

飞行控制软件测试中插桩技术的优化方法

郭君红, 李跃飞, 白成刚, 蔡开元

(北京航空航天大学自动化科学与电气工程学院, 北京 100191)

摘要: 飞行控制软件主要采用 Host-Target 的仿真测试模式, 并基于插桩技术实现覆盖测试。针对采用传统的程序插桩技术, 往往会大量增加程序运行时间, 降低程序的实时性甚至导致软件失效的问题, 通过分析插桩在程序仿真测试中对程序各阶段执行时间和程序实时性的影响, 提出一种优化桩信息传输过程的插桩测试方法。实验结果表明了该方法的有效性。

关键词: 飞行控制软件; 覆盖测试; 插桩

Optimal Method of Instrumentation Technique in Flight-control Software Test

GUO Jun-hong, LI Yue-fei, BAI Cheng-gang, CAI Kai-yuan

(School of Automation Science and Electrical Engineering, Beijing University of Aeronautics and Astronautics, Beijing 100191)

【Abstract】 Flight-control software adopts Host-Target simulation test mode and achieves coverage test based on instrumentation. Traditional program instrumentation can increase the executing time greatly and influence the real-time performance then even cause software failure. This paper analyzes the influence of instrumentation in different process of the simulation software test through some experiments. And an optimal method of software test process to the probe information transmission is presented. Experimental results show this method is effective and feasible.

【Key words】 flight-control software; coverage test; instrumentation

1 概述

飞行控制软件是飞行控制系统中最重要的组成部分, 要求可靠性极高。这需要使用白盒测试方法对软件进行完整详细的测试, 主要是通过插桩对程序进行动态覆盖测试以获得程序的覆盖率、运行剖面等信息。但因为软件的任务运行周期严格限制在 10 ms, 采用传统的软件测试插桩技术对嵌入式飞行控制软件进行结构覆盖测试, 会影响甚至破坏被测软件的时间特性、时序关系, 可能造成任务超时, 导致软件执行错误, 甚至不能运行^[1]。所以对嵌入式软件进行插桩的时候要采取灵活的插桩方案, 降低插桩对软件运行的影响^[2]。因此, 研究如何对插桩在覆盖测试中的应用进行优化, 找出在飞控软件测试时适用的插桩方法, 对嵌入式软件测试具有十分重要的意义。

2 飞行控制软件测试策略

目前对飞行控制软件主要采用的是 Host-Target 的仿真测试模式。Host-Target 是指编译链接和测试分析都运行在宿主主机上, 而被测程序运行在目标机上, 宿主机与目标机之间的通信通过串口线或是网络连接实现。其中, 主机(Host)上包括主控平台和信号仿真平台, 目标机(Target)上则是飞行控制软件的运行平台。仿真测试的构成示意图如图 1 所示。

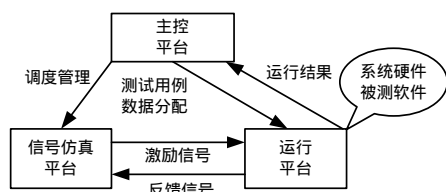


图 1 仿真测试的构成示意图

主控平台主要进行测试用例生成、测试运行管理、数据分配以及测试后的评估工作; 运行平台就是嵌入式系统的硬件及被测软件; 信号仿真平台则是模拟飞行控制系统外部运行环境, 向飞行控制系统提供激励信号并接收反馈信号。

3 插桩对仿真测试的影响

简单地说插桩就是借助向被测程序中插入操作来实现测试目的的方法。常常要在程序中插入一些桩函数, 其目的在于执行程序时, 获得测试者最为关心的信息。通过这些信息进一步了解程序执行过程中的一些动态特性^[3]。在整个过程中, 插桩的过程是静态的, 而数据的收集的过程是动态的^[4]。

在对飞行控制软件测试时, 需要对飞行控制软件的源程序进行程序插桩, 这必然会对程序的运行造成一定的影响。首先, 有些桩程序的信息对于软件测试是不必要的, 插桩程序的运行本身有一定的时间消耗, 再在被测软件中插入目标软件所不需要的桩程序会造成代码膨胀, 增大了执行程序的规模; 其次, 插桩程序捕获的数据信息要存入缓存区, 对系统的内存也有一定的消耗, 也会影响到系统的性能; 最后, 在仿真测试时, 为了降低测试时对飞行控制软件源程序运行效率的影响, 诸如覆盖率分析、路径跟踪等较为复杂的运算都是在主控计算机上完成。这些分析数据均来源于插桩程序收集的数据, 都需要通过通信传送到宿主主机上进行分析处理。

基金项目: 国家自然科学基金资助项目(60633010)

作者简介: 郭君红(1974 -), 女, 副教授、博士, 主研方向: 软件测试, 软件可靠性; 李跃飞, 硕士研究生; 白成刚, 副教授; 蔡开元, 长江学者、教授、博士生导师

收稿日期: 2009-09-20 **E-mail:** guojunhong@buaa.edu.cn

这些因素直接导致了程序运行时间的增长,而实时性是嵌入式软件最重要的特征^[5]。对嵌入式软件进行插桩测试很可能导致软件的失效,这样的测试就是失败的。

要得到一种对源程序影响较小的插桩方法,首先要分析造成这些影响的原因。当程序比较大而且桩点比较多的时候,由于通信量增加导致整个仿真测试时间的增长是明显的。下面将通过实验来验证这个现象。由于在 VxWorks 操作系统下获取时间的精度不是很高,因此以 2 台 Windows 操作系统的计算机分别作为目标机和宿主机来完成测试过程。首先,在主控程序的开始和结束的位置记录整个过程的运行时间 WholeTime;同时分别在主机信号模拟平台实现飞行状态模拟的全量方程程序以及目标机的控制律程序的控制指令。在飞行状态信息传递的位置记录时间分别得到飞行控制程序和全量方程的运行时间 TargetTime 和 HostTime;则通信时间 CommunicationTime = WholeTime - TargetTime - HostTime。

试验对象分别是经过插桩和没有插桩的飞行控制软件,选择 3 个不同的测试用例做 3 组实验,每组重复进行 20 次。对每组实验数据进行平均值计算,得到仿真测试各阶段执行时间对比数据,如图 2 所示。

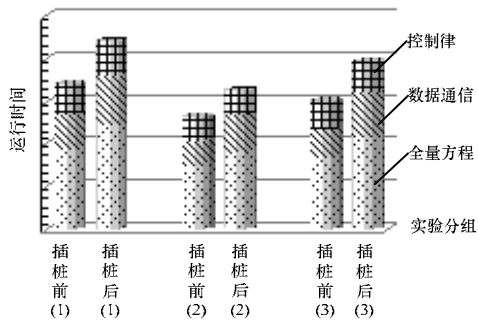


图 2 插桩前后运行时间对比

由图 2 可以看出程序插桩后进行仿真测试过程中,主机和目标机之间的通信时间增加的比例较大。所以为了提高测试的执行效率,有必要对仿真动态测试时的桩信息通信进行优化。

4 一种飞行控制软件测试中插桩通信的优化方法

从上面的实验分析可知,桩信息通信导致的测试时间增加是显然的。所以对此过程进行优化可以缩短测试的时间,提高测试的效率。通信模块中增加的信息量要适当,太大的话将导致测试时间的增长,太小的话则可能导致有效桩信息的丢失,影响覆盖率的计算。传统的方法是在每次目标机与主机交互的时候都把桩信息一并传输出去,当程序规模比较大、桩点较多的时候,导致了通信量的增加会很明显。而传送出去的桩信息中有些部分是无效的。针对这个特点,可以改变成每次传送的数据不是所有的桩信息,只是一部分有效的桩信息。优化流程如图 3 所示。

具体的优化方法如下:(1)声明一个桩点信息数组 $P[n]$ 以及一个记录桩点序列的数组 $S[n]$ (维数相同),并对序列数组的标志位进行初始化 $flag = 0$ 。(2)声明一个维数比较适当的用来传送桩信息的数组 $T[m]$ 。(3)在桩点处插入 $P[i]$ 的赋值函数, $P[i]=true$ 。(4)从 $S[i]$ 的标志位 $flag$ 之后顺序选取前 m 个相应 $P[i]$ 的元素为 true 的序列号作为被传送数据的对象赋给 $T[m]$ 。(5)如果 f 标志位之后所有满足的 $P[i]$ 值为 true 的数量 $j < m$,则从 f 标志位前补充作为 $T[i]$ 的元素进行赋值。(6)对序列数组 $S[i]$ 进行更新,将被传送出的数组的序号元素与标志位后

最近的没有被选择传输的元素进行交换,并修改标志位 $flag$ (f 之前的元素均是传送过的桩信息数组的序号),此步骤的程序添加在通信模块后面,实现在通信的同时执行,节省时间。(7)同样在宿主机上也声明这样的 3 个数组,并通过相反的算法解码得到桩点的信息进行实时覆盖率的计算。(8)如果程序没有结束则转到(4)。(9)仿真测试结束,得到最终覆盖率图。

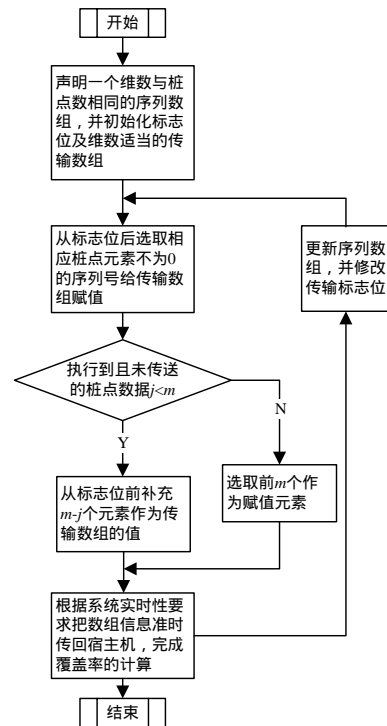


图 3 优化流程

5 实验对比

5.1 实验内容

以某型飞机的控制软件为测试对象。在某型飞机的飞行控制软件仿真过程中,控制律程序和全量方程的计算分别运行在目标机的 VxWorks 操作系统和宿主机的 Windows 操作系统中。它们之间的相互联系在于控制律程序。根据宿主机传来的状态信息产生控制指令,然后把控制指令传送到宿主机,最后经全量方程的计算得出飞行过程中的状态信息。该系统的一个周期为 50 ms。由这个过程可以看出,该系统是一个交互性很强的系统。实验过程中目标机与宿主机之间的通信采用网络套接字的方式实现。实验平台如图 4 所示。

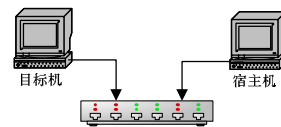


图 4 仿真实验平台

首先,采用本文所述的优化方法对插桩后的通信模块进行修改;然后,将修改后的控制律程序重新编译并下载到 VxWorks 操作系统的目标机中运行。主机控制界面将得到覆盖率的实时变化曲线以及各模块的执行次数等动态数据。

5.2 实验结果

对同一飞行控制软件进行 3 组实验,每组都使用相同的测试用例进行了 20 次测试,并统计程序运行时间进行对比。第 1 组运行的是没有经过插桩的源程序;第 2 组是使用传统插桩方法进行测试的程序;最后一组是按照前一部分内容对

(下转第 24 页)