Application of Automatic Code Generation Based on PSIM in Digital Control BUCK Circuit

Yang LU1 and Puqiong YANG1*

1 School of Electrical Engineering, University of South China, Hengyang, Hunan, 421001, China

*Corresponding author’s e-mail: yangpuqiong@usc.edu.cn

Abstract. In recent years, digital control has been widely used in the field of power electronics. Automatic code generation, as a graphical software writing method, has brought great convenience to the writing of electronic control software due to a series of advantages such as easy to write and easy to simulate and verify. Simcoder in PSIM is an automatic code generation software that enables the automatic generation of complete DSP control program projects. In this article, the current and voltage loops of a typical average current controlled BUCK circuit are designed as an example, and the use of PSIM and Simcoder is described in detail. Based on the main circuit and Matlab parameter calculations, the model is simulated and the self-generated code is downloaded into the TMS320F28335 to test the output voltage and current of the hardware circuit. The circuit output has the advantage of stable output voltage with small steady-state error and low voltage overshoot, demonstrating the practicality and efficiency of automatic code generation on this project.

1. Introduction

As a graphical software writing method, automatic code generation has been widely used in the field of power electronic digital control in recent years due to a series of advantages such as easy writing and easy simulation verification, which had brought great benefits to the writing of electronic control software. Conveniently, Simcoder in PSIM is a software that can automatically generate a complete set of DSP control program engineering, and can generate product-level readable and fast in execution embedded code.

Reference[1] used graphical programming to establish a 160km/h standard EMU air-conditioning inverter power control strategy model under the environment of automatic code generation, and executed the automatic code generation process; Reference [2] proposed a method based on Simulink The CAN underlying module library design method of automatic code generation technology used S-Function to design and configure CAN channel information modules and CAN message transceiver modules that can import DBC files, edit TLC files, and used RTW (real-time workshop) to implement Simulink Module automatic code generation. Reference [3] is based on the use of DSP’s quadrature encoded pulse (QEP) module and M/T algorithm to design the motor speed detection system, and the detection results were transmitted to the upper computer through the DSP serial port, which realized the real-time display of the motor speed. Reference [4] used automatic code technology to realize the design platform of automobile electronic control system, realized the development of the drive module required by the control system through the Matlab platform, and designed the hardware module of the automobile control system, created the software model of the automobile electronic control system, and used the
fuzzy controller to adjust the PI parameters in the control system. Finally, the designed automobile control system is tested. The test results show that the designed software has certain advantages in efficiency, and it has good resistance and stability in the actual system control process.

This paper first introduces the principle and work flow of automatic code generation based on PSIM, then builds a digital model of average current mode control, introduces the design of the circuit parameters, and introduces the use of PSIM and Simcoder. After software simulation, the Simcoder generated code is downloaded to TMS320F28335 and run. Through experiments on the hardware circuit, the validity of the automatically generated code on the circuit is verified.

2. Principle and process of automatic code generation

2.1 Principle
SimCoder is a tool for automatic code generation with the help of PSIM software. The control circuit can be converted into a digital control program by this tool, and the circuit can be actually replaced by DSP for another simulation, and finally the control program that has been verified by simulation can be simulated. Burn into the DSP chip, and do control and communication through the DSP to verify the correctness of the design circuit and controller.

2.2 Work process
As shown in Figure 1, when using automatic code generation, first build a PSIM model according to the design requirements of the control software, and convert the control loop in the model to a digital control loop. The SimCoder tool in PSIM can convert the digital control loop into a complete DSP project. Complete DSP project, compile the project and burn it into the DSP chip, control the actual hardware circuit, and receive the data from the hardware circuit through the interface of the DSP board.

2.3 Features
Power electronics analysis, design, simulation and physical verification can be performed simultaneously. It is highly efficient. It is mainly reflected in the graphical software design. There is no need to consider the grammatical problems of complex programming languages. Only need to focus on the research and development of algorithms, which can be easily controlled complex control logic and algorithms; efficiency is also reflected in the ability to simulate, through simulation, design vulnerabilities can be found and resolved before the software is deployed, which greatly reduces the time cost of debugging. Under the PSIM, the programming of the program is completed by the way of establishing the hardware circuit and the programming is carried out, so that the developer who does not know the programming of DSP can easily complete the programming of the program, and quickly enter the field of digital control.
3. Build a digital control BUCK converter circuit model

As shown in Figure 2, the BUCK circuit adopts average current-mode control, adopts double closed loop control of voltage loop and current loop. The outer loop is a voltage loop to adjust the voltage error and generate the current command of the current loop, which is different from the inductor current to generate PWM generation. The main parameters of the BUCK circuit are: the input voltage is 50V, the output voltage is 24V, the input capacitance is two 100uF/250V electrolytic capacitors in parallel, the inductance is 365uH, and the output capacitance is three 100uF/250V electrolytic capacitors. The capacitors are connected in parallel, and the switching frequency is 40kHz.

![Figure 2. Digital controls the basic topology of the BUCK converter](image)

According to the control requirements of the converter, use PSIM tools to build a control model, including a control loop based on average current and a soft start circuit.

3.1 Average current-mode control loop parameter calculation

3.1.1 Current loop design

The average current loop design is shown in the figure. Using the state average method, as shown in figure 3 above, we can get

\[ L \frac{dIL}{dt} = dVin - Vo \]  \hspace{1cm} (1)

Ignoring the changes of Vo and Vin, we can get from

\[ I = \frac{Vin}{sL} \]  \hspace{1cm} (2)

Considering the current sensing ratio and the gain of PWM, we can get

\[ H(s) = \frac{IL(s)}{Vo} = \frac{RiVin}{sLVo} \]  \hspace{1cm} (3)

Aiming at the first-order system, the current error amplifier (GCA) can be designed using the second-class error amplifier method as shown in Figure 4.

![Figure 3. Average current control chart](image)
Since the PWM control voltage can only intersect with its sawtooth wave signal once in a cycle, the maximum bandwidth \( w_{co} \) of the current loop is limited by the rising slope of \( V_{con} \) less than the rising slope of the PWM sawtooth wave \( (V_t) \). The rising slope of \( V_{con} \) can be sensed. The measured inductance current falling slope is determined by GCA amplification, and can be obtained from the above limitation

\[
\left( \frac{\Delta V}{L} \right) R_i G_{CA_{max}} (w_{co}) = Vtf_i
\]  

(4)

After finishing formula (4), we can get

\[
G_{CA_{max}} (w_{co}) = \frac{V_{con}}{R_i L} = \frac{Vtf_i L}{VoR_i}
\]  

(5)

From formulas (3) and (5) and use \( G_{CA_{max}} (w_{co}) H_i (w_{co}) = 1 \) we can get

\[
\frac{V_{of} L}{VoR_i} \frac{R_i V_i}{w_{co} L V_t} = 1
\]  

(6)

Formula (6) can be rearranged to get

\[
w_{co_{max}} = \frac{V_{inf} f_i}{w_{co}} \text{ (rad/s)}
\]  

(7)

\[
f_{co_{max}} = \frac{f_i}{2\pi D}
\]  

(8)

From equation (8), it can be seen that if the design is designed with the rising slope of the control voltage \( V_{con} \), the theoretical maximum current loop bandwidth may be higher or too close to the switching frequency, so it is impossible to set this value, generally limited to the noise ratio, bandwidth. The selection of \( w_{co} \) can be set at 1/4~1/8 of the switching frequency. Once the bandwidth \( w_{co} \) is selected, the K-factor method can be used to make the second type of error amplifier \( z = w_{co} / K \), \( p = w_{co} / K \).

3.1.2 Voltage loop design

Generally, the response speed of the voltage loop is much lower than that of the current loop. Therefore, when the voltage loop is modeled, the current loop can be regarded as ideal, that is, the response of the sensed inductor current and its command is regarded as 1, namely

\[
\frac{R_i L}{Ic(s)} = 1
\]  

(9)

\[
\frac{\Delta V}{IL} = R \frac{1 + s}{w_{co}}, w_z = \frac{1}{CR_i}, w_p = \frac{1}{CR}
\]  

(10)

Based on this assumption, the equivalent circuit of the voltage loop of the average current-mode control can be simplified as shown in Figure 5. From Figure 5, we can get
Using equations (9) and (10), the control flow chart of the voltage loop can be drawn as shown in Figure 6, and the voltage error difference amplifier (GEA) can also be designed using the aforementioned two types of error amplifiers as shown in Figure 7. The bandwidth of the voltage loop can be designed to be 1/3~1/5 of the bandwidth of the current loop.

According to the above-mentioned current loop voltage loop model and error amplifier design method, using MATLAB to write a program, the Bode plot of average current control is obtained, as shown in Figure 8 is the Bode plot of the current loop, Figure 9 is the Bode plot of the voltage loop, and Figure 8 is the bode plot of current loop.
3.2 PSIM average current-mode control circuit design and simulation

Based on the above design, draw the PSIM analog circuit, as shown in Figure 10. The voltage and current waveforms are shown in Figure 11. The load has a sudden change at 0.04s. As shown in the figure, the output voltage also changes when the load jumps, but it quickly responds to the set value and remains stable.
4. Automatic code generation and computer testing
In order to verify the function of the generated code, test on the hardware BUCK circuit, and connect the BUCK circuit hardware structure to the DSP c2000 board card series. Figures 12 and 13 show the BUCK hardware topology and the DSP digital control hardware circuit.

[Image: BUCK hardware topology, DSP digital control hardware circuit]

Download the automatically generated code to the DSP control board. Since the resistance load is used at this time, the load jump is not set, and the voltage and current waveforms of the oscilloscope output corresponding to the simulation are obtained, as shown in Figure 14. The output voltage and current value, as shown in Figure 15.

[Image: Voltage current output waveform, output voltage and current value]

Using the automatic code generation method, software functions can be debugged in the software simulation link, and most software problems can be corrected, which greatly shortens the time for prototype debugging. In addition, graphical programming improves the operability and controllability of the realization of software functions. While more effective reducing the loopholes in the function design, the software writing time is greatly reduced. For general electronic control software with functions such as working sequence, fault diagnosis, closed-loop modulation, communication, data storage, etc., from building the software according to design requirements to starting prototype testing after software deployment, it usually takes 1-2 weeks, which is more efficient than handwritten software significantly improved.

5. Conclusion
This paper explains the basic principle of using PSIM to realize automatic code generation. Taking the BUCK circuit with average current-mode control as an example, it introduces the construction of PSIM software model, model simulation and software deployment, and shows how automatic code generation is used in power control software development. In the application process. After hardware physical testing, the control software function realized by automatic code generation is the same as the simulation result, which can greatly shorten the software development time and improve the development quality.

Acknowledgments
This work was supported by the National Magnetic Confinement Fusion Energy Development Research Project (Grant No.2019YFE03070000), the National Natural Science Foundation of China (Grant No.11675074), and Hengyang Science and Technology Plan Project (Grant No.202002042159).

References
[1] Hou Y.C., Yang D.J., Zhen Y.W., Zhao X.Q., Shen L., Shen Z.Z. (2021) Application of automatic code generation in train air conditioning power supply software [J]. Electrical Drive, 51(12): 41-46 (in Chinese).
[2] Wang W., Mo G.X., Shen J., Xie Y.B., Wang W.M. (2020) The design of CAN underlying module library based on Simulink automatic code generation technology [J]. Control and Information Technology, (03): 93-96 (in Chinese).

[3] Cui Y.X., Chang Y.J., Zhang Q., Zheng J. (2019) Realization of speed measurement algorithm based on automatic generation of DSP code [J]. Journal of Guilin Institute of Aeronautics and Astronautics, 24(04): 3-7 (in Chinese).

[4] Gang J.Z., Li Y.P., Wen G.R. (2019) Automobile control system design based on automatic code generation technology [J]. Electronic Design Engineering, 27(18): 166-169 (in Chinese).