

基于处理器仿真的发动机控制软件平台

陶金伟, 车 鑫, 陈海泉

(中国航发商用航空发动机有限责任公司, 上海 200241)

摘要: 为提升商用航空发动机控制软件的效率、节省验证成本, 数字仿真技术被广泛采用, 在仿真平台保证功能逻辑正确的前提下, 再进行电子控制器硬件 (EEC) 的集成; 商用航空发动机控制软件不但要满足发动机的功能和性能要求, 还需要满足适航安全的目标, 由于仿真平台与真实硬件平台的差异, 在仿真平台获得的验证结果往往不被适航所采纳, 因此提高仿真度, 使得仿真平台能作为控制软件开发平台被适航所认可, 是进一步节省成本的有效方法; 基于通用仿真运行平台 SIMICS, 通过采用模拟硬件处理器的寄存器、指令集以及存储器、外设等, 使得在目标平台上运行的代码可以直接运行在仿真平台上, 在仿真平台达到目标码级别的验证; 通过软件时间行为的分析与模拟, 使得系统和硬件的仿真程度进一步提高, 更接近真实硬件, 在仿真平台所获取的验证数据, 更容易被适航所采纳。

关键词: 发动机; 控制软件; EEC; 仿真

Engine Control Software Development Platform Based Processor Simulation

Tao Jinwei, Che Xin, Chen Haiquan

(AECC Commercial Aircraft Engine Co., Ltd., Shanghai 200241, China)

Abstract: To improve the efficiency of the commercial aircraft engine control software development, simulation is extensively used, in simulation platform, it usually used to prove the logic is correct and then integrated the application software into the EEC hardware. But for commercial aircraft engine control software, the development objective is not only to meet the functional and performance requirements, but also to meet the airworthiness requirements, because there are differences between simulation platform and target hardware, the test result obtained in simulation platform can't be accepted by the certification authority. So minimal the differences between simulation and target hardware is important. Build a platform based SIMICS, and use the processor simulation technology, simulate the processor instructions, memory, devices, etc. The application executable object code can run on the simulation platform as same as the target, and by the way of analyze the behavior of system, to improve on the integrity of the test results in the simulation platform.

Keywords: digital watermark; wavelet packet; adaptive; feature of texture

0 引言

仿真技术由于其天生的优势, 在数控系统的研制过程中, 已成为重要工具和手段, 由于实际发动机数控试车投入高、风险大, 因此在数控系统装机之前, 需进行一系列仿真试验, 其中主要包括控制系统数字仿真、硬件在回路仿真、半物理仿真等环节, 各个环节相辅相成, 但功能、仿真回路、仿真目的不尽相同^[2]。

构建发动机控制系统全数字仿真平台不仅可以在发动机研制的初期起到控制系统需求确认的作用, 也可以在研制后期作为验证平台对需求进行验证; 仿真平台主要目的是在研制前期尽可能多的发现软件故障, 由于软件故障主要是在设计阶段产生的一种软件缺陷而造成的, 是在系统运行之前就已经存在的^[1], 因此利用计算机仿真验证平台可以很快的发现并修正软件错误, 对比于实物验证平台, 仿真平台可以并行验证, 通常情况下能有效的提升效率。

控制软件作为发动机控制逻辑和故障检测和保护逻辑

的实现载体, 通常在整体发动机研制的后期才能得到充分的验证, 且常常受制于硬件研制的进度滞后以及硬件研制数量限制的影响, 而导致控制软件的研制进度落后于其他部件。因此, 在发动机性能、部件和附件等仿真模型已具备的前提下, 建立发动机全数字仿真平台是解决上述问题的有效方法。而控制软件不受环境影响的特点, 使得其具备了在仿真环境下就可以完成验证的可能。

1 仿真平台的架构及原理

目前, 商用的大涵道比涡扇发动机一般采用全权限数字电子控制 (FADEC) 技术。电子控制器 (electronic engine control, EEC) 是整个控制系统的核心, 具有强大的数字计算能力和数据处理能力, 从而能完成更加先进、复杂的控制^[3]。EEC 接收来自飞机的指令, 采集发动机温度、压力、转速以及作动部件位置等参数, 实现发动机在所有使用条件下的稳态、过渡态、起动、点火、反推的控制功能和限制保护 (包括参数限制和极限保护) 功能, 保证发动机在飞行包线内不出现超温、超转、超压、失速及喘振等现象, 允许飞行员无约束操作而发动机不出现异常现象和破坏性故障。因此, 在构建控制系统仿真平台时, 需要集成如飞机模型、发动机模型、燃油系统模型、传感器模

收稿日期: 2020-05-22; 修回日期: 2020-05-28。

作者简介: 陶金伟 (1985-), 男, 江苏南京人, 硕士, 主要从事航空发动机控制系统方向的研究。

型、电子控制器硬件模型等仿真模块，在此基础上，集成操作系统构建控制软件的开发平台。具体的仿真平台架构如图 1 所示。

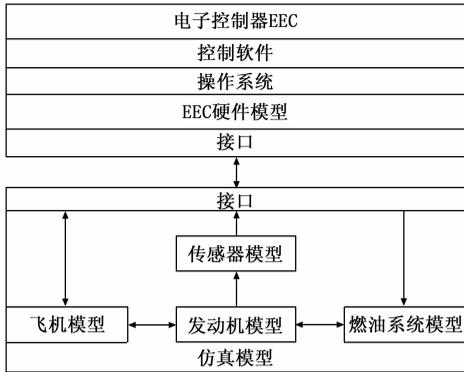


图 1 发动机全数字仿真架构

发动机控制软件开发平台构建的关键因素如下：

- 1) EEC 仿真，即仿真硬件设计，EEC 的仿真是构建闭环仿真平台的关键，模拟电子硬件，控制软件可以不经修改直接运行，使得控制软件是整个仿真系统中唯一的一个真件。
- 2) 仿真系统集成，即软件设计，调度各仿真部件和控制软件以一定的周期和时序运行，以模拟真实情况下的并行运行。

2 硬件设计

EEC 仿真主要基于风河公司的 SIMICS 平台，在此基础上开发处理器模型，存储器模型，外设模型等构建硬件仿真，同时在该硬件上模拟操作系统，共同构成 EEC 仿真平台。

2.1 电子控制器 (EEC)

考虑到系统的可靠性和安全性设计，电子控制器 EEC 一般采用冗余设计的模式，由功能完全相同的双通道构成，双控制通道分为 A、B 两个通道，其中一个作为主控通道，另一个作为热备份通道，当主控通道发生故障时，可以进行平稳切换至另一通道。EEC 的两个通道间通过数据总线进行数据交换，当主控通道的部分输入信号发生故障时，即可通过数据总线获取备份通道的数据，借用备份通道的输入信号进行系统重构。

2.2 计算机硬件模拟器 (SIMICS)

Simics 是一款高性能的计算机硬件模拟器，可以通过模拟不同硬件构建不同架构的计算机系统。Simics 可以实现指令级别、处理器内核处理器级别、硬件电路板级别等不同级别的模拟，同时还具有较高的性能^[4]。Simics 为开发者提供了一个通用的框架，基于该框架集成处理器、存储器、外设等构成硬件仿真，同时还可以支持固件、驱动程序和操作系统的开发等功能^[5]。

Simics 的主要功能主要有如下几方面：

- 1) 提供目标指令集运行的平台，Simics 平台提供驱动软件目标指令集运行的能力，使得用户可以在 PC 环境下，运行真实的软件目标程序；

- 2) 处理器仿真，Simics 能仿真诸如 ARM、PowerPc、X86 等多种架构的 CPU^[6]，并且可以在这些仿真硬件上运行多种实时嵌入式操作系统，包括 VxWorks、Linux 等；

- 3) 存储器和外设的仿真，利用 Simics 提供的语言对硬件设备进行建模，建模的对象包括处理器、存储器、外设、网络通信等多种设备。具体 SIMICS 的架构及开发如图 2 所示。

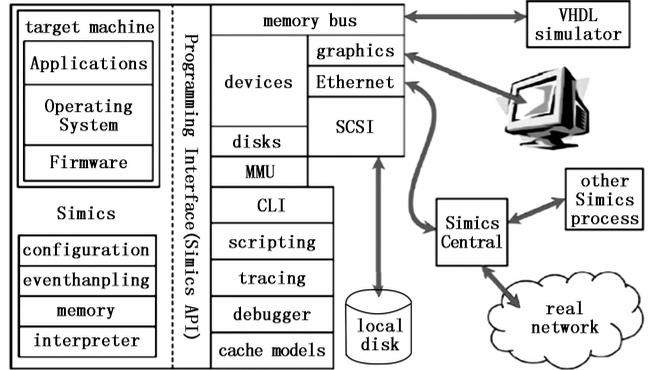


图 2 SIMICS 架构^[7]

综上，Simics 主要用来虚拟特定的嵌入式硬件平台来开发相应的软件^[8]。

2.3 EEC 仿真架构

发动机控制软件从大的方面可分为应用软件 (AS)、操作软件 (OS) 以及操作软件与应用软件的接口 (AS \ OS 接口)，操作软件直接与硬件互联，其功能主要是给应用软件提供运行环境，如任务调度，通信等，其行为主要包括调度的策略，异常的处理等，不具有主动行为，只要将其提供服务的能力限制在一定的范围内，不对整个系统的行为造成影响。因此，OS 仍以仿真的形式提供。具体 EEC 软硬件仿真架构如图 3 所示。



图 3 EEC 软硬件仿真架构

在该仿真架构下，由于操作系统为仿真件，因此存储器、IO、引导、驱动等的仿真不影响系统整体的仿真性能，所以，而处理器作为应用软件运行的载体，其仿真程度直接影响应用软件的行为，处理器仿真是该平台实现的关键。

2.3.1 处理器仿真

基于仿真平台的总体架构，该平台并未针对驱动等的验证，主要面向于为应用软件的运行提供仿真平台，因此，在处理器的仿真上，无需完全仿真真实的处理器，而只需要仿真其内核即可，因此选取 PowerPC 处理器系列下的 e500 内核，基于该内核，模拟器寄存器的模型，指令集等，使得通过商用的编译器编译得到的目标码能直接在仿真的

处理器上运行。

按照 PowerPC 体系下的 e500 内核实现寄存器模型,如通用寄存器,浮点寄存器,处理器控制寄存器,定时、中断寄存器,SPRs、BTB、Cache 控制、MMU、调试寄存器,SPE 和 SPE APU 等。

按照 PowerPC 体系下的 e500 内核实现指令集仿真,针对 e500 核的处理器架构,在 X86 计算机上,采用高级语言开发实现 PowerPC 指令^[9],主要涉及如整数运算指令,浮点指令,存取操作指令,流程控制指令,处理器控制指令,同步指令,信号处理指令等的仿真。

由于是在 X86 平台下的 PC 机中进行仿真,因此在执行指令时,需要通过解释执行或者动态翻译的方法将 PowerPC 的指令转换为 X86 的指令;ISS 的基本仿真策略分为解释型和编译型两种^[10],解释型指令集仿真策略,为 CPU 指令系统中每一条指令编写对应的函数来解释执行,执行这两种方法在运行效率上均不高,因此,在实际运行时,采用编译型策略加上一定的缓存技术,以提高指令运行的效率。

2.3.2 与真实硬件的差异

在寄存器与指令集本身的模拟上,除了部分特殊的 SPE 指令和 Cache 相关指令未实现外,其余均于真实处理器一致。

在指令的处理方面,由于真实的处理器存在流水线等架构,而仿真时并未考虑,如指令预取、译码、执行、写回等。

以上差异均与软件运行的效率,即运行时间相关,而在仿真运行时,由于并不实时,更多的考虑是在正常情况下完成功能,同时能通过故障注入的方式完成软件错误处理逻辑的验证;因此在软件设计时,可通过时间分区的调度设计、WCET 分析和验证、故障模式分析及超时故障策略设计与验证等消除仿真平台与真实平台差异带来的影响,使得在仿真环境与真实处理器环境下,软件的行为(如调度顺序、通信方式)等,是一致的。

3 软件设计

系统软件集成的主要目的是驱动各个模型以特定的频率运转起来,并相互交联完成系统的功能,因此在构建数字仿真平台时,软件集成工作主要关注点如下:

- 1) 模型以何种方式交联?
- 2) 模型以何种频率运行、运行的顺序如何、模块之间如何协同工作?

就以上两个问题,分别以软件架构以及同步设计两个方面进行介绍。

3.1 软件架构

总线结构的出现大大促进了计算机的发展。总线结构使计算机能够方便的组件化,设备接入总线,从总线上获取设备所需要的输入,通过总线发送处理后的输出。采用基于总线的思想,将总线结构引入软件中,协同仿真总线提供了基本的数据读写操作,从而满足各个模块之间的互联互通。

通过用例模拟飞行包线,驱动飞机模型和发动机模型,

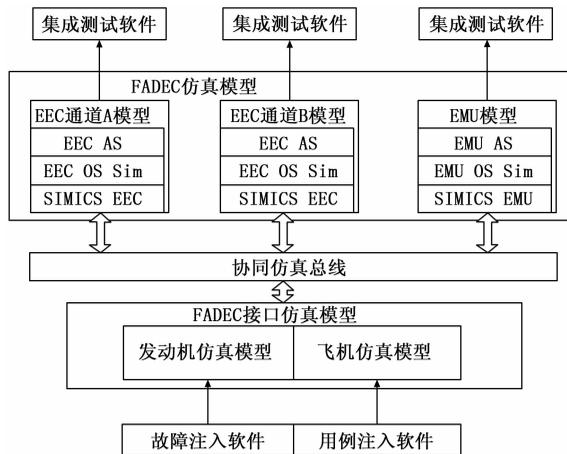


图4 软件架构图

控制发动机进入各个状态,从而达到发动机仿真的目的,同时,可以通过故障注入,模拟发动机不同飞行包线、不同状态下的故障,从而完整的验证 FADEC 的控制逻辑等。配合集成测试软件完成部分相关的验证活动,如测试覆盖率获取等。

3.2 同步设计

仿真平台区别于真实平台,其时间特性是相对的,因此在仿真频率的选取上,将发动机对于控制的最大频率的要求进行归一化,通过量化的频率作为分辨率进行控制时序的设计,因此,在运行效率上,仿真的效率是应该高于真实硬件的。

同步运行是仿真平台运行正常的关键,如果各个模块之间运行不同步,将造成各模型之间信号的传递出现延迟,从而造成整个系统功能丧失。最简便的同步方法是将所有的模型通过软件的方式集成编译至一个镜像内运行,但由于发动机模型等仿真模型集成方法的各不相同,该方法并不适用。只能采用类似分布式仿真的方法,各个模型独立运行,通过通信的方法进行同步。

考虑到如上两方面,本文设计的仿真运行环境主要有如下设计:

- 1) 单个模型异步运行,运行的时钟由协同仿真总线提供。
- 2) 协同仿真总线提供统一的仿真运行时钟,时钟的频率为系统控制要求频率的最大公倍数。该时钟为非周期时钟,需要等到所有的模型在单个仿真步长内完成所有工作后,才开始下一个仿真周期。
- 3) 对于多任务的模型,由模型内部自行调度,如 EEC 仿真模型,其运行时序由 OS 进行调度,其调度顺序应与目标平台一致。

如图 5 所示,由于并未采用绝对时间仿真,因此每一个仿真周期不相等,即达到了仿真同步的效果,又不影响模块的逻辑顺序,缺点周期内任务肯定能运行完成,所以无法模拟在单个周期内任务运行超时的故障,但是该故障可以通过其他方法模拟,如直接注入等。

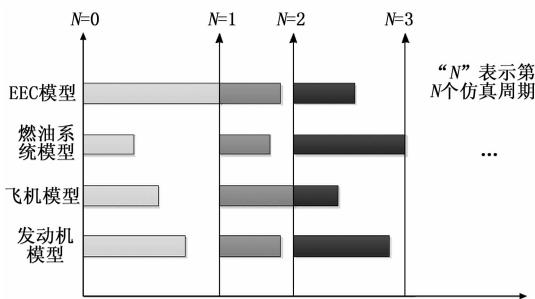


图 5 同步调度图

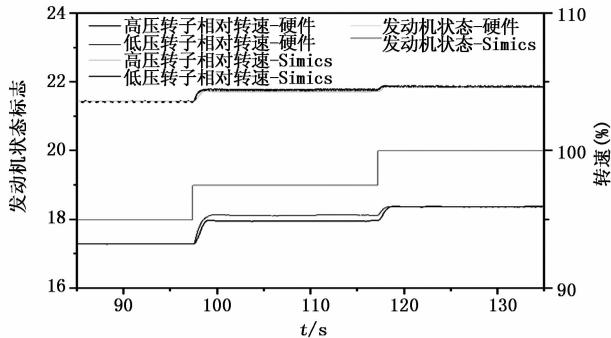


图 6 测试结果及对比

4 实验结果与分析

4.1 实验用例概述

为验证仿真平台的正确性，设计如下的用例：上电运行，待所有模块同步运行后，通过用例注入一个飞行循环，验证控制发动机在各个状态下的功能，具体如表 1 所示。

表 1 用例说明

预期执行发动机状态	执行时间
待机	0~4 s
起动至地面慢车	4~60 s
巡航	60~80 s
最大巡航	80~100 s
爬升	100~120 s
最大爬升	120~140 s
连续	140~160 s
最大连续	160~180 s
起飞	180~200 s
最大起飞	200~220 s
最大起飞至最大推力	220~240 s
最大推力	240~280 s
巡航	280~300 s
地面慢车	300~320 s
停车回待机	320 s 后

测试执行方法及预期输出：在基于 SIMICS 的仿真平台、真实硬件平台中，分别运行上述用例，得到结果，并对比两种平台运行的结果是否正确且一致。

4.2 验证结果

在 SIMICS 平台执行用例，将结果与真实硬件平台的结果进行对比，对比发动机状态从巡航状态→最大巡航状态→爬升状态转换的过程数据，如图 6 所示。

可以观察到：

1) 对 SIMICS 平台的仿真结果分析，输出的发动机状态切换符合预期期望，高压转子转速和低压转子转速变化趋势满足发动机的控制要求，SIMICS 仿真平台上用例运行结果正确。

2) 除在信号精度等方面存在差异外，在 SIMICS 平台和真实硬件平台上执行结果与 SIMICS 仿真平台执行结果基本保持一致。

5 结束语

基于以上的论述，以处理器仿真为核心，构建发动机控制软件与发动机模型的闭环仿真系统，能有效的进行控制系统控制逻辑的验证，同时减小 PC 机环境与真实目标环境的差异，可以作为发动机研制过程中，控制软件的开发与验证平台。

在验证方面，对仿真平台的正常功能进行了验证，但对于异常处理，如各类系统故障、硬件故障、随机瞬时故障（如由电磁干扰、粒子辐射等原因引起的短暂功能失效，并且是一次有效性^[11]）等，未进行详细的论述。以该平台为基础，模拟硬件和系统故障注入，发现更多的软件故障将是下一步的工作。

参考文献：

- [1] 王 强, 傅 强, 等. 航空发动机 FADEC 系统软件故障分析 [J]. 微计算机信息, 2005, 21 (12) 141-142.
- [2] 姚华廷, 王 曦, 苏三买, 等. 航空发动机数控系统综合仿真平台设计 [J]. 航空动力学报, 2010, 25 (9) 2148-2154.
- [3] 殷 锴, 陶金伟, 王鸿钧, 等. 民用航空发动机控制系统回路设计与仿真航空计算技术 [J]. 2012, 42 (6) 111-114.
- [4] 胡 倩, 王 超, 王海霞, 等. 基于 Simics 的系统级故障注入平台 [J]. 计算机工程, 2015, 41 (2) 63-68.
- [5] 韩德强, 冯云贺, 王宗侠, 等. 基于 Simics 的嵌入式系统的研究与开发 [J]. 电子技术应用, 2015, 41 (2) 31-34.
- [6] Ozcan Ozturk, Multicore Education Through Simulation [J]. IEEE Transactions on Education [J]. 2011, 54 (2) 119-126.
- [7] Wind River. Virtutech Simics, Alameda, CA, Last accessed [Z]. 2010.
- [8] 赵君建, 张盛兵, 黄小平, 等. 一种基于虚拟机的处理器高效原型验证方法 [J]. 微电子学与计算机, 2015, 27 (7) 100-102.
- [9] 付 琳, 胡 锦, 梁利平, 等. 指令集仿真器的关键技术信息科技计算机硬件技术 [J]. 2015, 35 (5) 1421-1425.
- [10] Wang Y, Wang S. Optimization design and realization of IA instruction set simulator [J]. Radio Engineering 2008, 38 (11) 49-51.
- [11] Gil D, Gracia J, Baraza J C, et al. A Study of the Effects of Transient Fault Injection into the VHDL Model of a Fault-tolerant Microcomputer System [C]. Proceedings of the 6th IEEE International On-line Testing Workshop. Palma de Mallorca, Spain; IEEE Press, 2000: 73-79.