

# 基于 Simulink 和 C++ 混合编程的 测试系统建模技术研究

戴 阳, 杨向忠

(北京青云航空仪表有限公司, 北京 100086)

**摘要:** 介绍了一种基于 Simulink 和 C++ 混合编程技术的自动飞行控制系统测试系统建模技术及调试方法; 基于 Simulink 搭建飞行仿真模型, 通过 RTW 自动代码生成工具将各仿真模块分别生成嵌入式代码, 并集成应用于 C++ 软件设计环境中; 通过对时钟的有效设定, 实现了基于 Windows 系统的实时飞行仿真测试系统的设计, 仿真软件的最小运算周期为 2 ms; 该技术不仅可以满足自动飞行控制系统飞行仿真试验的实时性要求, 并且在软件开放性、接口扩展性、板卡驱动的通用性以及软件设计功能多样性等方面占有较大优势; 基于这个技术建立而成的仿真测试系统, 某型自动飞行控制计算机已经成功的开展了多轮半物理仿真验证试验, 并取得了良好的试验效果, 从而为产品的试飞定型奠定了坚实的基础。

**关键词:** 飞行仿真; 测试系统; Simulink; C++; 混合编程

## Research on Testing System Modeling Based on Simulink and C++ Mixed Programming

Dai Yang, Yang Xiangzhong

(AVIC Beijing Keeven Aviation Instrument Co., LTD, Beijing 100086, China)

**Abstract:** This paper introduces a solution for designing a real-time flight testing system based on Simulink and C++ mixed programming. The modules of aircraft are designed and tested in Simulink, and generated into C code separately by RTW toolbox, then applied into C++ project in Windows system, which is reasonably using the Windows timer with a minimum sample time at 2ms. This flight testing system can not only satisfy the real time require of the flight simulation of the AFCS, but also provide an easier way to design GUI and board card driver API. Based on this flight testing system, hardware in the loop (HIL) ground experiments have been made successfully with the AFCS, which helped the product to finalize in the future.

**Key words:** flight simulation; test system; Simulink; C++; mixed programming

### 0 引言

随着 Matlab/Simulink 的广泛应用和快速原型化思想的深入推广, 越来越多的仿真测试系统采用 Simulink 建立数学模型, 通过 RTW (Real-Time Workshop) 工具箱自动生成代码, 并将代码下载到实时仿真计算机中, 通过多种数据通讯板卡与被试产品建立通讯, 进行半物理仿真测试<sup>[1]</sup>。支持这种测试系统建模方式的实时仿真测试系统主要包括 VxWorks、xPC、ADI、dSPACE、iHawk 等货架产品, 它们具有如下显著优势:

(1) 可以与 Simulink 实现无缝集成, 采用 Simulink 可视化建模方式, 仿真模型直观、可读性强;

(2) 仿真模型调试、验证方便, 使用便捷、维护简单;

(3) 采用自动代码生成技术, 无需手工编程, 可快速实现数学仿真到半实物仿真的无缝集成。

但是, 这些昂贵的实时仿真测试系统都可能存在如下局限性:

(1) 监控界面功能简陋, 不便实现复杂的监控界面;

(2) 各类接口板卡驱动有限, 不便于扩展开发;

(3) 数据记录功能不灵活;

(4) 不支持断点调试, 不便进行系统排故。

因此, 这些实时仿真系统往往不能够满足仿真测试系统开发者和使用者全部的试验需求。

针对以上分析, 本文提出一种基于 Simulink 自动生成代码与 C++ 手工编码的混合编程技术, 寻求了一条基于 Windows 系统的飞行仿真测试系统设计方案。该方法不仅继承了 Matlab 系列产品快速原型化设计的快速性与可靠性, 也秉承了传统 C++ 软件开发的优点, 抛弃了昂贵的实时仿真计算机的约束, 将仿真测试系统的功能在 Windows 平台上得到最大的延伸。

### 1 某飞行仿真测试系统硬件结构

某飞行仿真测试系统主要用于自动飞行控制系统(自动驾驶仪)软硬件的全功能仿真测试。如图 1 所示, 系统主要由飞行仿真计算机、控制计算机、视景计算机和飞行操纵设备等组成。控制计算机中运行试验管理软件, 具有数据装订、配置管理、试验监控、数据分析等功能; 视景计算机中运行视景软件和驾驶杆采集程序; 飞行仿真计算机中运行飞行仿真软件, 主功能是运行飞机仿真模型, 进行航电数据拆解包计算, 并通过航空数据总线 (ARINC-429, MIL-STD-1553B) 以及 DIO, AD, DA 等接口与被测设备进行双向数据通讯。

收稿日期: 2014-01-12; 修回日期: 2014-03-08。

**作者简介:** 戴 阳(1982-), 女, 北京人, 工程师, 硕士, 主要从事飞行控制算法设计、飞行仿真系统设计等方向的研究。

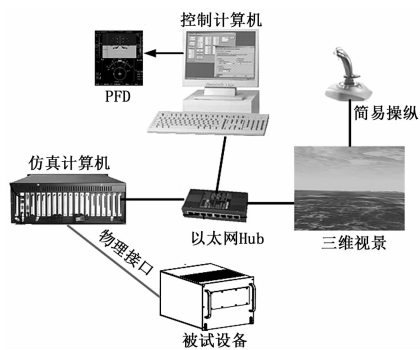


图 1 某飞行仿真测试系统结构示意图

## 2 基于混合编程的测试系统软件设计思想

本测试系统中, 仿真计算机是功能最复杂、硬件接口最多、复杂计算最多、对精度和实时性要求最高的设备, 这对其硬件配置和软件设计提出了较高的要求。从硬件成本、时间成本和软件设计能力等方面综合考虑, 本测试系统采用 Windows 系统作为仿真计算机的操作系统, 采用混合编程的方案对飞行仿真软件进行设计。

飞行仿真软件的工作流程如图 2 所示。当仿真软件启动后, 首先调用软件界面、进行总线板卡初始化, 建立 UDP 通讯, 生成飞行仿真和数据通讯两个线程。由于 Windows 为非实时内核系统, 定时器的选用对于飞行仿真软件是至关重要的。根据飞行仿真线程和数据通讯线程的精度和步长要求, 程序分别采用多媒体定时器 `timeSetEvent` 和系统计数器 `QueryPerformanceCounter` 对两个线程进行分别定时, 最小运算周期分别为 5 ms 和 2 ms, 保证飞行仿真软件的运算精度。为了防止线程抢占资源, 软件采用临界区对象 (`TCriticalSection`) 保护的方法, 锁定关键算法的运算顺序, 从而保证线程间的运算准确无误<sup>[2]</sup>。数据通讯线程的通讯种类多, 且各模块周期不一致, 因此, 程序中采用了多任务分时处理的算法, 把任务分为 20 个区循环运行, 一个循环 40 ms, 可以有序地囊括 20 ms、40 ms、50 ms、100 ms、500 ms 以及事件型的数据通讯需求; 例如, 20 ms 的任务在循环中出现两次; 50 ms 的任务在循环中出现 4 次并每次通过整除判断是否执行。

在保障仿真系统实时性的基础上, 由于飞行仿真模型的设计包含大量气动数据、数学公式、逻辑算法等复杂计算, 并需要进行充分的数学仿真测试, 因此采用 Simulink 模型进行飞行仿真模型的设计和调试, 部分模型可以通过 RTW 工具生成嵌入式代码, 直接被嵌入到 C++ 工程进行混合编程。

基于 Simulink 和 C++ 混合编程的实时飞行仿真软件, 不仅继承了 Simulink 平台大批量数据处理、快速设计验证迭代的成果, 还遗传了 C++ 环境友好的用户界面风格, 开发出灵活的数据记录和试验监控功能, 又满足复杂的板卡驱动和数据通讯的需求。同时, 该飞行仿真软件在调试过程中, 可以通过在线单步调试的方法运行, 为系统调试排故提供了很大的便利<sup>[3]</sup>。

## 3 某飞行仿真软件的实现

飞行仿真软件主要包括飞行仿真、航电数据打解包、总线数据通讯、UDP 数据通讯、数据记录及试验监控等功能。

首先, 采用 Simulink 平台对飞行仿真模型进行设计、实现、调试和测试。飞行仿真模型构架如图 3 所示, 在数学仿真模型中简化了与自驾仪产品的数据接口。图 3 中, 电传控制、襟缝

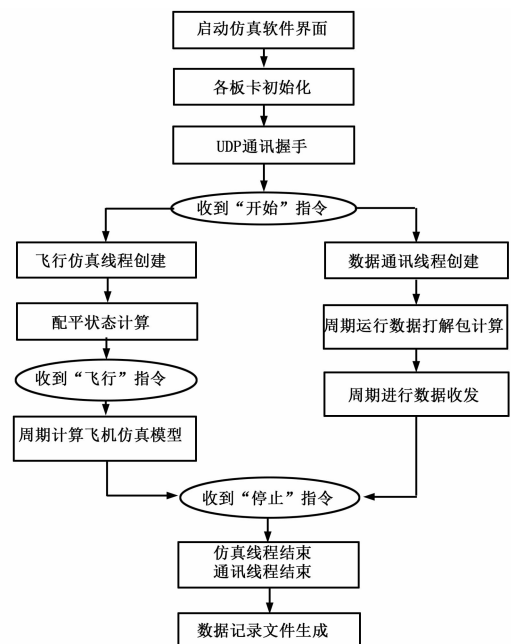


图 2 飞行仿真软件工作流程

翼、起落架、气动力计算、运动方程和发动机模型都通过 RTW 工具箱被直接生成了目标代码, 再通过人工编码的航电模型形成总线数据与自驾仪产品进行数据通讯。

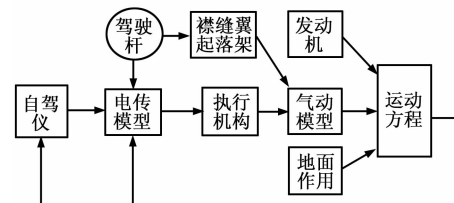


图 3 Simulink 飞行仿真模型结构示意图

另外, 对具有高阶特性的执行机构模型, 软件采用手工编码的方式进行设计。通过仿真试验表明, 由手工编码的执行机构模型与 RTW 生成的其它飞行仿真模型代码相结合, 可以较好地体现系统高阶特性, 同时保证了系统资源的稳定性。

从 Simulink 模型通过 RTW 自动生成代码到 C++ 工程的应用流程均可在 VC++、C++ Builder 等主流设计平台中实现<sup>[4]</sup>。应用流程主要包括:

(1) 目标语言编译器设置。在 Matlab 命令行中通过“`mex - setup`”命令对编译器进行设置, 选用 Microsoft Visual C++ 系列编译器。在模型的配置参数对话框中对 Real-Time Workshop 参数进行配置, 选择嵌入式目标系统文件 `ert.tlc` (Visual C/C++ Projects), 选择 C++ 作为目标代码语言, 并勾选支持 continuous time。

(2) Simulink 模型整理。为了便于模型代码的应用和接口调试, 依次对图 3 中的每一个子系统模块进行单独代码生成。将被生成模块独立保存成一个模型文件 `modulename.mdl`, 并在工作空间中初始化所有已知变量。生成 C/C++ 代码的模型中不能含有 Matlab Function 和 m 文件, 可以通过 Embedded Matlab Function 代替。将模块的所有接口信号连接输入或输出模块 (In/Out), 并命名为对应的信号名称。模型应选择定步长求解器, 并设置合理的仿真步长。

对于连接总线数据输入 (Bus data) 的模型, 如果将模型单独保存成文件, 由于总线数据失去了信号源, Bus Selector 模块将会报错。若通过 Bus Selector 选择的信号不是很多, 可将 Bus Selector 删除, 并替换为输入模块 In。若经过 Bus Selector 选择的信号数据较大, 可以不逐一替换数据接口, 而在该模块原模型文件中右键点击该模块, 选择 “Real-Time Workshop/Build Subsystem”, 这样作为 Bus Selector 输入的所有信号都会被生成该模块的输入, 在使用生成代码时需要甄别哪些信号是被模块使用的, 并给它们赋值。

(3) 编译生成嵌入式代码。在 RTW 对话框中点击 “Build 按钮” 或按键盘 “Ctrl+B” 开始编译, 编译成功后工作路径中会产生一个名为 “modulename\_ert\_rtw” 的文件夹。其中, modulename.h 文件中定义了模型的输入输出接口变量的定义、参数变量定义等; modulename.c 文件中包含输入输出接口变量的声明、模型主算法函数、初始化及结束函数的定义; modulename\_data.c 文件中对模型参数变量进行赋值。

(4) RTW 代码的使用。根据上文所述的操作, 各模块已经完成代码生成, 形成了多个不同名字的 “\_ert\_rtw 文件夹”。将这些文件夹中的文件全部拷贝到 C++ 工程目录下, 如遇重名文件可以进行覆盖。在工程的主程序头文件中包含所有生成模型头文件 modulename.h, 在工程的主程序文件中包含所有模型源文件 modulename.c。将所有模型源文件中模型接口变量 (类型为 ExternalInputs\_modulename 和 ExternalOutputs\_modulename) 的定义移到主程序文件中进行定义, 并删除原变量的定义 (位于 modulename.c 中), 因为这些变量需要被整个工程全局调用<sup>[5]</sup>。

在工程初始化函数中增加各模型代码的初始化函数 modulename\_initialize (); 在工程的结束函数中增加各模型代码的结束函数 modulename\_terminate ()。在各模型代码需要被调用的地方对各函数的数据输入结构体 (modulename\_U) 中的各个变量进行赋值, 调用其主仿真函数 modulename\_step (), 随后可以对其数据输出结构体 (modulename\_Y) 中的各变量进行调用。

由于各模型单独进行代码生成, 在各模型中可能会使用相同的模块和子系统, 因此模型代码中就可能都存在相同名字的函数。当这些模型代码同时应用于同一个 C++ 工程中时, 由于函数重名的情况可能会引起计算错误。这些模型代码主要包括查找表函数、插值函数、积分器等, 可以在采用各函数名后增加模型名后缀的方法, 通过人工对这些函数的名称进行修改。

工程编译时可能存在数学库函数重复定义的警告; 由于数学库函数是统一标准的, 这些警告可以暂时保留, 但是相应的数学库源文件 (如 rt\_atan2.c rt\_pow\_snf.c 等) 和头文件需要被拷贝到工程路径中, 并包含进来。若工程编译报错缺少头文件, 则是因为生成代码仍然需要调用 RTW 工具箱相关的头文件才能够运行, 可以通过将 Matlab 工作路径中相应文件拷贝到工程所在目录的方法加以解决。

#### 4 半实物飞行仿真试验系统验证

仿真系统内部联调结束之后, 首先应对飞行仿真软件进行独立测试, 对自然飞机模型进行控制响应分析, 并与数学仿真结果进行对比; 此时可选取典型状态点, 分别进行平飞、纵向横向和航向控制激励、推力阶跃响应等测试。

其次, 需要确保仿真环境与被测设备之间总线数据通讯的正确性。该阶段测试不仅要验证数据线路的连接状态, 同时还要对航电数据的打解包算法进行验证。测试过程中先采用静态数据测试方法, 一端发一端收, 完成后换向, 验证标准为设备接口设计方案 (IDD); 完成单向测试后, 进行双向通讯测试, 同时借助数据记录功能, 考察动态数据的通讯正确性。

最后, 进行产品联试, 将被测设备连入仿真系统中, 进行硬件在回路仿真。本系统的被测设备为自动飞行控制计算机。接入后, 仿真系统形成了一个闭环控制回路, 同时人在回路中参与设置试验模态发送控制指令, 从而对自动飞行控制计算机的软硬件进行全面的综合测试。试验方法及验证标准可参考分系统铁鸟试验大纲。仿真控制软件界面及仿真曲线如图 4 所示。

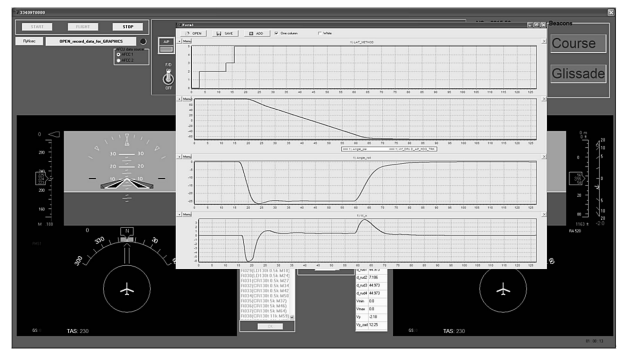


图 4 仿真控制软件及仿真曲线

#### 5 结论

本文基于 Matlab 快速原型的设计思想, 通过 Simulink 建立各飞机分系统模型并进行自动代码生成, 并将其代码应用于 C++ 设计环境中, 通过手工编码完成软件界面、系统定时器、数据通讯, 板卡驱动等功能, 实现了基于 Windows 系统的飞行仿真软件的设计。

基于该软件的飞行仿真测试系统, 设计周期短、硬件成本低, 在保证仿真实时性的基础上, 大大提高了测试系统的灵活性和开放性, 同时兼顾了快速原型设计的快速迭代性和可继承性, 为试验人员提供了便捷的试验监控手段, 为被试产品提供了可靠的实时飞行仿真试验环境。

利用搭建的飞行仿真测试系统, 某型飞机自动飞行控制计算机已经成功的开展了多轮半物理仿真验证试验, 并取得了良好的试验效果, 从而为产品的试飞定型奠定了坚实的基础。

#### 参考文献:

- [1] 杨向忠, 崔文革, 范智刚. 基于模型的飞行控制系统一体化设计技术研究 [J]. 系统仿真学报, 2007, 19 (19): 4411-4414.
- [2] 郭文夷, 戴芳胜. C++ Builder 6.0 程序设计 [M]. 西安: 电子科技大学出版社, 2006.
- [3] 樊晓丹, 孙应飞, 李衍达. 一种基于 RTW 的实时控制系统快速开发方法 [J]. 清华大学学报 (自然科学版), 2003, (7): 895-898.
- [4] RealTime Workshop (r) Embedded Coder User's Guide [Z]. 2007.
- [5] 齐振恒, 孙中立, 李涛. RTW 嵌入式代码自动生成机制与代码结构分析 [J]. 计算机测量与控制, 2010, 18 (3): 639-642.