

基于 RTW 和 VC 的半物理仿真系统的开发

常广晖¹, 王永生¹, 梁述海¹, 王雪山²

(1. 海军工程大学船舶与动力学院, 武汉 430033; 2. 中国人民解放军海军驻大连地区军代表室, 大连 116021)

摘要:通过 RTW 将 Simulink 仿真模型转化成可移植的嵌入式 C++ 代码, 提供了一种从 Simulink 仿真模型平滑过渡到 Visual C++ 集成开发环境的方法, 该方法充分利用了 Simulink 丰富便捷的建模环境, 发挥了 VC 强大的硬件控制功能和人机界面设计灵活的优点。并结合科研实例介绍了利用该方法开发半物理仿真系统的具体步骤和实现方法。

关键词:实时工作间; 半物理仿真; Simulink; Visual C++

Development of HILM System Based on RTW and VC

CHANG Guang-hui¹, WANG Yong-sheng¹, LIANG Shu-hai¹, WANG Xue-shan²

(1. College of Naval Architecture and Power, Naval Univ. of Engineering, Wuhan 430033; 2. PLA Naval Deputy Office of Dalian, Dalian 116021)

【Abstract】This paper presents an approach to integrate simulation model with Visual C++ easily by translating simulation model to embedded C++ codes. The advantages of Simulink and Visual C++ are fully used by this approach. According to a real example the development steps and realization of HILM system based on this approach is discussed.

【Key words】RTW; HILM; Simulink; Visual C++

进行控制系统半物理仿真除了设计和构建半物理仿真的硬件平台外, 一个工作量很大的内容就是半物理仿真软件的开发。Simulink 是一个图形化的建模仿真工具, 能进行电子、电路、控制工程、信号处理等领域的仿真, 具有建模方便灵活的特点, 被越来越广泛地应用于控制系统的设计和仿真。

半物理仿真系统需要连接特定的硬件设备、定制控制软件等, 而 Simulink 无法实现这部分的功能, 但这恰恰是可视化 C++ (Visual C++, VC) 的强项。VC 具有强大的硬件控制功能和灵活丰富的人机界面设计功能。

本文结合 MTU396 柴油发电机组转速控制半物理仿真系统科研实例, 介绍了一种 Simulink 和 VC 相结合, 在 VC 环境下开发半物理仿真系统软件的方法, 使它们各取所长, 功能互补, 大大减少了半物理仿真软件的开发量, 缩短了开发周期。

1 半物理仿真系统简介

电子调速器是柴油机的关键部件之一, 当实际柴油机及其负载性能发生变化且与调速器设计参数不匹配时, 柴油发电机组就无法正常工作, 这时需要修改调速器的控制参数。另外, 电子调速器在出厂前和维修后都需要进行性能试验, 以设定合适的控制参数满足其调节性能。在柴油机上直接进行电子调速器的参数整定、性能测试、维修后性能恢复, 既不安全又不经济, 因此, 进行半物理仿真是调速器进行实际配机试验之前不可缺少的环节, 其工程思路是建立柴油机发电机组全工作范围动态数学模型, 通过输入输出接口电路与电子调速器相连构成一个闭环控制系统, 从而完成电子调速器的相关性能测试试验。只要仿真模型能较真实地模拟柴油机及其负载特性, 测试结果就能较为真实地反映出调速器的实际配机性能。

MTU396 柴油发电机组转速控制半物理仿真系统的输入输出信号主要有 3 种类型: (1) 脉宽调制(PWM)信号: 执行机

构齿条驱动信号; (2) 频率信号: 模拟柴油转速脉冲信号; (3) 开关量信号: 对电子调速器的各种控制信号, 如启动、停止、增速减速控制等。本系统选用研华的 PCI-1780 计数器卡, 它提供了 8 个 16 位计数器通道 (8 路数字量输出和 8 路数字量输入), 可以满足上述需要。

2 半物理仿真系统的开发

本文开发方法需要安装的软件有 Matlab7.04、RTW6.2、Simulink6.2、Visual C++6.0。

2.1 Simulink 中仿真模型的创建

合理地建立柴油发电机组全工作范围仿真模型是实现半物理仿真的基础。由于柴油机机构复杂、影响因素多, 因此在建模分析中要作许多假设和简化处理。建立模型时主要考虑 2 个方面: (1) 较准确地反映柴油机在整个工况范围内变化的稳、动态特性; (2) 满足实时性的要求。本文采用神经网络的方法建立柴油发电机组模型——mtu039.mdl。Simulink 仿真试验表明, 其精度和实时性达到了工程要求。

RTW 能把 Simulink 模型中的某些参数或信号设置为全局变量, 使模型自动生成的可执行代码在目标系统中运行时, 可以方便地与 Simulink 模型交互, 实现在线参数调整和信号通信。利用 RTW 这种功能, 可以把模型中需要调整、监测的参数或者信号设置为全局变量。柴油发电机组模型需要实时监测或调整的信号有 3 个: 柴油机转速, 柴油机齿条位移驱动和发电机负载, 分别设定为 g_speed , g_rack , g_load 。这些变量构成 VC 仿真程序中模块间交互的桥梁。

具体的操作方法为: 在 Simulink 模型中选中需要设置的信号, 点击右键, 选取 Signal Properties... 菜单, 弹出图 1 的

作者简介:常广晖(1979 -), 男, 助教、硕士, 主研方向: 舰船动力装置自动化与仿真; 王永生, 教授、博士生导师; 梁述海, 教授; 王雪山, 工程师

收稿日期:2006-11-04 **E-mail:** cgh_navy@163.com

对话框，根据图 1 所示把信号设置为全局变量。

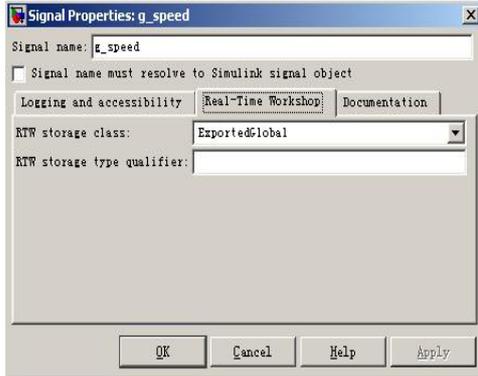


图 1 信号属性的设置

2.2 可移植 C++模型代码的生成

RTW 支持多种目标，所以 RTW 自动生成 C++模型代码有多种选择，对于 VC 而言有 3 种目标可以选择：通用实时 (GRT)目标，通用实时 malloc(GRTM)目标，嵌入式目标。GRT 目标采用实时代码格式，其内存的分配在编译时被静态声明。GRTM 目标采用实时 malloc 代码格式，它与实时代码格式非常相似，区别主要在于实时 malloc 代码格式对内存进行动态声明。嵌入式目标可按嵌入式代码格式生成 C++代码，在运行速度、内存使用量和简化等方面都进行了优化。嵌入式代码采用静态内存分配方式。本文选择了嵌入式目标来自动生成嵌入式代码。

RTW 自动生成的代码分为 2 部分：模型代码和代码运行界面(run-time interface)。这些代码被切分成很多源代码文件，为了管理这个庞大的项目，RTW 还为其自动生成了一个 make 文件。用 VC 打开 make 文件并进行编译，然后将此项目中引用的所有源文件从 Matlab 安装目录中复制出来，并和模型代码共同组成仿真模型代码。模型代码中最关键的 3 个函数如图 2 所示，这 3 个函数构成了仿真模型代码同 VC 程序交互的接口。

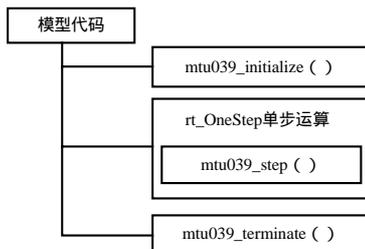


图 2 仿真运算模块接口函数

2.3 模型代码和 VC 程序的集成

模型代码同 VC 程序的整合过程比较简单，把 2.2 节仿真运算模块所有涉及的源文件添加到 VC 程序项目中，需要注意 2 点：

(1)仿真模块的源文件添加到 VC 程序项目后默认应用预编译头文件选项，必须手工取消此项设置，否则编译出错。具体操作方法如下：在 VC 中选择 project 菜单->选取 Settings 项，弹出图 3 的对话框后按图 2 所示设置即可。

(2)在 VC 程序中调用仿真代码的接口函数需要进行全局声明，即在 VC 程序的应用程序类头文件中加入如下代码：

```
extern void mtu039_initialize(boolean_T firstTime);
extern void mtu039_terminate(void);
```

```
extern void rt_OneStep();
```



图 3 项目属性设置对话框

2.4 VC 中仿真系统的开发与整合

半物理仿真软件一般由人机界面模块、数据采集模块、仿真运算模块、数据处理模块组成。MTU396 柴油发电机转速控制半物理仿真系统原理如图 4 所示，人机交互界面如图 5 所示。

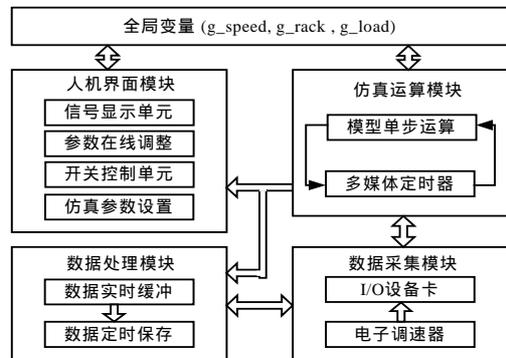


图 4 系统原理框图

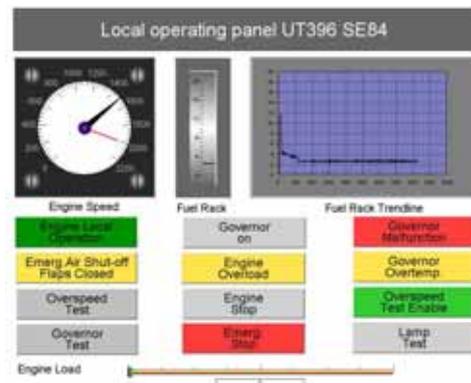


图 5 系统人机交互界面

各模块的实现过程如下：

(1)人机界面模块的实现

人机界面模块构成了 VC 程序的主线程，主要完成变量的实时显示、参数在线调整、控制参数设置和程序的运行控制等功能，其他模块都由它来调度。本系统中涉及的实时变量有柴油机转速、齿杆位移、柴油机负载等，并设计了转速表、棒图、趋势曲线、调整滑杆、功能按钮等图元，充分利用面向对象方法的封装、继承等特点实现了整个图元库，提高了软件系统的可修改性、可重用性和可扩展性。

(2)数据采集模块的实现

数据采集模块通过 PCI-1780 设备卡同电子调速器进行

数据交互,电子调速器的供油量信号以脉宽调制(PWM)的形式由 PCI-1780 设备卡传给仿真程序,仿真程序通过仿真计算把柴油机的转速以频率信号的形式经 PCI-1780 设备卡反馈给电子调速器。数据采集模块由厂家提供的设备驱动 API 函数实现。

(3)仿真运算模块的实现

仿真运算模块在 RTW 转化的仿真模型代码基础上还要加入仿真同步控制,以满足仿真的实时性要求。实时性的关键就是 I/O 数据定时的准确性。众所周知 Windows 不是实时操作系统而是多任务的操作系统,采用 VC 编制的仿真程序很难实现硬件中断,采用软件定时是比较可行的方法,因为现在的计算机速度都很快,能够保证一般硬件在回路仿真的实时性的需要。

VC 中实现精确定时一般有 2 种方法:(1)使用优先级很高的多媒体定时器。它使用单独线程调用一个特殊的回调函数,最小定时间隔为 1ms,能够满足一般实时任务的定时要求。(2)使用高性能计数器。在 Windows2000 中由硬件实现高精度的计数器(high-resolution performance counter),利用它可以获得高精度的定时间隔。由于本应用中电子调速器输出频率为 80Hz,因此选择多媒体计时器实现了仿真控制,定时间隔选为 2ms,即多媒体定时器每隔 2ms 就在其回调函数中调用模型代码接口函数 rt_OneStep(模型单步运算),保证了整个仿真的向前推进,满足了系统实时性要求。仿真运算模块完成模型实时计算是整个系统的核心。

(4)数据处理模块的实现

数据处理模块完成仿真数据的实时记录和定时保存。为了满足数据记录的实时性,仿真程序采用了多线程的技术,将数据定时保存功能放在独立的数据保存线程中执行。数据处理线程和主线程间的同步由事件信号驱动。在 C++ 标准模板库(STL), deque 容器基础上创建了 CPointsCollection 类来完成数据的实时记录和定时保存。采用的策略是:

CPointsCollection 先在内存中开辟 2 块缓冲区。第 1 缓冲区数据放满后,数据开始存放到第 2 缓冲区,然后主线程同时通知数据保存线程将第 1 缓冲区中的数据采用 VC 序列化功能保存到硬盘的文本文件中,等第 2 缓冲区数据放满后,数据开始存放到第 1 缓冲区,主线程同时通知数据保存线程将第 2 缓冲的数据保存到硬盘的文本文件中,这样依次循环保证数据记录的实时性。

MTU396 柴油发电机组转速控制半物理仿真系统的人机交互界面如图 5 所示。

3 结论

本文结合科研实例详细介绍了一种基于 RTW 和 Visual C++ 的半物理仿真系统快速开发方法,该方法充分利用了 Simulink 和 VC 各自的优势。首先利用 Simulink 可视化的建模方法建立系统的仿真模型,然后通过 RTW 将 Simulink 仿真模型转化成可移植的嵌入式 C++ 代码,最后利用 VC 灵活的定制性和强大的界面功能实现了完全独立运行的实时半物理仿真系统。该仿真程序在 VC 环境下可以方便地对仿真程序进行调试。与传统设计方法相比,这种方法具有费用低、效率高等特点。

参考文献

- 1 Real-time Windows Target User's Guide Version 2[M/CD]. U.S: Mathworks Company, 2004-10.
- 2 Writing S-functions Version 6[M/CD]. U.S.A: Mathworks Company, 2005-09.
- 3 PCI-1780 User Manual[M/CD]. Taiwan, China: ADVANTECH Co. Ltd., 2005.
- 4 杨 涤,李立涛,杨 旭,等.系统实时仿真开发环境与应用[M].北京:清华大学出版社,2002.
- 5 Kruglinski D J. Visual C++技术内幕[M].4 版.北京:清华大学出版社,1999.

(上接第 274 页)

4.5 测试与运行结果

Kaffe 在目标平台的运行相当于一个系统进程,在目标平台的调试环境下,新建任务 kaffe_task,让 Kaffe 虚拟机运行起来,调试中虚拟机运行实例在执行不同的 Java 测试类过程中,均能得到正确的结果,测试其 AWT 图形处理功能时,通过执行 Kaffe 官方的 AWT 测试类 WidgetDemo,在电视机屏幕上输出正常的图形界面,而且能够正常响应遥控器的按键操作。上述的测试结果证明,Kaffe 移植成功。

5 Java 虚拟机性能优化

移植后的 Java 虚拟机虽然能够在目标平台上正常工作,但是运行速度相当慢,无法实现正常的人机交互,为此,笔者采取改进的 JIT 方法,进行执行引擎的性能优化。将虚拟机启动中“需要翻译的 Java 方法”翻译成目标指令保存起来,作为资源文件,与虚拟机的代码一起进行联编,同时将 JIT 这些方法的调用接口进行修改,当虚拟机执行此类方法时,直接调用,省去第 1 次运行时指令翻译上的时间开销,优化后,虚拟机的执行速度明显得到改善,对 MHP 应用程序的执行过程进行跟踪,将调用超过一次以上的方法进行上述的

优化测试,完全满足人机交互的要求。上述的优化测试结果,证明这种方法是可行的。因为一旦 Java API 的库确认下来,它们的字节码方法不变,完全可以采用事先编译的策略予以实现,不过,需要牺牲目标平台的一些存储空间。

6 结束语

介绍了 MHP 软件结构、Java 技术、Java 虚拟机的基本概念及其软件层次结构,给出了嵌入式 Java 虚拟机在数字电视机顶盒平台上的移植与实现过程,同时提出了在嵌入式环境下虚拟机的性能优化策略。通过嵌入式中间件 Java 虚拟机,使得 MHP 应用程序能够在不同厂商的机顶盒平台上运行。

参考文献

- 1 European Telecom Standard Insititute. ETSITS101812V1.3.1-2003 Digital Video Broadcasting Multimedia Home Platform Specification 1.0.3[S]. 2003-06.
- 2 Venners B. Inside the Java Virtual Machine[M].北京:机械工业出版社,2003-09.
- 3 赵俊良. MIPS 处理器设计透视[M].北京:北京航空航天大学出版社,2005-06.