

基于模型设计的 SVPWM 嵌入式软件开发^{*}

张世垣

(青岛科技大学 自动化与电子工程学院 青岛 266061)

摘要: 在电力电子领域,复杂控制算法的应用增加了软件的复杂度,高端 DSP 的使用也使得开发硬件驱动代码变得繁琐,传统手工编码和测试手段难以高效率地开发控制系统的嵌入式软件。针对这一问题,研究了一种以 Simulink 为平台、可实现自动代码生成和早期验证的嵌入式软件开发方式——基于模型设计(MBD)。首先介绍 MBP 基本概念和开发流程,然后针对 SVPWM 软件的开发,详细给出算法模型的建立过程及模型早期验证和定点化方法,之后进行软件在环测试和算法模型与底层驱动的集成,最后一键式自动生成 C 代码,并将代码嵌入 DSP 控制的电机控制系统上进行测试。研究结果表明,通过 MBP 的规范流程,可高效地实现复杂控制算法的嵌入式软件开发。

关键词: 电力电子;嵌入式软件开发;Simulink;基于模型设计;SVPWM

中图分类号: TN773 **文献标识码:** A **国家标准学科分类代码:** 510

Embedded software development of SVPWM based on model-based design

Zhang Shiyuan

(College of Automation & Electronic Engineering, Qingdao University of Science and Technology, Qingdao 266061, China)

Abstract: The complexity of software is enlarged greatly in power electronic field because of complicated control algorithm applied, and the development of hardware drive codes become tedious with advanced DSP used. As a result, the traditional methods of programming and test by manual can't develop the embedded software of control system efficiently. Aiming at this problem, a kind of embedded software developing method, model based design (MBD), is studied in this paper, by which the embedded codes can be generated automatically and early test can be executed on Simulink platform. Firstly, the basic conception and workflow of MBD are introduced. Secondly, aiming at the software development of SVPWM algorithm, the build process of algorithm model, model early test and means of fixed-point are explained in detail, and after that, software-in-the-loop test and integration of algorithm model and bottom drive are demonstrated clearly. Finally, the C codes are generated automatically by one-click and embed the code on the DSP-controlled motor control system for testing. The study results show that the embedded software of complicated algorithm can be developed efficiently through the standardized processes of MBD.

Keywords: power electronic; embedded software development; Simulink; model based design; space vector pulse width modulation (SVPWM)

0 引言

随着电力电子技术不断进步和复杂控制理论的应用,电力电子系统的嵌入式软件规模变得庞大,其开发负担显著增加。而传统的嵌入式软件开发流程为需求分析、算法设计、代码实现、验证与测试^[1],4个阶段彼此孤立,呈单流水线模式,尤其是算法设计与代码实现的分离和软件测试的滞后性,严重影响了开发效率,传统方式难以高效率地开

发复杂控制软件。

基于模型设计(model based design, MBD),是一种通过 Simulink 建模来进行算法设计与代码实现的嵌入式软件开发方式;其主要思想是以模型设计为中心,验证与测试贯穿整个开发过程,而代码采用自动生成方式^[1]。MBD 具有的早期验证和代码自动生成优势,将工程师从繁重的编码和测试工作中解放出来,而专注于算法和模型的研究设计,能够极大减少开发过程中的错误并提高开发效率,已在

收稿日期:2018-08-23

^{*} 基金项目:国家自然科学基金(60940018)、山东省自然科学基金(ZR2018LF008)、山东省本科高校教学改革研究(Z2016M007)项目资助

欧美地区得到成熟应用^[2-4]。

对于 SVPWM,它是目前性能最为优越的变频技术,在电力电子领域得到广泛应用。然而其原理和实现过程也较为复杂,国内对其代码的开发多采用手工编程,效率较低。鉴于国内对 MBD 了解较少、有关 MBD 用于电力电子软件开发的介绍也较少的情况,本文将首先介绍 MBD 对嵌入式软件开发的支持和开发流程,然后结合 SVPWM 技术,详细研究 MBD 开发 SVPWM 软件的方法流程及关键问题,展示 MBD 在复杂代码开发中的高效率优势,最后将生成的代码下载到 TMS320F2812 中进行测试,通过硬件实验来验证其有效性。

1 基于模型设计的代码开发与流程

嵌入式软件包括应用层代码和底层硬件驱动代码,应用层代码是控制算法的实现,而底层驱动是针对硬件的操作,如单片机的初始化、寄存器配置等。在 MBD 中,应用层代码可以直接由算法模型自动生成,底层驱动的开发可以通过硬件支持包进行。MBD 支持众多半导体厂商的微控制器,如广泛应用于电力电子系统的 TI_C2000 系列 DSP,其硬件支持包内 DSP 片上外设的底层驱动均被图形化封装,用户只需对封装模块进行少量参数设置便可完成驱动程序的开发,能够极大地提高工作效率。

MBD 应用于嵌入式软件的开发流程^[1-5]如图 1 所示。由开发流程可见,MBD 能够实现模型和代码的早期验证,改善传统软件测试的滞后性;代码由模型自动生成,且开发中产生的错误是通过修改模型来解决而并非直接修改代码,这种方式更加简便易行。

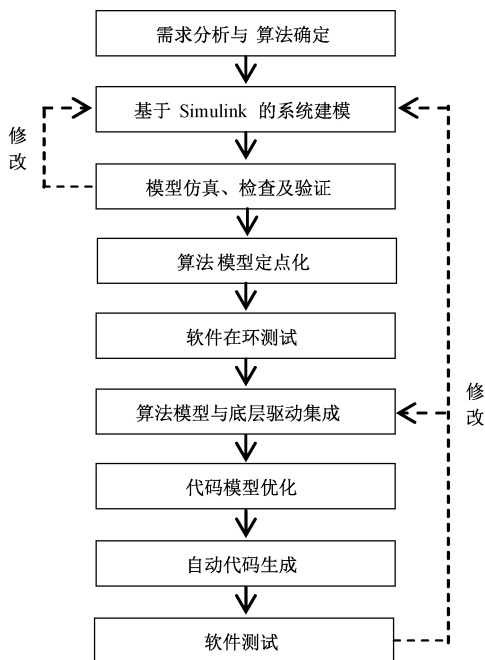


图 1 MBD 的嵌入式软件开发流程

2 SVPWM 嵌入式软件的开发

SVPWM 最初的应用领域是交流电机控制,主要思想是将逆变器和电机视为一体,以产生圆形旋转磁链为目标来控制逆变器工作^[6]。通用的三相逆变器共有 8 种开关状态,可产生 8 个基本电压空间矢量,如图 2 所示。其中 6 个有效工作矢量 $u_1 \sim u_6$ 将空间分为 6 个扇区,并形成正六边形旋转磁链,其余两个为零矢量 u_0 、 u_7 。若要产生接近于圆形的旋转磁链,则需要任意空间角度的电压矢量,方法是通过相邻的两个基本电压矢量合成,如图 2 中 u_1 、 u_2 合成期望电压矢量 u_s ,其中 T_s 是逆变器开关周期, t_1 、 t_2 分别为 u_1 、 u_2 的作用时间,剩余时间使用零矢量来填充。

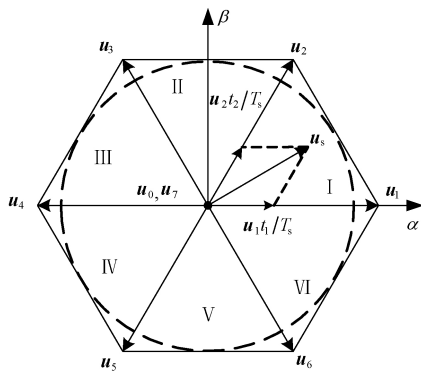


图 2 基本电压空间矢量与矢量合成

2.1 SVPWM 系统模型的建立

对于 MBD,系统模型的建立尤为重要,正确而完善的模型能够减少错误的发生,加快后续开发进程。系统模型包括算法模型和硬件电路模型两部分。

1) SVPWM 算法模型的建立

为计算方便,并减少代码量,基于两相静止正交 $\alpha\beta$ 坐标系对 SVPWM 算法进行分析和建模,主要分为以下 3 个步骤^[7-9]。

(1) 判断期望电压矢量 u_s 所属扇区

计算 u_s 位于不同扇区时其 $\alpha\beta$ 轴上的分量,并分析各扇区 u_s 的角度取值范围,可知扇区值由以下 3 个变量 U_1 、 U_2 、 U_3 的符号决定:

$$\begin{cases} U_1 = u_{s\beta} \\ U_2 = \sqrt{3}u_{sa} - u_{s\beta} \\ U_3 = -\sqrt{3}u_{sa} - u_{s\beta} \end{cases} \quad (1)$$

式中: u_{sa} 和 $u_{s\beta}$ 为 u_s 在 $\alpha\beta$ 轴上的分量大小。

为简化判断过程,定义 A、B、C 3 个布尔变量,并使其取值规则如下:

若 $U_1 > 0$, 则 $A = 1$, 否则 $A = 0$;

若 $U_2 > 0$, 则 $B = 1$, 否则 $B = 0$;

若 $U_3 > 0$, 则 $C = 1$, 否则 $C = 0$ 。

令：

$$N = A + 2B + 4C \tag{2}$$

由于式(1)的约束, A、B、C 只有 6 种组合, 故 N 的取值只有 6 种, 其取值与 u_s 所属扇区存在一一对应关系, 如表 1 所示。

表 1 N 值与 u_s 所属扇区的对应关系

N	3	1	5	4	6	2
扇区	I	II	III	IV	V	VI

扇区判断的 Simulink 模型如图 3 所示, 模型输入为 $\alpha\beta$ 坐标系下 u_s 分量大小, 输出即为扇区值。在建模过程中, 应将重点关注的信号和变量予以命名, 以使后续生成的代码中产生相同的变量名, 便于代码与模型间的双向追踪。

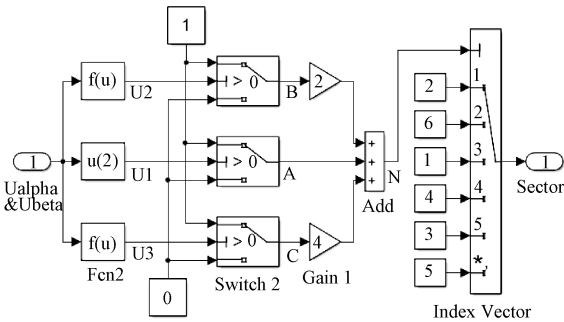


图 3 扇区判断模型

(2) 确定两相邻基本电压矢量的作用时间

计算 u_s 位于不同扇区时基本电压矢量的作用时间, 可得到各扇区共同的中间变量 X、Y、Z。

$$\begin{cases} X = (\sqrt{3} T_s / U_{dc}) u_{s\beta} \\ Y = (\sqrt{3} T_s / 2U_{dc}) (\sqrt{3} u_{sa} + u_{s\beta}) \\ Z = (\sqrt{3} T_s / 2U_{dc}) (-\sqrt{3} u_{sa} + u_{s\beta}) \end{cases} \tag{3}$$

式中: U_{dc} 为直流母线电压。

进一步可确定出各扇区相邻基本电压矢量作用时间 t_x 、 t_y 与 X、Y、Z 的对应关系, 如表 2 所示, t_0 为零矢量作用时间。

表 2 基本电压矢量作用时间

扇区	I	II	III	IV	V	VI
t_x	-Z	Z	X	-X	-Y	Y
t_y	X	Y	-Y	Z	-Z	-X
t_0	$T_s - t_x - t_y$					

当期望电压矢量 u_s 过大时, 相邻两电压矢量不足以合成 u_s , 会出现过调制现象^[8], 导致 $t_x + t_y > T_s$, 过调制会使逆变器输出电压波形失真, 必须予以处理。常用处理方法如下, 令：

$$\begin{cases} t_x = t_x T_s / (t_x + t_y) \\ t_y = t_y T_s / (t_x + t_y) \end{cases} \tag{4}$$

基本电压空间矢量作用时间的 Simulink 求解模型如图 4 所示。

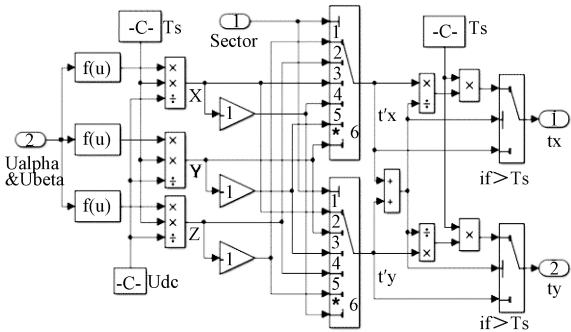


图 4 基本电压矢量作用时间的求解模型

(3) 确定逆变器开关状态切换时刻

基本电压矢量的正确施加是通过逆变器开关状态切换来实现的, 所以需要确定逆变器切换时刻。当采用七段式切换方法^[6-7], 并取三角载波幅值为 T_s , 计算出各扇区切换时刻通过以下变量 t_a 、 t_b 、 t_c 确定。

$$\begin{cases} t_a = (T_s - t_x - t_y) / 2 \\ t_b = (T_s + t_x - t_y) / 2 \\ t_c = (T_s + t_x + t_y) / 2 \end{cases} \tag{5}$$

定义 T_a 、 T_b 、 T_c 为各扇区开关状态切换时刻, 其与 t_a 、 t_b 、 t_c 的对应关系如表 3 所示。

表 3 逆变器开关状态的切换时刻

扇区	I	II	III	IV	V	VI
T_a	t_a	t_b	t_c	t_c	t_b	t_a
T_b	t_b	t_a	t_a	t_b	t_c	t_c
T_c	t_c	t_c	t_b	t_a	t_a	t_b

开关状态切换时刻的求解模型如图 5 所示。

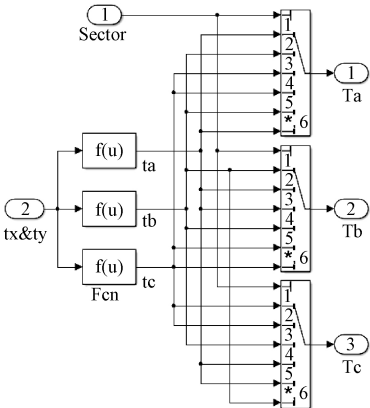


图 5 逆变器开关状态切换时刻的求解模型

T_a 、 T_b 、 T_c 构成了 SVPWM 的三相调制波,此调制波对三角载波进行调制,产生驱动逆变器工作的 6 路 PWM 波^[7-8],调制波与载波的调制模型如图 6 所示。至此, SVPWM 的 3 部分算法模型建立完成。

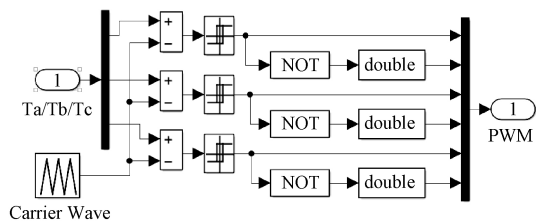


图 6 调制波与载波的调制模型

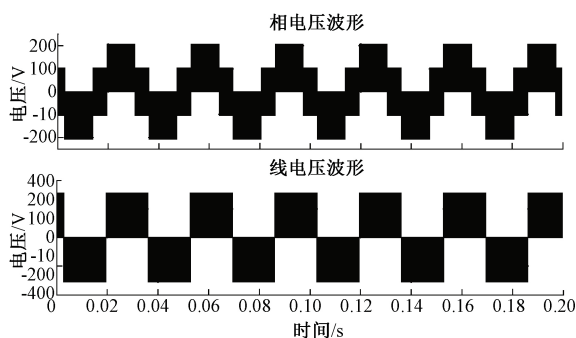


图 8 SVPWM 系统模型仿真结果

2) 硬件电路模型的建立

硬件电路模型可通过 SimPowerSystems 工具箱中的电力器件快速建立^[10],根据实际硬件情况,此处采用 310 V 直流母线电压和三相通用逆变器,负载采用三相 RL 电路模拟。将建立好的硬件电路模型与算法模型整合,得到如图 7 所示的 SVPWM 系统模型。

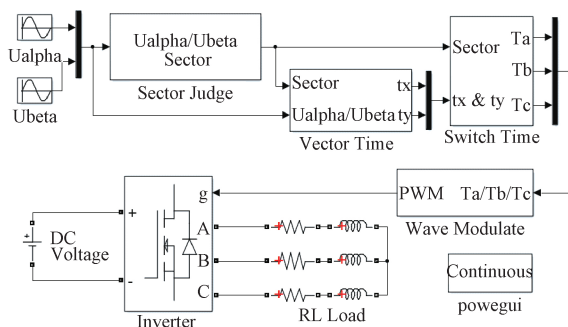
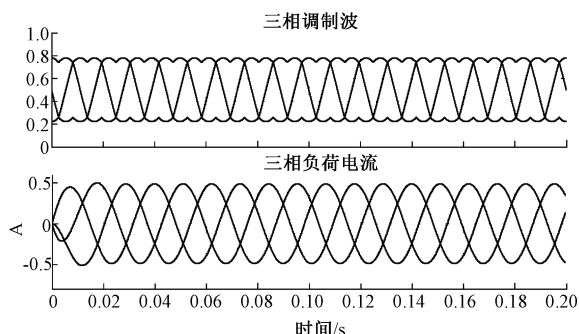


图 7 SVPWM 系统模型

2.2 SVPWM 模型的早期验证

在 MBD 中,模型的早期验证手段包括仿真、Model Advisor 检查和 Design Verifier 验证^[11]。仿真工作能够帮助用户及早发现模型中的逻辑错误,不断修改错误直到仿真结果符合控制需求。SVPWM 系统模型的仿真结果如图 8 所示,其中图 8(a)所示为三相马鞍形调制波和三相 RL 负载电流波形,图 8(b)所示为负载相电压波形和线电压波形。仿真结果均符合理论分析并满足控制需求。



(a) 三相调制波与负载电流仿真波形

系统仿真成功后,需要对模型进行 Model Advisor 检查和 Design Verifier 验证。这些工作能够识别出算法模型中隐含的问题和限制代码效率的部分,寻找设计中的冗余或缺失的需求;尤其对于复杂模型,这些验证测试工作必不可少。两个验证工具位于 Analysis 菜单下,用户可根据需求从多个方面对模型进行验证测试, MATLAB 会自动完成测试工作并给出修改建议,用户应根据建议改进模型直至验证通过。

2.3 算法模型定点化

定点型处理器由于低成本、低功耗等优势而得到广泛应用,但其浮点运算能力较差,为使之能够高效地处理数据,通常需要将数据进行定点化。

传统手工编程方式中,定点化工作需要手动定标或调用定点库函数^[12],编码效率低、难度高且易出错。而在 MBD 中,数据的定点化是通过对算法模型的定点化来实现的,右键单击算法模型后选择 Fixed-Point Tool,此工具有自动定标功能,用户只需在其指导下设置好定标参数和处理器型号, MATLAB 就能一次性完成针对具体型号处理器的定标工作,并且可以检查出定标后的数据是否存在溢出现象,保证用户安全而高效地完成定标工作。定点化的模型将来会生成定点化的嵌入式代码。本文使用的 TMS320F2812 为定点型 DSP,需要进行算法模型定点化。如果用户使用浮点型处理器,则无需此项工作。

2.4 软件在环测试

上述工作完成后,便可进行代码早期测试。在 MBD 中常用方法是软件在环(SIL)测试^[11],即在 Simulink 环境中,将算法模型自动生成 C 代码并编译成 SIL 模块,与原算法模型同时进行非实时性仿真,比较 SIL 模块与原算法模型是否存在功能差异,以完成对代码的早期检验。在模型参数配置的 Code Generation 中设置生成 SIL 选项,然后右键单击算法模型并编译即可生成 SIL 模块。本文的软件在环测试模型如图 9 所示,图 9 中 SVPWM 即编译生成的 SIL 模块,采用的比较方法是将 SIL 模块的输出与算法模型的输出相减,若比较结果为 0,则说明通过了软件在环测试。

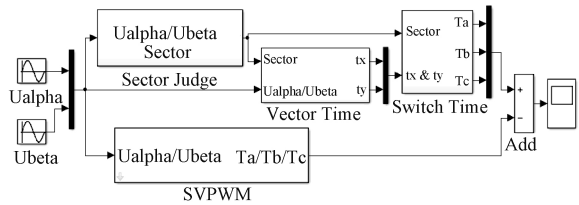


图 9 SIL 模块与原算法的比较模型

2.5 算法模型与底层驱动的集成

所有验证通过后,便可将算法模型与底层驱动进行集成,以得到能够用于嵌入式 C 代码生成的代码模型。本文使用的 TMS320F2812 的所有驱动模块集成在 Embedded Coder 工具箱中,配置这些驱动模块只需要了解 DSP 及其片上外设的基本工作原理,重点掌握少量与算法功能实现有关的寄存器即可,并不需要熟悉所有寄存器,更不需要逐个位域配置寄存器。

例如,本文用到事件管理器 EVA 的 PWM 波形发生功能,需配置通用定时器 T1 的时钟频率、计数方式和周期寄存器等^[12]。假设,T1 的时钟频率配置为高速外设时钟频率 75 MHz 的 1/4 倍,计数方式采用增减计数,而 SVPWM 的载波频率采用 10 kHz,则周期寄存器的值 T1PR 计算方法如下:

$$T1PR = \frac{75 \text{ MHz}/4}{10 \text{ kHz}} = 1875 \quad (6)$$

相应驱动模块 C28xPWM 的定时器页面配置如图 10 所示,只需填写数值和设置选项即可。

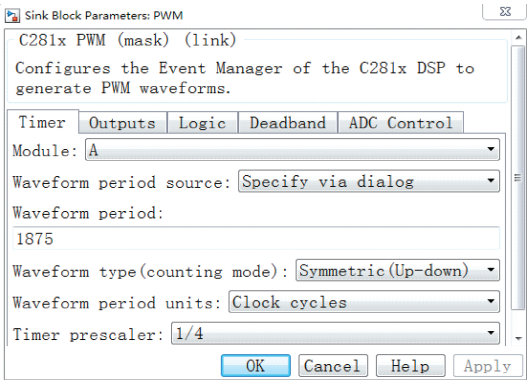


图 10 C28xPWM 的定时器配置

再将模块的逻辑输出、死区时间设置正确后就完成了 PWM 模块的底层配置工作。上述配置将来会生成 DSP 片上外设的驱动代码。

为使原算法模型的调制波幅值与 DSP 中真实的载波幅值相匹配,需将调制波乘以 1875 再送入 PWM 模块作为比较寄存器的值。集成后的代码模型如图 11 所示,图 11 中 F2812 模块是目标板配置模块,DSP 的初始化和型号配置通过此模块进行设置。

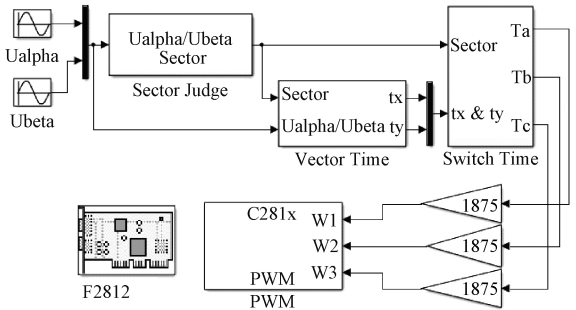


图 11 集成后的代码模型

代码模型将用于嵌入式代码的生成,应只保留算法模型和驱动模块,硬件电路模型要予以删除,因为实际系统中硬件设备都是真实的,对硬件模型进行代码生成没有意义。在 DSP 中,调制波和载波的调制过程也是由硬件自动完成的,故代码模型中也无需包含图 6 所示的调制模型。

2.6 自动代码生成

代码模型建立后,便可进行自动代码生成。代码生成前还可进行一些优化工作,以生成更加符合需求的 C 代码。可在模型参数配置的 Code Generation 页面中选择代码优化目标,从代码的高效性、节约存储空间、易读性等多方面进行优先级设置;还可为处理器指定代码生成时的代码替代库,这样能够将纯粹的应用层算法生成具有底层针对性的嵌入式代码,极大提高代码的运行效率。本文使用的处理器是 TMS320F2812,故指定为 TI28X 替代库。

优化设置完成后,点击代码模型中的 Build Model 按钮,MATLAB 会启动 CCS 软件进行代码生成,生成的工程文件在 CCS 中自动打开,且附有详细代码注释。使用 MBD 方式开发嵌入式软件,如果模型建立得足够完善,模型和代码的早期验证足够细致,那么将会一次性生成符合控制需求的 C 代码。

3 SVPWM 软件的运行测试

3.1 仿真测试

通过仿真测试初步验证模型的准确性。将 SVPWM 系统模型应用于永磁同步电机(permanent magnet synchronous motor, PMSM)矢量控制系统中^[13],得到仿真模型如图 12 所示。

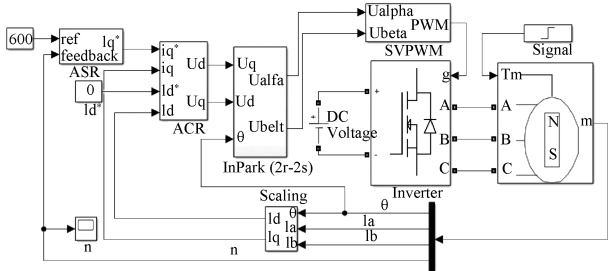


图 12 PMSM 矢量控制系统的仿真模型

搭建好仿真模型后,合理设置仿真参数进行仿真测试实验。电机的相电压与线电压仿真波形与 SVPWM 系统模型的仿真结果一致,其转速仿真波形如图 13 所示。从图 13 可以看出,系统运行正常,具有良好的动静态特性。

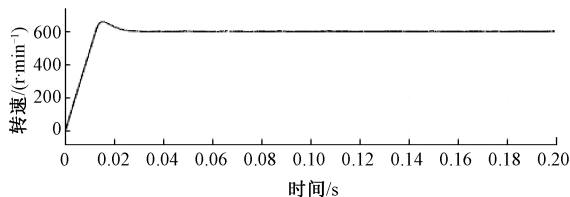
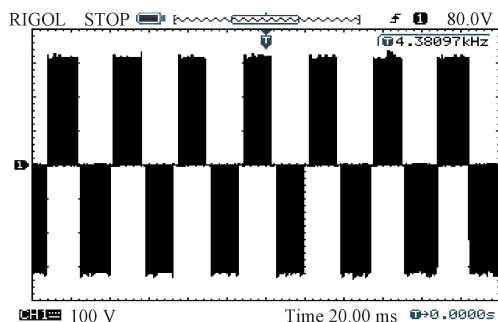


图 13 电机转速仿真波形

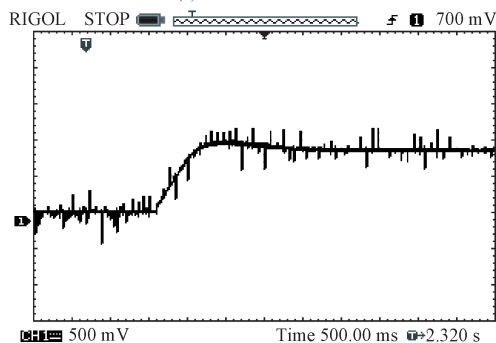
3.2 实物测试

利用天煌公司的 THRS2 型 DSP 电机控制综合实验开发平台,将生成的 SVPWM 模块的 C 代码替代其人工编写的相应代码部分,然后将 C 代码通过仿真器下载到 DSP 中,运行程序^[14-15]。

使用双踪示波器进行观察,观察结果如图 14 所示,其中图 14(a)所示为电机线电压波形,图 14(b)所示为电机启动的转速反馈波形。



(a) 电机线电压波形



(b) 电机启动的转速反馈波形

图 14 电机线电压波形及启动的转速反馈波形

硬件实验结果与模型仿真结果相符,并满足控制需求,即验证了 MBD 方式开发 SVPWM 嵌入式软件的有效性;同时,完善的模型和早期验证工作,使得软件测试过程非常顺利,也表明了 MBD 开发复杂电力电子软件的高效性。

4 结 论

本文将基于 MBD 的嵌入式软件开发方式引入电力电子系统的软件开发中,并将其用于 SVPWM 的控制代码实现。首先在 Simulink 下建立 SVPWM 算法及硬件电路模型,进而经过模型仿真、软件在环测试、自动代码生成等步骤快速得到 C 代码,最后将自动生成的代码编译下载到 DSP 中运行,并通过了软件测试。研究结果表明,使用 MBD 方式开发电力电子嵌入式软件是可行并且非常高效的,它在更复杂的电力电子系统,如交流电动机的矢量控制、电动汽车控制等领域的嵌入式软件开发中会更显优越性,具有广阔的应用前景。本文的研究内容为电力电子软件开发提供了新思路,对相关领域研究者也有较好的参考意义。

参考文献

- [1] 刘杰. 基于模型的设计——MSP430/F28027/F28335 DSP 篇[M]. 北京:国防工业出版社,2011.
- [2] MHADHBI I, SAOUD S B. Model-based design approach for embedded digital controllers design[J]. International Journal of Automation and Control, 2011, 5(3): 267-283.
- [3] O'SULLIVAN, DARA, SORESENSEN, et al. Model based design tools in closed loop motor control[C]. PCIM Europe, International Exhibition & Conference for Power Electronics, 2014.
- [4] MORELLI M, MORO F, RIZANO T, et al. A robotic vehicle testbench for the application of MBD-MDE development technologies [C]. Emerging Technologies & Factory Automation, IEEE, 2013. 1-4.
- [5] WANG L M, WANG G N, WANG W, et al. MBD-DSP: a model based design solution for DSP [C]. International Conference on Electrical and Control Engineering, IEEE Computer Society, 2011: 4561-4564.
- [6] 阮毅, 陈伯时. 电力拖动自动控制系统: 运动控制系统[M]. 第 4 版. 北京:机械工业出版社, 2009.
- [7] 达正花. 电压空间矢量脉宽调制原理[J]. 甘肃科学学报, 2004, 16(3): 65-68.
- [8] 丁硕, 崔总泽, 巫庆辉, 等. 基于 SVPWM 的永磁同步电机矢量控制仿真研究[J]. 国外电子测量技术, 2014, 33(6): 81-85.
- [9] 吴丽华, 高慧芝, 刘政, 等. 一种改进型的 SVPWM 算法研究[J]. 电机与控制学报, 2013, 17(4): 57-63.
- [10] 王瑾. 基于 DSP 和 SVPWM 技术的变频器设计[J]. 电子测量技术, 2009, 31(2): 121-123, 155.
- [11] 张亚楠, 谢冬红. 基于模型设计的嵌入式软件测试技术研究[J]. 电子世界, 2018(9): 66-68.

- [12] 顾卫刚. 手把手教你学 DSP——基于 TMS320X2812[M]. 北京:北京航空航天大学出版社,2011.
- [13] 逢海萍, 郭浩. 基于模型设计的 DSP 电机控制代码开发[J]. 微特电机, 2017, 45(3): 53-56.
- [14] 郭丹, 刘杰. 基于模型设计与传统方式的联合设计开发[J]. 单片机与嵌入式系统应用, 2012, 12(9): 16-19.

- [15] 李满, 江剑. 基于 DSP 的伺服电机驱动器的设计[J]. 机械制造与自动化, 2016, 45(5): 187-191.

作者简介

张世垣,工学硕士,主要研究方向为电力电子技术及其应用、现代交流调速、智能控制等。

E-mail: qkZ121231@163.com

是德科技推出新的 PathWave ADS 先进设计系统 解决方案,加速产品开发工作流程

PathWave Memory Designer 软件缩短设计周期,避免项目延迟

2019 年 2 月 11 日,中国北京——是德科技(NYSE: KEYS)宣布,推出一款全新的 DDR(双倍数据速率)存储器仿真软件——PathWave Memory Designer。作为 PathWave ADS 先进设计系统 2019 的一部分,这款新软件可让开发人员非常方便地对仿真数据和实际测量结果进行比较,从而减少完成产品开发流程所需要的时间。是德科技是一家领先的技术公司,致力于帮助企业、服务提供商和政府客户加速创新,创建一个安全互联的世界。

随着 DDR 存储器不断推陈出新,其设计也变得越来越复杂。同时,这也增加了进行仿真和测试的配置难度,以及相应的设置时间。更高的复杂性意味着关联仿真数据和测试数据存在更大挑战,并且影响设计的置信度,导致更长的故障诊断周期,最终使设计难以如期交付。PathWave ADS Memory Designer 对仿真和测试流程加以整合,同时采用新的工作流程,帮助克服 DDR 存储器设计中的常见挑战。新流程会自动执行常规任务和进行数据分析,从而更快地得到可操作的方案来帮助完善仿真数据。

是德科技副总裁兼设计与测试软件事业部总经理 Todd Cutler 表示:“在整个行业中,提高产品开发周期的效率面临

着重重困难;架起设计与测试之间的桥梁更是如此。Keysight PathWave 平台可以帮助工程师高效地收集、共享和分析测试与测量数据。是德科技的 PathWave ADS Memory Designer 为 DDR 仿真提供了新的工作流程,能够显著缩短设计周期,避免项目延迟。”

Keysight PathWave 是一款功能强大的软件,可以支持敏捷、互联的设计与测试,帮助客户加快工程设计流程,它具有以下特性:

1) 提供开放式 API 和开源组件,使客户能够定制平台,轻松集成当前的硬件和软件;

2) 可扩展的体系结构,能够在专用工作站、公共云或私有云、或者在嵌入式仪器软件上灵活地进行测试计算;

3) 灵活的消费模式,可让用户随时随地根据他们的需要来使用软件;

4) 拥有业内先进的测量技术,可在设计、测试和制造环境中提供始终一致的精确结果。

有关 Keysight Memory Designer 的更多信息,请访问 Memory Designer in PathWave ADS。