

文章编号:1007-6735(2017)01-0035-07

DOI:10.13255/j.cnki.jusst.2017.01.007

基于仿真的汽车电子控制器软件开发方法研究

张万福, 黄影平

(上海理工大学 光电信息与计算机工程学院·上海 200093)

摘要:以车身控制系统中的车窗控制器和雨刮控制器为例,研究一种基于仿真及代码自动生成的汽车电子控制器软件的新型开发方法.利用 Simulink/Stateflow 建立汽车电子控制器算法模型,进行离线仿真验证和快速模型验证,然后通过 RTW 自动生成可执行的代码,最后可将自动生成的代码直接嵌入到控制器硬件中,实现了汽车电子控制系统的软件开发.通过与 Simulink 中仿真结果进行对比,验证了该方法的可行性和正确性.经过实践证明该开发模式可以大大缩短控制系统的开发周期,对于汽车电子系统的开发具有一定的实用价值.另外,在软件开发的各个阶段对设计功能进行测试验证,提高设计开发的正确性.

关键词: 仿真; 汽车电子控制器; 自动代码生成

中图分类号: U 463

文献标志码: A

Development of Automotive Electronic Controller Software Based on Simulation Method

ZHANG Wanfu, HUANG Yingping

(School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract: Taking the power window system and wiper control of an automobile as examples, a new-typed development method of the automotive electronic controller software was proposed, which incorporates the simulation and code generation techniques. The method employs the Simulink/Stateflow to establish controller algorithm models for offline simulation and fast verification, and generates an executable automotive code by RTW which can be directly embedded in the electric controller unit. The method was tested and verified to be feasible and accurate, by comparing with the simulation results of Simulink. It is proved that the development mode can greatly shorten the development cycle of the control system and has certain practical value for the development of automotive electronic control system. Moreover, the method allows validation testing being conducted in the early stage of development cycle. It greatly increases the correctness of the design and development.

Keywords: simulation; automotive electronic controller; automotive code generation

收稿日期: 2016-06-29

基金项目: 国家自然科学基金资助项目(61374197); 教育部博士点基金资助项目(20133120110006); 上海科委科技创新行动计划资助项目(13510502600)

第一作者: 张万福(1990-), 男, 硕士研究生. 研究方向: 机器视觉检测和汽车电子仿真. E-mail: 495775041@qq.com

通信作者: 黄影平(1966-), 男, 教授. 研究方向: 智能汽车障碍物探测和汽车电子仿真. E-mail: huangyingping@usst.edu.cn

1 汽车电子控制器开发现状

随着汽车电子系统的复杂程度以及在汽车中所占比重的不断增加,汽车电子控制器的开发测试成为新车开发的一个重要而复杂艰巨的任务.图1(a)是汽车电子系统开发的V模型,具体过程包括如下步骤:电子系统总体设计,电控单元功能设计,电控单元硬件和软件开发,电控单元测试,集成测试,系统测试,样车测试.传统的开发方法存在两方面的问题.一方面,在设计初始阶段,对系统功能指标要求依赖于文字描述,这种方法难以准确直观地描述子系统内部功能以及子系统之间的互相关联和影响,而且使用者对文字描述的理解往往不同.另一方面,对设计和开发的验证严重依赖于对实体器件的测试.传统的电控系统开发验证都是在一个实体的汽车电子系统验证平台上来进行的.它包含车内所有的电气排线和电控单元.其缺点是对设计和开发的验证必须等到所有的ECU、线缆、执行器、传感器等制造完成后才能开始,而且测试效率低下,测试覆盖范围有限.这样的方法由于不能早期发现研发中的问题,常常导致许多重复性的修改,增加了研发的费用和时间.

实时计算机仿真技术被用于模拟复杂的汽车电子系统,并在此基础上进行设计正确性及有效性的测试和验证^[1-2].这种虚拟集成技术对汽车电子系统的开发有3方面的好处.a.可以模拟汽车的各种实际运行状况,建立一个虚拟的汽车电子平台,对其功能和子系统之间错综复杂的相互影响和关联进行

动态的测试.b.在ECU软件开发方面,用可执行的模型来描述系统功能、状态转换,减少文字叙述的含糊性,而且所开发的功能模型,可以直接用于自动生成代码,从而大大提高软件开发的效率^[3-4].c.在设计验证和开发测试方面,汽车制造厂或供应商可以在研发不同阶段对正在开发的电子系统进行测试.例如,在研发的最初阶段,所有的电控单元处在用模型描述的功能设计阶段,称为模型在环(MIL),功能模型可以帮助设计人员进行设计有效性的测试;在电控单元软硬件开发阶段,模型被转换成代码在快速成型控制器上执行,称为软件在环(SIL),可以测试软件开发正确性;在集成测试阶段,部分ECU被制造出来,虚拟平台中对应的软件模型可以被取代,仿真系统可用于验证这部分器件的功能,包含部分硬件,称为硬件在环(HIL).图1(b)显示对应于传统V模型的模拟测试的3个阶段.

基于仿真的开发方法已经应用于一些汽车子系统的开发,例如,汽车发动机控制系统^[5]、传动系统^[6]、刹车系统^[7]、车身控制^[8].这些工作都是对汽车局部子系统的仿真和开发.文献[9]提出了利用仿真技术测试电控单元的在线诊断功能.文献[10]提出了建立汽车电子系统的整体模型,重点验证整车电子系统的集成性能.作者曾经成功地将仿真技术和机器视觉技术结合用于汽车信息系统和仪表系统的功能验证^[11-12].本文以车身控制系统中的车窗控制和雨刮器控制为例,研究基于仿真的汽车电子控制器的开发过程.以MathWorks公司的Matlab软件作为主要的开发平台,利用它所提供的Simulink工具包建立相应的功能及控制模型,利用基于事件

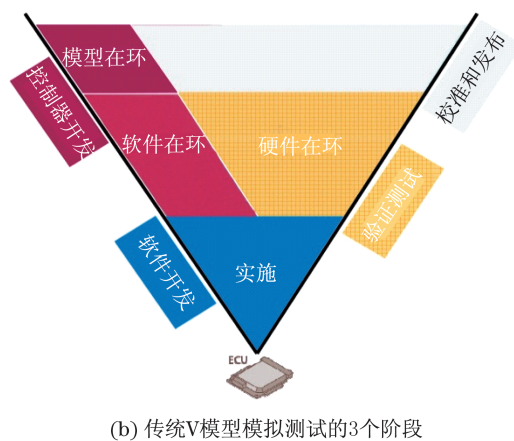
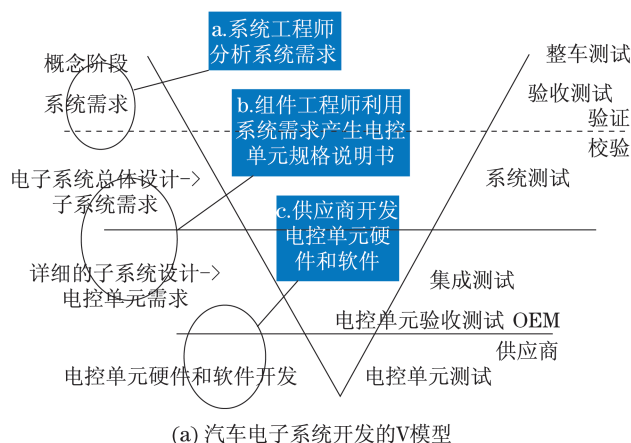


图1 汽车电子系统V模型开发流程

Fig.1 V-model development process of automotive electronic systems

驱动的 Stateflow 模块构建控制算法模型,最后使用 Real-Time Workshop 将模型编译成可直接嵌入到控制器的 C 代码,提高控制器的开发效率。另外,在开发的各个阶段对设计功能进行测试,提高设计开发的正确性。

2 功能介绍及模型开发

2.1 电动车窗功能及控制器模型介绍

电动车窗的主要功能如下:

a. 手动控制功能. 每扇车门都可以手动控制车窗的升降,打开(关闭)车窗,可按住(拉起)相应车窗的开关键。

b. 自动控制功能. 每扇车门都可以控制车窗的自动升降,自动打开(关闭)车窗,可按压(拉起)相应车窗的开关键小于 1 s,车窗会启动自动打开(关闭)模式,停止自动打开(关闭)车窗,向上拉起(按压)车窗开关。

c. 自动避障功能. 如果车窗在手动关闭或自动关闭的过程中遇到障碍物将会停住,然后向下倒退。若想关闭车窗应拿开障碍物,然后再次操作开关。因

此,在车窗完全关闭之前应检查车窗上乘客手和物件等是否移开。

控制器模型的开发主要采用基于事件驱动的 Stateflow 状态图进行模拟仿真。控制模块首先确认司机和乘客输入的有效性,判断车窗的位置是否达到了顶端或者底端,以及是否存在障碍物,将相应的输入信号传递给利用 Stateflow 状态图 (statechart) 建立的控制算法模型,输出控制命令。状态机包括电动车窗的基本状态:上升、自动上升、下降、自动下降、静止状态以及紧急状态。状态机仿真这些基本状态之间的状态切换以及驾驶员和乘客对电动车窗的优先级状态,同样也包括了在车窗上升过程中遇到障碍物的仿真。如图 2 所示,每一个方框代表电动车窗的一个状态。正常状态下,控制器处于无障碍物即 safe 状态;当控制器模型只接收到乘客下降的控制信号时,控制器首先进入乘客控制下降状态;100 ms 过后如果还检测到乘客下降的控制信号,则进入乘客手动控制下降状态,否则进入乘客自动控制下降状态。在乘客或者司机控制模式时,如果遇到障碍物,控制器会自动进入紧急状态,将车窗下降。

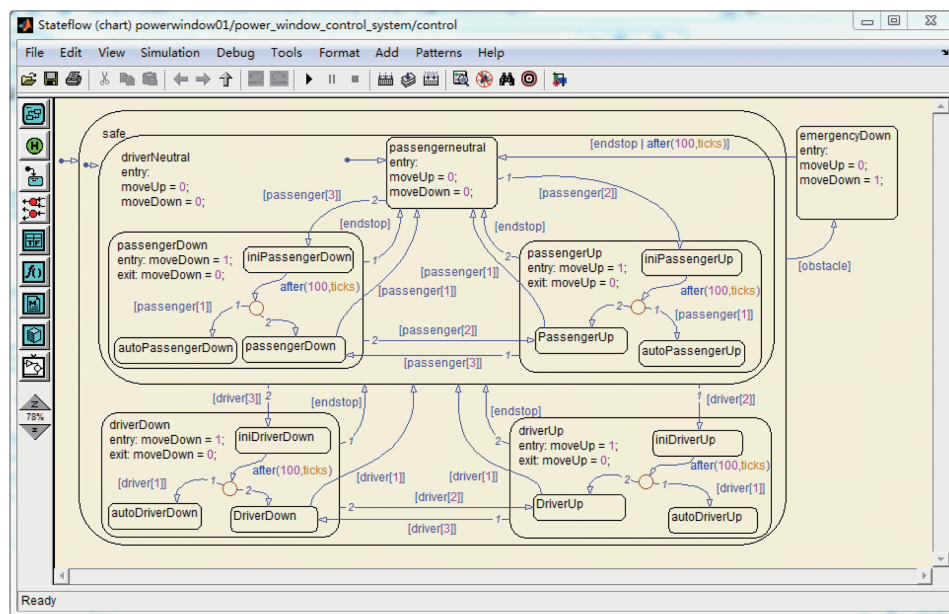


图2 电动车窗控制模型

Fig.2 Power window controller model

2.2 雨刮器系统的功能及控制模型器介绍

雨刮器的功能如图 3(见下页)所示,点火开关在 ON 位置时雨刮器按规定模式进行操作。

a. MIST:要使雨刮器运转 1 次,移动控制杆到

MIST 位置并释放。如果保持控制杆在此位置,雨刮器持续运转。

b. OFF:雨刮器不工作。

c. INT:雨刮器按同样的刮水间隔时间间歇运

转. 要改变间歇时间设置, 可以转动间歇时间调整钮 B.

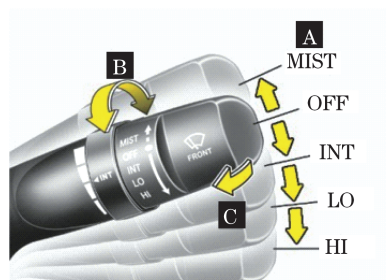
d. AUTO: 雨传感器位于挡风玻璃上端, 检测雨传感器降雨量并以适当间隔时间控制刮水周期. 降雨量越多, 雨刮器运转间歇时间越短. 停雨时, 雨刮器停止. 要改变间歇时间设置, 可以转动间歇时间调整钮 B.

e. LO: 雨刮器低速运转.

f. HI: 雨刮器高速运转.

雨刮器模型(如图 4 所示, 图中的每一个方框代表雨刮器所处的状态, 控制器模型根据接收到的控制信号, 对状态进行相应的切换, 如控制器模型接收到高速模式信号, 控制器进入高速模式状态, 其他雨刮器控制模式同理)仿真了雨刮器的各种状态, 随着输入信号的切换, 雨刮器模型切换到对应的状态中.

状态机包括了雨刮器的基本状态: 雨刮器高速状态、低速状态、间歇状态、点动状态、停止状态和自动状态. 状态机仿真这些基本状态之间的状态切换, 用于验证控制算法的有效性.



A—雨刮器速度控制 B—雨刮器间歇控制
C—喷水器喷水控制(如有配备)

图 3 雨刮器功能图

Fig.3 Wiper function diagram

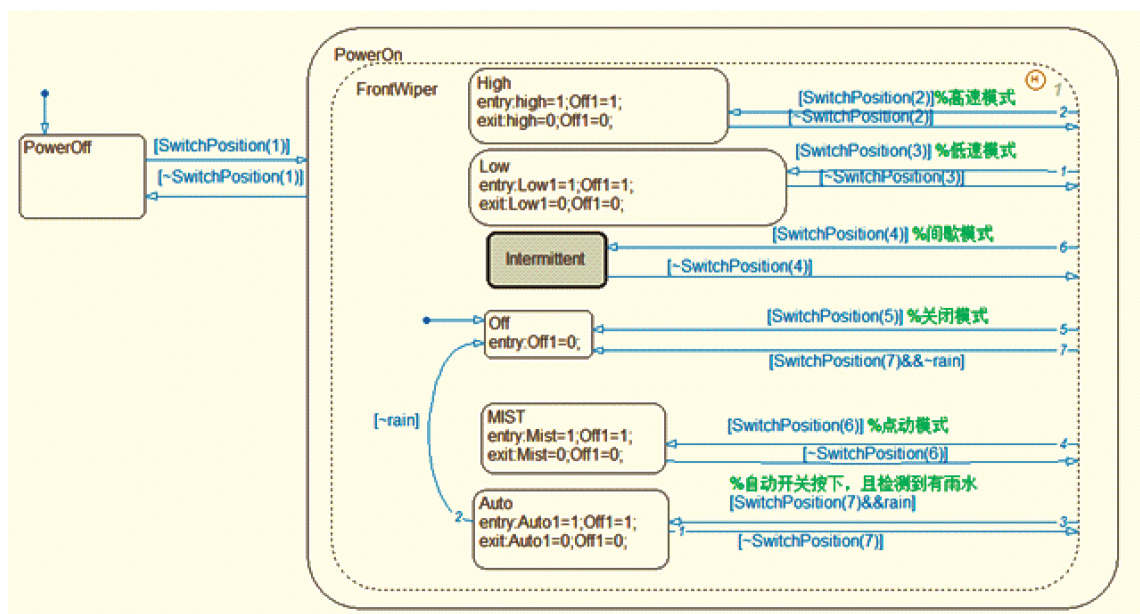


图 4 雨刮器控制模型

Fig.4 Wiper controller model

3 实验

3.1 仿真模型覆盖率的验证

3.1.1 电动车窗模型覆盖率验证

利用 Matlab/Simulink 提供的模型覆盖测试工具可以对汽车电动车窗模型测试序列覆盖率进行验证. 模型测试案例通过每个条件分支到达相应状态的百分率就是模型的覆盖率, 它有助于我们评估每一种状态, 度量测试案例训练模型的程度. 多个条件

分支可以到达一个状态, 例如, 当 100 ms 之后或者到达顶端, 可以从紧急状态切换到自由状态. 要实现全覆盖, 就要求测试案例的每一个分支都要验证 true 和 false 的情况. 雨刮器模型覆盖率测试案例如图 5 所示.

在这个测试案例中, 在步骤 0, 所有开关都是关闭的, 然后定义了乘客手动上升信号、乘客自动上升信号、乘客手动下降信号、乘客自动下降信号, 以及对应的驾驶员相应输入信号和障碍物信号. 乘客和驾驶员输入信号中, 1 表示相应按钮按下, 0 表示相

step	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
passenger up	0	1	1	0	1	1	1	1	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
passenger down	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0
driver up	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
driver down	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
object	1	1	1	1	0.3	0.3	0.3	0.3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

图5 电动车窗测试信号序列

Fig.5 Power window test signal sequences

应按钮没有按下.在此建立的车窗运动模型的最高位置为0.4,将障碍物设置为1表示不存在障碍物,设置为0.3表示在车窗0.3位置存在障碍物.

生成测试和创建 Simulink 确认和验证覆盖报告的命令.

a. 创建一个保存覆盖结果的测试对象.

```
testObj1 = cvtest('powerwindow01','first_test','load firsttest')
```

b. 仿真系统 65 s.

```
[dataObj1,T,X,Y] = cvsim(testObj1,[0 65])
```

c. 创建和打开一个包含覆盖数据的 HTML 报告.

```
cvhtml('powerwindow01 report',dataObj1)
```

报告中各个模块的判定结果如下:

“driver switch”和“passenger switch”模块的判定结果是覆盖 75%,没有子模块.原因是输入信号序列中没有测试上升和下降同时成立的情况. “power window control system”覆盖结果 96%.子模块“driver down”,“driver neutral”,“passenger down”,“passenger up”,“driver up”的覆盖情况分别为 100%,95%,100%,100%,100%.执行模块“window system”覆盖 100%,上升和下降的动作都覆盖到了.覆盖报告验证了测试序列案例较为完好地覆盖了绝大部分状态,能反映出各个状态的执行状况.

3.1.2 雨刮器模型覆盖率验证

雨刮器模型覆盖率测试案例如图 6 所示.

time	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	
Power	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	
High	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Low	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Int	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Off	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	1	0	0
Mist	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
Auto	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0
Fre	1	1	1	1	1	1	1	1	1	1	2	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
rain	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0

图6 雨刮器测试信号序列

Fig.6 Wiper test signal sequences

这个测试案例分别测试了雨刮器的高速、低速、间歇、关闭、点动和自动的状态.生成的测试报告显示,所有的子状态的覆盖率均为 100%,说明测试案例覆盖了雨刮器系统各个功能的状态验证.

3.2 代码生成

Real-Time Workshop 是一个基于 Simulink 的代码自动生成环境,能直接从 Simulink 的模块中产生优化的、可移植的和个性化的代码,并根据目标配置自动生成多种环境下的程序.利用它可加速仿真过程,生成可在不同的快速原型化实时环境或产品目标下运行的程序.

Simulink 模型构建完成之后,利用 Real-Time Workshop 编译成 C 代码和普通的实时程序,可以验

证模型的功能是否完善.步骤如下:

a. 设置程序参数.在生成代码之前,必须要对一些参数进行设置,如果采取系统默认的参数设置,在生成代码时将会出现错误,这里采用之前设置好的参数即可.

b. 生成 C 代码.在 Simulink 窗口中选择 Simulink → Configuration Parameters 命令,打开 Simulink 系统参数设置对话框,单击 Real-Time Workshop,选中 Generate HTML report 复选框,单击 Build 按钮,即可生成 C 代码.也可以利用 make_rtw 命令执行代码生成. Build 命令调用目标文件 grt.tlc,使用指定的模板文件 grt_default_tmf 来生成 makefile,然后 Real-Time Workshop 利用这个

makefile 来建立程序.

生成代码主要包括了以下头文件和源文件.

a. 算法函数源文件 model.C, 主要包括以下 3 个重要函数: (a) 初始化函数, model_initialize; (b) 算法运行函数, model_step; (c) 终止函数(一般为空), model_terminate.

b. 本函数所使用的函数和数据声明 model_private.h.

c. 算法函数头文件 model.h.

d. 参数结构体定义, 当被调制成全局变量时不会生成 model_data.C.

e. 数据和参数结构定义 model_types.h.

4 仿真结果分析

4.1 电动车窗模型仿真与软件运行结果分析

Simulink 模型构建完成之后, 可以直接利用 M 脚本文件运行模型, 记录模型运行时间, 绘制出电动车窗位置图, 运行结果如图 7 红线所示. 图中, 蓝线为预期结果, 红线为仿真结果. 可以看出电动车窗的运动轨迹与预期结果相同(图中将模型的仿真结果值减去了 0.05, 用于对比).

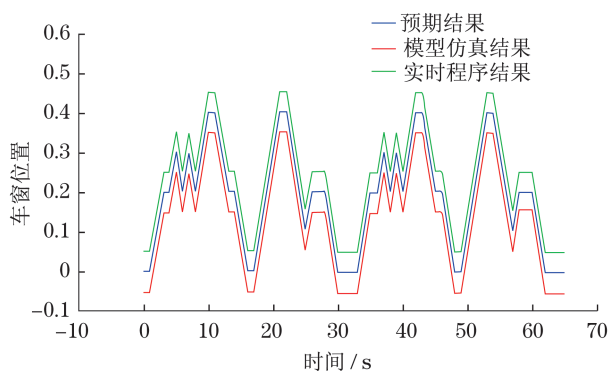


图 7 电动车窗 Simulink 模型运行结果和代码生成程序运行结果对比

Fig.7 Power window simulink model test results compared with code test results

通过 powerwindow01 命令可以引导生成的实时程序的运行, 在 M 脚本文件中引导程序运行, 记录时间, 读取生成的数据, 与 Simulink 模型运行的结果以及期望的结果作比较. 由运行结果图 7 绿色曲线所示, 电动车窗运行的位置和 Simulink 模型运行的结果以及期望的结果相同, 验证了生成代码的可靠性(图中将生成代码运行结果值加上了

0.05). 运行时间上, Simulink 模型的运行时间大约为 0.955 s, 生成代码的运行时间大约为 0.227 s, 自动代码生成工具所产生的程序可以大幅提高 Simulink 模型运行的效率.

4.2 雨刮器系统模型仿真结果与分析

Simulink 模型构建完成之后, 可以直接利用 M 脚本文件运行模型, 记录模型运行时间, 绘制出雨刮器各个状态的运行结果. 如图 8 红线所示, 可以看出仿真的结果与预期结果相同. 为了与预期结果对比, 图 8 以高速模式为例, 将模型仿真的结果增加 0.2, exe 程序运行结果增加 0.4, 雨刮器的其他控制模式的仿真验证结果同样与预期相同.

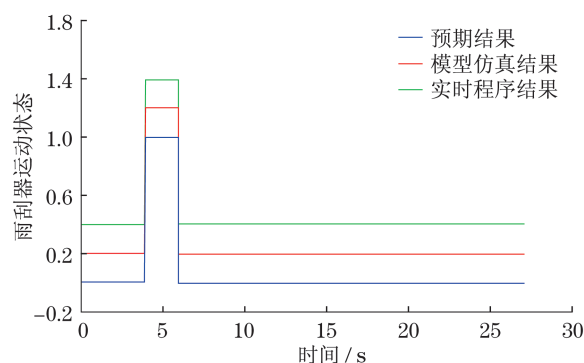


图 8 雨刮器 Simulink 模型运行结果和代码生成程序运行结果对比

Fig.8 Wiper simulink model test results compared with code test results

通过 powerwindow01 命令可以引导生成的实时程序的运行, 在 M 脚本文件中引导程序运行, 记录时间, 读取生成的数据, 与 Simulink 模型运行的结果以及期望的结果作比较. 由运行结果图 8 绿色曲线所示, 生成代码运行的结果和 Simulink 模型运行的结果以及期望的结果相同, 验证了生成代码的可靠性. Simulink 模型的运行时间大约为 2.033 s, 生成代码的运行时间大约是 0.227 148 s.

5 结束语

通过电动车窗和雨刮系统两个实例介绍了基于仿真的汽车电子控制器软件的开发方法, 在汽车电子控制器的开发阶段, 采用建立模型及代码自动生成技术的方法, 开发电控单元软件, 是现阶段基于 V 模式汽车电子开发方法中重要的一个环节. 实验表明产生的代码与模型运行结果一致, 同时可以利用

模型和产生的代码对设计进行验证.该方法可以提高汽车电子控制器软件的开发效率,提高设计的正确性.对于电子控制器外围环境的建模,比如机械系统和执行器的模拟还有待进一步的开发,从而完善电子系统的仿真建模.

参考文献:

- [1] KENDALL I R, JONES R P. An investigation into the use of hardware-in-the-loop simulation testing for automotive electronic control systems[J]. *Control Engineering Practice*, 1999, 7(11): 1343 - 1356.
- [2] 黄影平. 基于模型的现代汽车电子系统开发和测试[J]. *汽车技术*, 2008(2): 31 - 34.
- [3] BEINE M, OTTERBACH R, JUNGMAHN M. Development of safety-critical software using automatic code generation[C] // 2004 SAE World Congress. Detroit: SAE International, 2004.
- [4] LIN A, TSAI G L, CHEN B C. Embedded systems development of an electric power assisted steering system using MATLAB/SIMULINK/Real-time workshop[C] // 2004 SAE World Congress. Detroit: SAE International, 2004.
- [5] LEE W, YOON M, SUNWOO M. A cost-and time-effective hardware-in-the-loop simulation platform for automotive engine control systems[J]. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 2003, 217(1): 41 - 52.
- [6] 张炳力, 杜红亮, 金朝勇, 等. 6AT 控制器的快速控制原型及硬件在环仿真研究[J]. *汽车工程*, 2011, 33(8): 680 - 684.
- [7] LEE S J, KIM Y J, PARK K, et al. Development of hardware-in-the-loop simulator and vehicle dynamic model for testing ABS [C] // 2003 SAE World Congress. Detroit: SAE International, 2003.
- [8] 杨国胜, 贾天阳, 王贺飞. 基于标准架构及 Simulink/Stateflow 的车身控制器软件开发平台[J]. *汽车电器*, 2015(4): 54 - 55, 58.
- [9] MOUZAKITIS A, COPP D, PARKER R, et al. Hardware-in-the-loop system for testing automotive ECU diagnostic software [J]. *Measurement and Control*, 2009, 42(8): 238 - 245.
- [10] TSAMPARDOUKAS G, MOUZAKITIS A. Deployment of full vehicle simulator for electrical control system validation[C] // *Proceedings of UKACC International Conference on Control*. Cardiff: IEEE, 2012: 551 - 556.
- [11] HUANG Y P, MCMURRAN R, AMOR-SEGAN M, et al. Development of an automated testing system for vehicle infotainment system [J]. *The International Journal of Advanced Manufacturing Technology*, 2010, 51(1/2/3/4): 233 - 246.
- [12] HUANG Y P, MCMURRAN R, DHADYALLA G, et al. Model-based testing of a vehicle instrument cluster for design validation using machine vision[J]. *Measurement Science and Technology*, 2009, 20(6): 065502.

(编辑: 丁红艺)