

doi: 10.7690/bgzdh.2016.03.012

光电伺服系统控制程序快速开发方法研究

魏 维, 祁 超, 朱文亮, 范大鹏
(国防科技大学机电工程与自动化学院, 长沙 410073)

摘要: 针对控制系统的设计和开发中, 手工方式编写代码存在开发周期延长、效率低下且容易出错的问题, 提出一种基于模型设计思路实现控制程序的自动生成方法。以嵌入式光电运动控制系统为实验对象, 利用 Simulink 的工具箱设计控制器以及搭建系统模型, 并由框图模型生成产品级代码; 通过对比半实物仿真系统和嵌入式系统对电机速度控制的不同效果, 对系统进行验证。实验结果显示: 由 Simulink 模型快速生成的代码能有效地完成直流电机速度控制功能, 控制程序的开发时间缩短 50% 以上, 证明基于模型设计方法在伺服系统开发中的有效性和快速性。

关键词: 光电伺服系统; 基于模型的设计; 代码自动生成; 直流伺服电机

中图分类号: TP273 **文献标志码:** A

Research on Rapid Development Method of Control Program for Electro-optical Servo System

Wei Wei, Qi Chao, Zhu Wenliang, Fan Dapeng

(College of Mechatronic Engineering & Automation, National University of Defense Technology, Changsha 410073, China)

Abstract: In terms of the disadvantages of long development period and inefficiency by manual programming during designing and developing control system, this paper presents a model-based design method with automated code generation. Regard the electro-optical control system as the object of research, use Simulink tools to design controller and build system model, and then generate code automatically. Conduct the experiment and compare the different control effect between semi-physical simulation system and embedded system. The result of experiment demonstrates that the code generated from Simulink model can realize the function of motor speed-control, time consumed in development progress of control program is reduced by 50% or more, indicating that model-based design method is efficient and rapid in development of servo control system.

Keywords: electro-optical servo system; model-based design; code generation automatically; DC servo motor

0 引言

随着科学技术的发展、观测目标机动性能的增强, 对光电伺服控制系统响应速度、稳定精度和跟踪精度的要求也越来越高。基于这样的需求, 国内外科技工作者为提升光电伺服系统的性能进行了深入的控制算法研究; 然而, 随着伺服控制算法复杂程度的增加, 通过手工编程实现算法的难度也在不断加大, 手工方式编写代码会导致系统开发周期延长且效率低下, 还容易出错^[1]。基于此, 笔者以一套嵌入式光电运动控制系统为实验对象, 提出一种利用基于模型设计(model-based design, MBD)思路, 实现控制程序自动生成的快速开发方法。这种方法最大的优势是可以直接从 Matlab/Simulink 框图模型生成产品级代码, 并将代码自动加载到集成开发环境 CCS 中^[2], 显著减少了控制程序代码的编写量, 有效提升了系统的开发效率。

1 dSPACE 半实物仿真系统

在进行控制程序自动生成之前, 需要对系统模

型进行半实物仿真以验证模型的有效性。dSPACE 半实物仿真系统是由德国 dSPACE 公司研制的一种控制系统开发及测试平台, 能够实现与 Matlab 的无缝连接^[1]。该系统拥有具备高速计算能力的硬件系统, 包括处理器、I/O 等, 还拥有方便易用的实现代码生成/下载及试验/调试的软件, 主要应用于控制系统的快速原型设计和硬件在环仿真。

利用 dSPACE 平台进行的伺服系统开发, 可以完成从系统建模、分析、离线仿真到实时仿真的全过程, 使开发人员专注于控制方案的构思而不必再在琐碎的杂务上花费大量的时间, 从而缩短开发周期; 然而在实际系统开发中, 往往需要能在主机上正确运行的仿真模型, 以及能兼容具体硬件环境的嵌入式代码, 如果这种由模型到代码的转换过程采用手工执行, 则耗时耗力, 且容易出错。针对半实物仿真系统的上述特点, MBD 方法及代码自动生成技术可以很好地与 dSPACE 系统实现互补, 使得仿真模型与最终产品级代码无缝连接。

收稿日期: 2015-11-08; 修回日期: 2015-12-26

基金项目: 装备预先项目支持资助(40401020101)

作者简介: 魏 维(1990—), 男, 内蒙古人, 在读硕士, 从事精密伺服控制与数控技术研究。

2 TI C2000 系列 DSP Simulink 支持包

为了使 MBD 方法所适用的嵌入式平台与 Simulink 无缝连接, MathWorks 公司为不同的处理器开发了专门的 Simulink 支持包^[3-5]。本节所介绍的 Embedded Coder Support Package for TI C2000 Processors, 就是 TI 公司专门针对 C2000 系列 DSP 开发的 Simulink 支持包。

TI 公司的 C2000 系列 DSP 支持包几乎包含了 DSP 的所有外设模块, 如图 1 所示。用户需要做的就是根据实际开发需求, 考虑如何将这些模块添加到自己的 Simulink 模型当中, 并对这些模块的工作参数进行配置以满足使用要求。笔者将结合一般直流伺服电机系统的组成, 就 Matlab 中集成的 DSP F28x3x 外设模块的使用和配置作简要介绍。

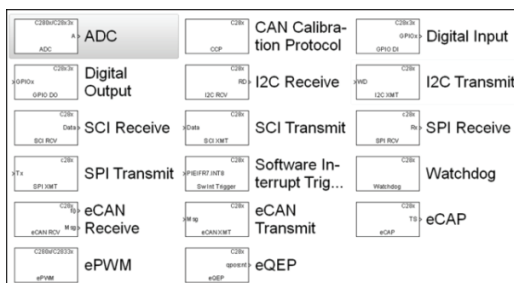


图 1 TI C2000 系列 F28x3xDSP Simulink 模块库

2.1 模型参数配置

在搭建框图模型前, 需要对模型的参数进行配置, 这样才能在模型搭建完毕后生成所需要的代码, 这也是 MBD 方法能否顺利实现的关键环节^[6]。图 2 所示是 Configuration parameters 配置界面, 包含了将框图模型转换成代码所需的配置选项, 其中用户必须进行配置的选项有如下几个: Solver、Hardware Implementation 和 Code Generation。

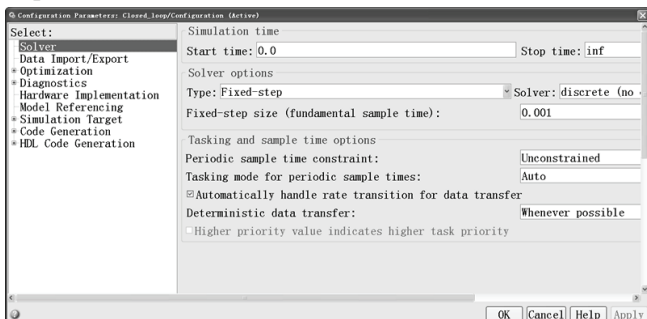


图 2 Configuration parameters 配置界面

下面对上述选项的配置进行详细说明。

Solver: 具体参数设置界面如图 3 所示。根据 Simulink 中代码自动生成的要求, 需要将 Stop time 设置为 inf, 求解器类型选择为 fixed-step 的

discrete(no continuous states)。特别需要说明的是, Fixed-step size 选项所对应的值是生成程序的定时中断周期, 默认为 auto 时, 定时中断周期为 0.2 s^[7], 其余选项为默认值。定时中断周期会决定控制系统的控制周期。

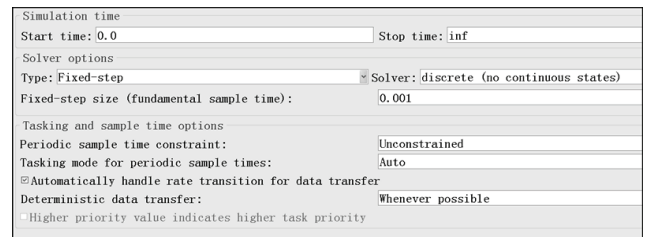


图 3 Solver 参数设置界面

Hardware Implementation: 具体参数设置界面如图 4 所示。这个界面的选项不直接控制硬件或者编译器, 它的目的是描述硬件以及编译器的特性, 从而使 Matlab 软件根据这些配置参数生成能够在指定平台上运行的代码。根据本实验所用的控制芯片, Device vendor 选择 Texas Instruments, Device type 为默认的 C2000 类型。Byte ordering 根据实际情况选择大、小端模式, 其余为默认项。

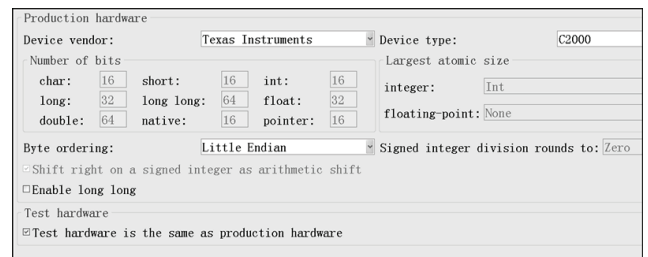


图 4 Hardware Implementation 参数设置界面

Code Generation: 这个界面上的选项是为了激活模型配置而提供通用的信息, 包括目标板选择、文档以及建立过程参数, 具体参数设置界面如图 5 所示。配置模型的第 1 步是选择和配置代码生成的目标。当用户选定目标板, 其他模型配置参数会自动改变以适应所需的目标板。如果用户需要在 IDE 软件中生成工程文件, 则需要选择 idelink_ert.tlc 或者 idelink_grt.tlc 系统目标文件。

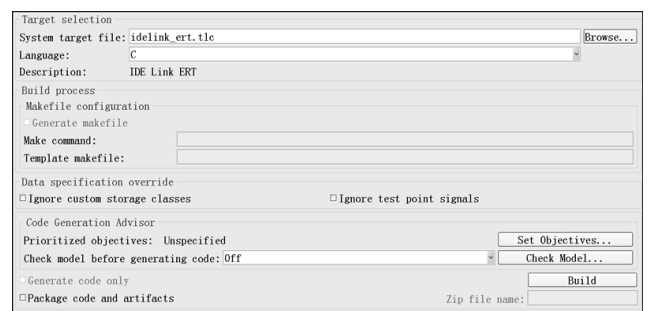


图 5 Code Generation 参数设置界面

2.2 硬件中断设置

基于时间中断的执行时序模型不能满足某些外部事件中断的需求，Hardware Interrupt 模块可用来对 DSP 的外设中断进行配置，它通过允许在 C281x/C28x3x DSP 支持库中由事件管理的其他模块去触发异步中断程序。在 C28x3x 模块中能为中断生成异步程序的模块：ADC, eCAN Receive, SCI Receive, SCI Transmit, SPI Receive, SPI Transmit 等。

一个模型中只能有一个 Hardware Interrupt 模块可被使用，想要增加中断，需放置一个 Demux 模块在 Hardware Interrupt 的输出端并连接相应个数的 function call 子系统。图 6 为利用 Hardware Interrupt 模块实现 2 路 SCI 外部中断的 Simulink 模型以及 Hardware Interrupt 模块的设置界面。Hardware Interrupt 模块的输出是一个函数调用。Function call 的线数等于这个模块手动设定的中断数量。每个中断在这个模块的设定对话框中都有 4 个长度相等的参数待设定。第 1 组参数为外设中断的 CPU 中断号，第 2 组参数为外设中断的 PIE 中断号。以上 2 组参数需要查阅相关芯片的技术手册才能确定。第 3 个参数是外设中断的优先级参数。在添加多个硬件中断后，需要对各个中断设定任务优先级。

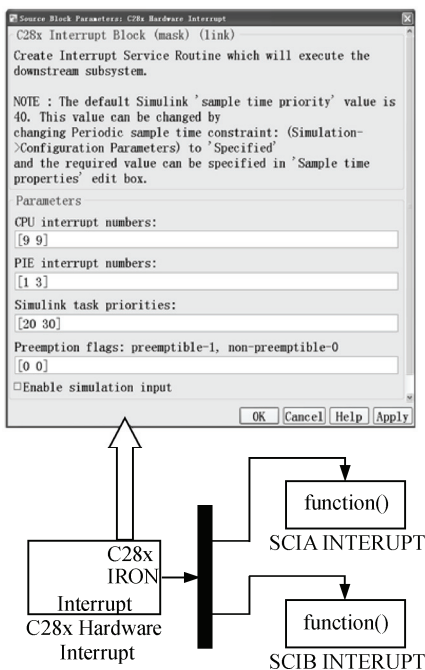


图 6 Hardware Interrupt 模块的使用及其参数设置界面

2.3 串行通信接口 (SCI)

在一般的伺服系统中，常会用到绝对式光电编码器来获取电机的位置信息，这就需要通过 SCI 接口实现编码器与主控板的串口通信。在 C2000 的支

持库中，SCI 通信模块包括 SCI Receive 模块和 SCI Transmit 2 个模块，这组模块用来对 DSP 的 SCI 外设进行配置，并分别进行串口数据的接收与发送，其 Simulink 模块图如图 7 所示。

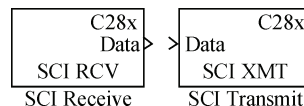


图 7 SCI Receive 和 SCI Transmit Simulink 模块

SCI Receive 模块通过串行通信接口接收主控板数据。SCI Receive 模块支持目标和其他外设间的异步串行数字通信，使用时需要设置的主要参数有：SCI module，用来选择 SCI 模块；Data type，选择输出数据的数据类型；Data length，设置模块接收到的数据类型个数。SCI Transmit 模块通过串行通信接口向主控板传送数据。SCI Transmit 模块通过 COM 口信号传送 int8 或 uint8 型数据。

2.4 串行外设接口 (SPI)

SPI 是一个高速同步串行 I/O 端口，通常情况下用于控制器与外设或者其他处理器之间的通信。典型的应用包括外部 I/O 或设备外扩外设，文中所用到的外设是 D/A 芯片。在 C2000 支持库中，SPI 通信模块包括 SPI Receive 模块和 SPI Transmit 模块，这组模块的功能是用来对 DSP 的 SPI 外设进行配置，其 Simulink 模块如图 8 所示。

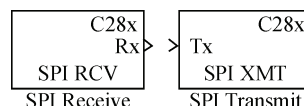


图 8 SPI Receive 和 SPI Transmit Simulink 模块

这组模块在主模式和从模式下都能运行。主模式时，SPISIMO 引脚发送数据，SPISOMI 引脚接收数据。当选为主模式时，SPI 通过发送一个用于串行通信连接的串行时钟信号 (SPICLK) 来开启数据传送。数据传送与 SPICLK 同步，这样保证主机和从机同时接收和发送数据。这个时钟的最大值是 DSP 时钟频率的 1/4。

对于给定的模型，一个模块只能有一个串行接收模块。有 2 个模块类型 A 和 B，可以通过目标参数模块来设置。SPI 发送模块支持 DSP 控制器与外部外设或其他控制器之间的同步串行外设 I/O 口通信。这个模块在主、从 2 种模式下都能运行。采样率是从输入端口继承的，支持的数据类型是 uint16。很多串口接收模块的设置都是在 DSP 板的目标参数模块中设置的，要保证设置正确，符合应用要求。

2.5 模数转换模块(ADC)

ADC 模块用来对 DSP 的模数转换器进行配置。使用 ADC 模块对模数转换器配置后，它才能正常工作，将连接到 ADC 输入引脚的模拟信号转化为数字量，并把转换结果保存到结果寄存器中。其输出是一个 uint16 型适量，范围为 0~4 095(12 为模数转换)。同时 ADC 模块还支持双通道或级联模式。其 Simulink 模型如图 9 所示。

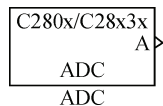


图 9 ADC Simulink 模块

以上对 Matlab 中集成的 DSP F28x3x 外设模块的使用作了简要介绍，设计人员掌握了这些模块的使用方法以及模型所需的配置参数后，就可以用这些模块代替 dSPACE 半实物仿真模型中用于系统验证模型的实时硬件接口。在基于模型设计方法的软件工具中，Matlab 是基础开发平台，其提供的 Real-Time Workshop(RTW)为 CCS 生成工程文件，经过 Embedded IDE Link CC 编译连接到 CCS，随后 CCS 连接到 DSP 并将嵌入式代码下载到目标板上^[8-10]，经过以上步骤就完成了从框图模型到可执行代码的转换。

3 利用基于模型的设计方法进行系统设计

笔者以实现直流伺服电机速度控制为目标进行系统模型搭建。伺服系统电机速度环组成示意图如图 10 所示。其中嵌入式控制板上搭载有用于信号处理的 DSP F28335 芯片，用于模数、数模转换的 A/D、D/A 芯片以及能够实现电机电流闭环控制的硬件电路，嵌入式控制板输出的信号经过电机驱动板转化成用于驱动电机运转的驱动信号，电机的速度信息经测速机测得后反馈到嵌入式控制板中。经过以上环节，在硬件上形成了电机控制的速度闭环回路。

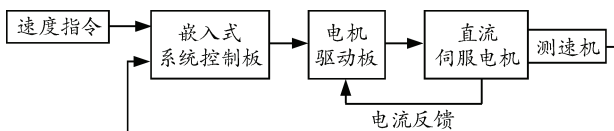


图 10 伺服电机速度闭环控制组成示意图

3.1 系统 Simulink 模型

根据上文所描述的系统构成，以及第 2 章中所介绍的 Simulink 模块，笔者构建了针对嵌入式系统电机速度闭环控制的 Simulink 模型框图，如图 11 所示。模型可实现的功能是通过 In1 接口接收外界

模拟电压信号，经 A/D 转换后为电机指令信号 Cmd_input，经过控制器后控制信号由 Output_DA 模块输出，电机测速机 Tach_input 信号通过 In2 接口反馈到系统中。

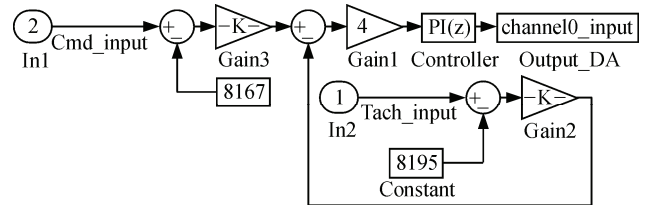


图 11 基于嵌入式伺服系统的 Simulink 模型框图

3.2 D/A 信号输出模块

Output_DA 模块内部构成如图 12 所示。本系统中由于 DSP 片内 D/A 转换模块在分辨率和输出范围方面不能满足使用需求，因而选择使用片外 D/A 芯片 DAC7614 实现数模转换功能，通过 DSP 上的 SPI 模块实现与 D/A 芯片的通信。

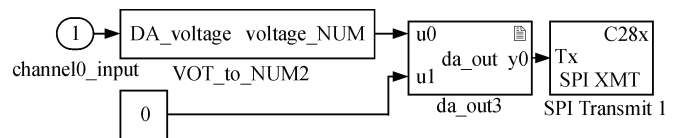


图 12 Output_DA 控制信号输出模块

3.3 A/D 信号采集模块

由于 DSP 片内 A/D 模块的采集范围有限，不能满足测速机信号的采集要求，故在硬件上选用 A/D 芯片 TLC3574 采集测速机的电压信号，将 DSP 内 Mcbsp 配置成 SPI 模式与 A/D 芯片进行通信。但在实际使用时，TI 提供的 Embedded Coder Support Package for TI C2000 Processors 库中没有与 Mcbsp 相对应的模块，所以需要在 Simulink 模型中集成现有的用户代码，实现对 DSP 芯片上 Mcbsp 的配置和使用。

具体实现方法是：

第 1 步，在模型文件的 Configuration Parameter->Code Generation->Target Hardware Resources->Board 中添加源文件 DSP2833x_Mcbsp.c 和头文件 DSP2833x_Mcbsp.h 所在的文件路径；

第 2 步，在生成的工程文件中，找到 xxx_main.c(xxx 表示 Simulink 模型文件名)源文件，在函数 void main()中添加代码：

```

InitGpio_Mcbspa();//初始化 Mcbsp 的 IO 引脚
InitPeri_Mcbspa();//初始化配置 Mcbsp 的寄存器参数
InitChip_AD();//初始化 AD 芯片 TLC3574
    
```

第 3 步，在生成的代码中找到 xxx.c 源文件，在函数 void xxx_step(void)中添加语句：

Tach_input=AnalogSignalInput(0x2000);// 接收并解析得到测速机输出的速度值

Cmd_input=AnalogSignalInput(0x3000);// 接收并解析得到速度指令值

这一步就实现了将 Simulink 模型中的模块与现有程序中代码关联起来的作用。

代码添加后，对模型进行编译运行自动生成代码。需要说明的是，每次进行代码生成后都需按上述步骤添加相应的现有用户代码。图 13 为 Matlab 自动生成的代码，在 CCS 中编译运行的实验截图。

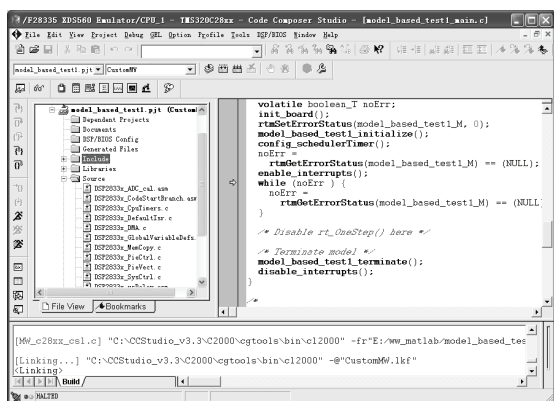


图 13 CCS 调试自动生成代码的界面

4 实验验证

根据前文所研究的内容，为了说明基于模型的设计方法能够自动快速生成可执行的嵌入式代码，能够有效地应用于直流伺服系统的开发过程中，同时对比半实物仿真系统和嵌入式系统对电机速度控制的不同效果，笔者进行了相应的实验验证。图 14 是所用到的实验设备连接示意图。嵌入式系统和 dSPACE 系统可以分别控制电机运行，嵌入式调试主机负责编译，连接自动生成的代码并将其下载到 DSP 中运行；dSPACE 上位机的功能是运行 dSPACE 系统实现对电机的控制和运行数据显示；嵌入式系统板和 dSPACE 接口板都可以接收信号发生器和电机测速的电压信号；2 套系统可分别通过插头与电机驱动板相连接，输出控制信号。

具体操作步骤如下：

第 1 步，利用 dSPACE 半实物仿真系统搭建系统模型，如图 15 所示。将已整定好的 PI 控制器参数输入模型当中。为了控制实验变量，基于 dSPACE 系统的模型和嵌入式控制系统模型中的 A/D、D/A 转换当量均一致，PI 控制器参数也相同。

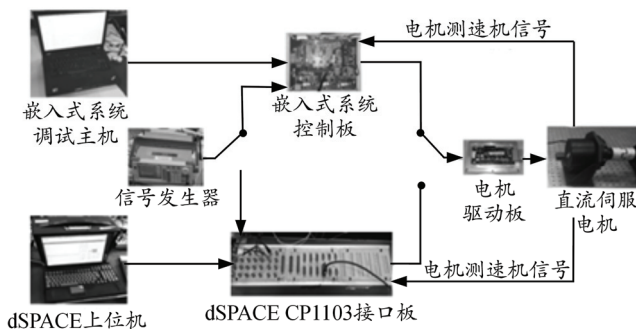


图 14 实验设备连接示意图

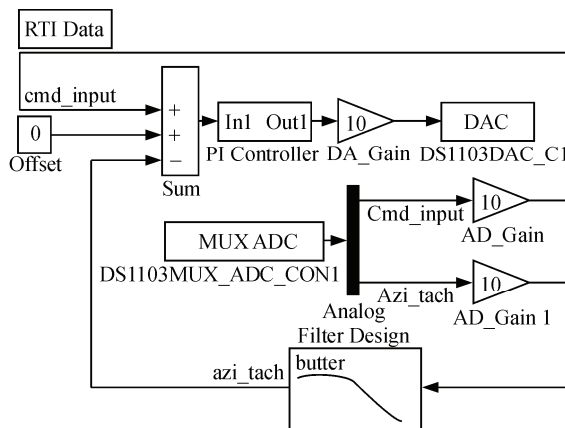
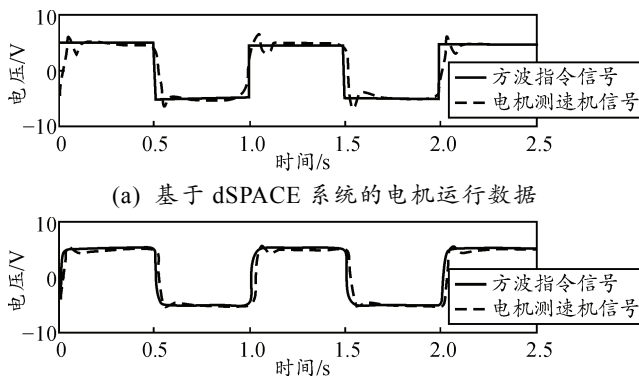


图 15 基于 dSPACE 系统的 Simulink 模型框图

第 2 步，利用 dSPACE 半实物仿真系统进行电机速度闭环控制实验。电机运行指令是由外部信号发生器发出的周期 1 s、幅值 5 V 的方波信号。在 ControlDesk 中记录系统输入及电机运行速度数据。

第 3 步，利用嵌入式系统进行电机速度的闭环控制。在 Matlab 环境下将 3.1 节所搭建的模型进行编译自动生成代码，并在目标系统中运行程序。系统输入也是外界信号发生器发出的方波信号，参数与前述实验相同。在 ControlDesk 中记录信号发生器输入信号以及电机运行的速度数据。

第 4 步，将以上 2 组实验所测得数据绘制成图，如图 16 所示。



(a) 基于 dSPACE 系统的电机运行数据

(b) 基于嵌入式系统的电机运行数据

图 16 实验测试结果

图 16(a)中曲线是在 dSPACE 平台上进行的电机实验数据曲线，图(b)是在嵌入式平台上进行的电机实验数据曲线。图中 2 条测速机曲线略有差异，经过分析，造成这种差异的原因是：嵌入式平台与 dSPACE 平台中 A/D 通道滤波器参数不完全相同，影响了最终实验结果的一致性。

综上所述：通过嵌入式系统实现的电机控制与在半实物仿真平台上实现的电机控制，可得到近似相同的运行性能，说明了代码自动生成技术在嵌入式控制系统开发过程中应用的可行性和有效性。除此之外，Simulink 模型易于搭建，并且可以灵活地集成现有用户代码。经过实际计时，在对 Simulink 建模较为熟悉且控制器参数已确定的前提下，从系统重新建模到嵌入式代码正常运行的时间小于 20 min，远远短于手工编程的时间。

5 结论

笔者以光电伺服系统为研究对象，在基于模型的设计方法及控制程序开发应用方面作了介绍，研究了 TI C2000 DSP Simulink 支持包中常用外设模块在实际系统开发中的应用方法，并将半实物仿真技术和代码自动生成技术相结合应用于伺服系统的开发过程中，进行了相应的实验验证，可使得控制

(上接第 47 页)

表 7 模块 XHJK 和 FXKZ 容错改进措施

| AEID | 可以采用的措施 | 实际采用的措施 |
|------|----------------------------------|--------------|
| XHJK | 在线监控，看门狗，重启，资源自检，接口自检 | 在线监控，接口自检 |
| FXKZ | 在线监控，看门狗，N 版本编程，检查单，重启，资源自检，接口自检 | 在线监控，重启，资源自检 |

5 结论

SABRA 方法是以 FMEA 和 FTA 为基础，在软件需求分析和设计阶段早期，通过对整个嵌入式软件系统潜在故障模式进行分析，识别故障模式之间的联系，找到整个体系结构的敏感元素和不可靠元素，对相应的不可靠部分进行容错、避错等设计，从而达到提高整个体系结构可靠性的方法。在该方法中不仅要确定系统的失效模式，还对失效原因和失效影响附加了特征元素，通过对这些特征的分析建立了事件树，找到体系结构的薄弱环节。从对飞控系统的整个例子来看，SABRA 方法简单、有效。

参考文献：

[1] 刘刚. 军用嵌入式软件可靠性及其保证[J]. 兵工自动化,

程序的开发时间缩短 50%以上，说明了上述方法在光电伺服控制系统中应用的有效性和快速性。

参考文献：

[1] 潘峰, 薛定宇, 徐心和. 基于 dSPACE 半实物仿真技术的伺服控制研究与应用[J]. 系统仿真学报, 2004(5): 936-939.
 [2] 华镛. 基于模型设计的控制程序自动生成[J]. 技术进展, 2010(5): 48-50.
 [3] 刘杰. 基于模型的设计及其嵌入式实现[M]. 北京: 北京航空航天大学出版社, 2010: 42-44.
 [4] 刘杰. 基于模型的设计—MCU 篇[M]. 北京: 北京航空航天大学出版社, 2011: 239-240.
 [5] 刘杰, 周宇博. 基于模型的设计—MSP430/F28027/F28335 DSP 篇[M]. 北京: 国防工业出版社, 2011: 179-186.
 [6] The MathWorks, Inc. Real-Time Workshop Getting Started[Z]. 2010: 72-73.
 [7] The MathWorks, Inc. Simulink 7 Getting Started Guide[Z]. October, 2010: 24-25.
 [8] 陈特. MATLAB-DSP 集成开发环境的研究与设计[D]. 天津大学, 2007: 13-14.
 [9] 齐振恒, 孙中杰, 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): 639-645.
 [10] 高子旺, 顾美康. 基于模型的设计的代码自动生成[J]. 计算机与信息技术, 2009(5): 9-11.
 2008, 27(1): 1-3.
 [2] Bijay K. Jayaswal. 可信软件设计[M]. 吴峻申, 等. 译. 北京: 机械工业出版社, 2013: 36.
 [3] 黄锡滋. 软件可靠性分配 FMEA 及 FTA[J]. 质量与可靠性, 1994, 1(5): 11-14.
 [4] Bachman F, Bass L. Deriving architectural tactics: a step toward methodical architectural design[D]. Carnegie-Mellon University, Pittsburgh, PA. Mar, 2003: 38-40.
 [5] 戴蒙, 高建华. 软件错误的分类、原因及特征[J]. 福建电脑, 2003, 1(5): 1-2.
 [6] Clements P, Kazman R, Klein M. Evaluation Software Architectures: Methods and Case Studies[M]. New Jersey: Addison-Wesley. 2002: 126-128.
 [7] 陆民燕. 软件可靠性工程[M]. 北京: 国防工业出版社, 2011: 168-173.
 [8] 李鹏, 赵逢禹. 软件缺陷关联分析与缺陷排除研究[J]. 计算机科学, 2013, 40(10): 159-162.
 [9] 徐新海. 硬件故障在程序中的传播行为分析及容错技术研究[D]. 长沙: 国防科技大学, 2012: 11-12.
 [10] Dugan J. B. Handbook of software reliability engineering [M]. In: Lyu, M. R. (Ed.), Software System Analysis Using Fault Trees. New York: McGraw-Hill, 1996: 615-659.