A Method Based on Interpreter for Cyclic Voltage Waveform Generation

Guifu Yang & Wei Wang
School of Information Science and Technology, Northeast Normal University, Changchun, Jilin, China

Zhenbang Liu & Yu Bao
Changchun Institute of Applied Chemistry Chinese Academy of Sciences, Changchun, Jilin, China

Yiran Guan*
School of Information Science and Technology, Northeast Normal University, Changchun, Jilin, China

ABSTRACT: In order to meet the demand of generating cyclic voltage waveform as an excitation signal acting on the electrode in electrochemical experiment, a kind of new method of software design based on interpreter is proposed. In this method, the host computer uses the inner DSL to abstract the control instructions of the electrochemical experiment into interpreter source code, and the slave computer adopts hardware timer to interrupt main procedure, and then trigger regularly the interpreter to read, parse and execute the interpreter source transmitted by the host computer, which generates the cyclic voltage waveform. The waveform generation method based on interpreter realizes cyclic voltage waveform much easier. It replaces the traditional waveform generator, makes the operating much simpler and reduces the cost of the system. Electrochemical experiments test results show that the design can complete the generation of the cyclic voltage waveform of more than 30 electrochemical methods.

Keywords: electrochemical instrument; waveform generation; interpreter; instruction sequence

1 INTRODUCTION

As one of main branches of chemistry, electrochemistry ties chemical change and electricity phenomenon closely, and analyzes the experiment process and result by using the basic principle and experiment techniques of electrochemistry, which forms various kinds of electrochemical methods.

Electrochemical methods [1] mainly use electrical signals as excitation and detection means, produce chemical reaction through the generated voltage waveform acting on the electrode resulting in changes in physical and chemical properties, and make qualitative or quantitative analysis of substance. A device used to measure above changes is called electrochemical instrument.

The research and development of electrochemical instrument plays a decisive role in the development of electrochemistry. The generation of voltage waveform in electrochemical instruments ensures smooth test of electrochemical method and correct experimental results, so designing a reasonable and effective voltage waveform generation method is essential in the instruction system of the slave computer. In this thesis, a method based on interpreter for generating cyclic voltage waveform is designed and implemented.

2 RESEARCH REVIEW

About the waveform generating, there are two main classes of methods: one is based on hardware; the other one is based on software. The waveform generated based on hardware [2-7] can only be applied to a small number of electrochemical methods. Moreover, it has complex circuit structure, poor flexibility, and troublesome real-time adjustment.

In comparison, the method based on software solves above disadvantages. At present, a variety of methods for SPWM [8] waveform are proposed.

Natural sampling method [8-9] uses sine wave as modulation wave, and triangle wave as carrier wave,
and then controls the turn-on and turn-off of switching device at the natural intersection point of the two waveforms. However, this method needs to solve a transcendental equation by a great mount of computation, which is also difficult to control in real-time. Regular sampling method \(^\text{[8-9]}\) is derived from the natural sampling method, which takes sample of sine wave using triangle wave to get step wave, and then controls the on-off of the switching device at the intersection point of the step wave and the triangle wave. However, it has poor precision and smaller linear control range. SPWM area equivalent \(^\text{[8-10]}\) divides a semi-sine wave into N parts, then using equal height rectangular pulse with equal area instead of the area enclosed by each sine line and the transverse axis, but its disadvantage is that the calculation accuracy is insufficient. SHEPWM \(^\text{[9-11]}\) can control the output voltage component and selectively eliminate certain lower harmonics by reasonably arranging the position and width of the notch which set on the voltage waveform, but its drawback is that the calculation is complex and can only eliminate harmonics of a certain number of times. In HISPWM \(^\text{[11]}\) algorithm, a certain component of triple-harmonic is incorporated into a three phase SPWM artificially by using the ability of line voltage to automatically eliminate the 3K harmonic in phase voltage, so that the peaks of the synthetic modulated wave are reduced in order to avoid over modulation within certain range.

The above methods are all aimed at SPWM wave, where there is a common problem that the methods can only generate sine wave instead of triangle wave, step wave and other waveform.

In electrochemistry, the principle \(^\text{[12]}\) of measuring voltage and current is that a certain excitation signal acts on the electrode of the electrochemical cell in order to produce electrochemical reaction of the substance to be tested on an electrode, then those electrochemical experts can study the relationship between voltage and current in various excitation waveform by measuring of voltage and current in order to study the chemical properties of the substance to be tested.

![Figure 1. Waveform of voltage.](image)

In electrochemical experiments, there are many types of excitation waveform required to act on the electrodes, such as triangle wave, square wave, and trapezoidal wave. As shown in Figure 1, the potential acting on the electrode is composed of tiny voltages which are increased at a fixed time interval. During the scanning, the current is detected at a fixed time interval on each potential step, and meanwhile the relation curve of voltage and current is measured. As excitation waveform, the experiment in Figure 1 requires a dual-cycle in the waveform. In the figure, A represents a small waveform cycle, B represents a large one, and then A and B together form a dual-cycle in the waveform. B consists of multiple A, and the entire waveform consists of multiple B.

For the excitation waveform in electrochemical experiments which is similar to Figure 1, the current waveform generation methods based on software are not available yet. Therefore, we proposed a new method of waveform software generation based on interpreter \(^\text{[13-14]}\). Using a variable-length array to store instruction sequences, IP as array index, SFR as I/O, the method uses nesting of three-layer register and JNZ to realize the generation of cyclic voltage waveform.

Compared to existing methods, the new method proposed in this paper can generate various types of waveform, which has high reusability, simple implementation and low cost. It can generate various voltage waveform required in more than 30 kinds of electrochemical experiments, which not only adjusts the voltage, but also detects the current. The method can meet the demand for generating cyclic voltage waveform as excitation signal acting on the electrode well.

### 3 WORKFLOW OF INTERPRETER

By analyzing the voltage waveform generation process of more than 30 kinds of electrochemical methods, we found that the instructions which control waveform generation are different, but the processing flow is the same. Therefore, in the slave computer, an interpreter is designed and implemented by using C language to control the generation of voltage waveform.

The host computer abstracts the processing flow which controls voltage waveform generation into inner DSL. \(^\text{[15-17]}\) source, then the interpreter uses the inner DSL source as input, and waits for the hardware timer to trigger it regularly. When the interpreter is triggered, the interpreter engine begins to parse and execute the source code that conforms to the DSL syntax rules, and generates the corresponding cyclic voltage waveform.

### 4 STRUCTURE OF INTERPRETER

The interpreter designed in this paper is composed of interpreter source code and interpreter engine. Its structure is shown in Figure 2.

Interpreter source code is a variable-length array expressed by C language, and the array is used to store the instruction sequences which control waveform
4.1 Source of interpreter source code

In order to improve the extensibility of the product, reduce the repeated work in the software development process, and improve the production efficiency and accuracy of the host computer software, we analyzed the variability and commonness of more than 30 kinds of electrochemical methods, and abstracted low-level duplicate code into DSL sources which conform to the DSL syntax rules. This is the origin of the interpreter source code.

4.2 Interpreter engine API

As the number of instruction sequences for controlling waveform generation of more than 30 kinds of electrochemical methods is different, every instruction consists of an operator and a number of operands, and the instructions are different in length. Therefore, we use the variable-length array, which can dynamically allocate memory according to need, to store instruction sequences, which not only saves space, but also simplifies operation. When defining the array, we use the register IP as the index of the variable-length array, and store all instruction sequences by IP’s self-increment. Meanwhile, IP represents the storage address where the array element is located.

As shown in Figure 3, the first element of the array stores operator1 of instruction1, and the array index IP is 0, which represents the storage address of operator1; the second element of the array stores operand11 of instruction1, and the array index IP increases to 1, which represents the storage address of operand11, and so on.

4.3 Structure of interpreter engine

Interpreter engine is mainly composed of a register set\textsuperscript{[18]}, an instruction set\textsuperscript{[18-20]} and a number of instruction sequences\textsuperscript{[18-20]}. The register set includes special function register (SFR) which is used as I/O, instruction register IP which identified the storage location of the instruction, register CX, register DX and register EX which is used as loop counters; the instruction set mainly supports the non-zero transfer instruction JNZ, the input and output instruction I/O, the interpreter termination instruction SHUTDOWN, the counter decrement instruction and the interpreter timing interrupt instruction HALT; the instruction sequences consists of a series of instructions generated by control waveform, and addressing mode of instruction used skip addressing.

4.3.1 Register set

In this paper, the register set used by the interpreter engine mainly includes instruction register, counter register and special function register.

Instruction register IP is used to identify the storage location of the instruction through array index. The instruction sequence for controlling waveform generation is stored in a variable-length array, and the value of the array index IP corresponds to the storage location of the current instruction.

The special function register SFR is used as the interpreter’s I/O. The host computer will input the electrochemical experiment parameters as the source of the interpreter to the interpreter through SFR, and then the output waveform will be output to the drive control circuit of hardware instrument after the interpreter engine executes.

Register CX, Register DX and Register EX are put together to implement the control of the triple cycle. CX serves as the innermost loop counter of the three loop, the initial value of CX is used as the upper bound of the first loop, and the innermost loop is implemented by self-subtraction of CX; DX serves as the second loop counter of the three loop, the initial value of DX is used as the upper bound of the second loop, and the second loop is realized by self-subtraction of DX, and so on. CX is nested within the loop of DX, and DX is nested within the loop of EX, and implements the control of the triple cycle together.

4.3.2 Instruction set

The instruction set used by the interpreter engine mainly has conditional transfer instruction, input and
output instructions, decrement instruction counter, SHUTDOWN, HALT and so on. The function of instruction set is shown in Table 1.

| Name     | Type             | Function Description                                                                                                                                 |
|----------|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| JNZ      | Condition branch | Format: JNZ A, LOL; A is not equal to 0, then jump to LOL. Ending the execution of interpreter.                                                     |
| SHUTDOWN | Close instruction | Responsible for entering the DSL source code which control generated voltage control generated voltage waveform into the interpreter.             |
| IN       | Close instruction | Input instruction                                                                                                                                     |
| OUT      | Output instruction| Responsible for Output the experimental data to the I/O port.                                                                                     |
| DEC      | Decrement        | Decrement Instruction                                                                                                                                   |
| HALT     | Interrupt        | Making the interpreter into suspended state, and waiting for the hardware timer to trigger again.                                                    |

4.3.3 Timing of instruction

Electrochemical experiment requires not only the accurate setting of experimental parameters, but also the accurate timing. Because the software system is not precise enough for timing control, in the interpreter we use hardware clock to interrupt control in order to implement accurate timing and the timing interval is obtained by prior knowledge. According to the fixed time interval which is set by prior knowledge, the timer triggers interrupts periodically, the main program calls the interrupt handler - waveform generation interpreter, and then the interpreter completes within a fixed time interval, and waits until the next interrupt of the timer.

4.4 Implementation of interpreter engine

The execution of interpreter engine is the mapping of the interpreter engine structure to the specific implementation, and is the process of instruction set together with the register set to complete the execution of the instruction sequences. The concrete implementation of the interpreter uses C language, by adopting a variable-length array of C language to store the instruction sequences, and mainly using four mechanisms to support the traversal and execution of instruction sequences.

1) OPERATER OPRAND*

This is the format of instruction defined in our interpreter. Each instruction consists of an operator and a number of operands. The operands of each instruction are optional, and meanwhile, the length of each instruction is indeterminate.

2) While(1)

While(1) statement ensures that executing sequences of instructions controls the waveform generation cyclically in the variable-length array. Each instruction is executed and jumps to the next instruction by ++IP or address.

3) switch(arr[IP]) ... case: IP=IP+len(1+OPRAND); break

Switch statement is used to parse each instruction, matching the operator of the instruction, and then executes the instruction corresponding to the operator. For example, as the operator matches with “HALT”, the interpreter stops running and goes into the suspended state, waiting for the hardware timer to trigger the interpreter again.

4) IP = IP + OPRAND

This statement implements the jump of IP by non-zero conditional transfer instruction JNZ. In this statement, the OPRAND is the operand that gets from the instruction, which represents the address offset for the next instruction.

The specific implementation process is shown in Figure 4. The IP address of the current execution instruction is 1020, when switching (arr[IP]) to match the transfer instruction JNZ, and gets the operand value OPRAND corresponding to this instruction. If the OPRAND is -15, then the statement jumps to the address label 1005 corresponding to IP = IP + OPRAND, and continues to execute the next instruction.

5 EXPERIMENTAL RESULTS

In order to verify the correctness of the method proposed in this paper, it was tested on three electrochemical instruments which were developed through collaborating with Changchun Institute of Applied Chemistry of Chinese Academy of Sciences. We carried out 80 electrochemical experiments and proved that this method can generate exciting waveform according to different electrochemical experiments with good results.
Figure 5 is a diagram of cyclic voltammetry which is generated by our electrochemical test. The diagram records the relationship between the current obtained from the working electrode and the applied voltage which is generated by the method designed in this paper, and the detection time of the current strength is also determined by it.

6 CONCLUSIONS

A new method based on interpreter is presented for generating waveform in this paper, which is very simple because it only uses 654 lines of code in C language to realize the waveform generation. In addition, it has the advantages of high reusability, simple implementation, flexibility and ease to use. The effectiveness of the proposed method has been verified through experimentation, which can be used as a replacement for traditional waveform generator, and can also provide reference for other projects that need to generate various types of waveform.

ACKNOWLEDGEMENT

The authors gratefully acknowledge financial support from NSFC (21527806) and the Science and Technology Development Planning of Jilin Province (20160204043GX, 20130206097SF). Moreover, many thanks to the staffs in Changchun Institute of Applied Chemistry of Chinese Academy of Sciences for providing us a lot of help, so that our work can be carried out smoothly.

REFERENCES

[1] Allen J. Bard, Larry R. Faulkner. 2000. Electrochemical Methods: Fundamentals and Applications, Wiley.
[2] Ranjan R K, Mazumdar K, Pal R, et al. 2017. Generation of square and triangular wave with independently controllable frequency and amplitude using OTAs only and its application in PWM. Analog Integrated Circuits & Signal Processing, 92(1): 1-13.
[3] Garcia E O, Cumplido R, Arias M. 2008. Microcontroller-based function generator. Al-Khwarizmi Engineering Journal, 4(1): 48-57.
[4] Orino Y, Kurosawa MK, Katagiri T. 2005. Direct-digital synthesis using delta-sigma modulated signals. Circuits and Systems, 2005 IEEE International Symposium on. ISCAS2005. pp: 1706-1709.
[5] Garcia E O, Cumplido R, Arias M. 2006. Pipelined CORDIC design on FPGA for a digital sine and cosine waves generator. Electrical and Electronics Engineering, 2006 3rd International Conference on. pp: 1-4.
[6] Yu Hongyu, Li Zhi, Liu Chang. 2009. Research and implementation of SCPI interpreter for function/arbitrary waveform generator. Foreign Electronic Measurement Technology, (4): 70-73.
[7] Wang Xin. 2013. Waveform design based on DDS. Modern Electronics Technique, 36(14): 1-3.
[8] Zhang Fang. 2008. Study on SPWM algorithm technique. Fire Control Radar Technology, (4): 105-107.
[9] Xiong Junhua, Wang Tingling, Chen Jianming, Sun Biao. 2008. Analysis and realization of three algorithms for SPWM waves. Microcomputer Information, 24(19): 307-309.
[10] Wang Q, Gao T. 2012. Research of SPWM algorithm based on equivalent area compensation. Advanced Materials Research, 562-564: 2174-2177.
[11] Tuncer S, Tatar Y. 2005. An application of SHEPWM technique in a cascade multilevel inverter. COMPEL, 24(1): 81-93.
[12] Bertolini L, Elsener B, Pedeverfi P, et al. 2013. Electrochemical Techniques. Corrosion of Steel in Concrete: Prevention, Diagnosis, Repair (2nd Edition). Wiley-VCH Verlag GmbH & Co. KGaA, pp: 365-406.
[13] Abo A V, Lam M S, Sethi R, et al. 2007. Compilers: Principles, Techniques, Tools, Addison-Wesley.
[14] Harold Abelson, Gerald Jay Sussman, Julie Sussman. 1996. Structure and Interpretation of Computer Programs (2nd Edition), MIT Press.
[15] Martin Fowler. 2010. Domain-Specific Languages. Addison-Wesley Professional.
[16] Kosar T, Mernik M, Carver J C. 2012. Program comprehension of domain-specific and general-purpose languages: comparison using a family of experiments. Empirical Software Engineering, 17(3): 276-304.
[17] Kosar T, Mernik M, Gray J, et al. 2014. Debugging measurement systems using a domain-specific modeling language. Computers in Industry, 65(4): 622-635.
[18] Kip R. Irvine. 2014. Assembly Language for x86 Processors. Prentice Hall.
[19] Bryant, RandallE. 2016. Computer Systems: A Programmer's Perspective (3rd Edition), Pearson.
[20] Bryant R E, O'Hallaron D R. 2001. Introducing computer systems from a programmer's perspective. SIGCSE Technical Symposium on Computer Science Education. ACM, pp: 90-94.