳电子技术，2018，55 (8)：588－592.
[16] 钱　磊，赵鹤鸣，张晓峰，等. 基于参数激励的 MEMS 环式陀螺驱动方法与实现 [J]. 计算机测量与控制，2018，26 (11)：293－296.
[17] JIA J，DING B，DING X，et al. In-run self-calibration of scale factor temperature drifts for MEMS gyroscope [C] // 2019 IEEE Sensors，IEEE，2019.
[18] ALQARNI S A，OBEID A M，BENSALEH M S，et al. A fully integrated CMOS interface ASIC for two-axis piezoelectric angular rate MEMS inertial sensors [C] // Sensors，IEEE，2016.
[19] 陆铭洋，宣　琳，刘　靖，等. 硅微轴对称陀螺速率积分模式下角速率输出方法研究 [J]. 传感技术学报，2022，35 (6)：745－750.
[20] 刘　楠，苏　言，童　鑫，等. 硅微机械陀螺仪温度补偿方法的研究现状 [J]. 微纳电子技术，2017，54 (6)：395－400.
[21] 刘春雅，冯向莉. 基于 CORDIC 算法的数控振荡器及其 FPGA 实现 [J]. 自动化与仪器仪表，2015 (8)：131－133.
[22] 谭会生，徐界铭，张驾祥. BP 神经网络 FPGA 实现结构的优化设计 [J]. 计算机工程与应用，2022，58 (21)：264－271.