Design and Implementation of CNC System for WEDM Machine Tool Based on Active Object

Chen YuanLong, Wang ZhuangZhuang, Dong BoLin

Hefei university of technology institute of mechanical engineering, Anhui Province, China
chenyuanlong@sina.com,18895497877@163.com, 936740595@qq.com

Abstract. A software framework design method for embedded real-time control system is proposed for the conversion of internal behavior state of CNC machine tool control system. This method uses the active object calculation model to plan and model the internal state of the linear cutting CNC system. On this basis, an open machine tool control system is built, and the design of the linear cutting machine bed control system is realized. The system design realizes the middle layer of the system based on a lightweight framework, and truly realizes the separation of the hardware and software of the control system. Each object is independent of each other and does not share resources. Information events are distributed through the "Publication-Subscription" method, and the line is designed to be cut. It can solve the problem of poor reconstruction ability of the original system and improve the portability of the system.

1. Introduction

WEDM is a kind of special processing, which is developed on the basis of EDM. After more than 70 years of development, WEDM has been widely used, accounting for more than 60% of the domestic market of EDM machine tools. Now WEDM has become an indispensable part of manufacturing technology, and is developing towards intellectualization, high-speed and openness[1].

At present, the mainstream in the market is the integrated system of programming and control such as HF, which initially realizes the integrated control from design to processing[2-3]. However, the research on the control system of WEDM machine tool is still in its infancy. Based on DOS and WIN98, HF and other editing and control systems depend on the system environment and drawing plug-ins[4]. They have high cost, high power consumption, unfriendly human-computer interaction interface, and their dependence on external plug-ins also limits the improvement of reconfiguration. CNC wire cutting machine tool system has various modules, complex working behavior, high technical requirements of control system, and the updating of hardware system will certainly change the system behavior logic[5]. When part of the machine tool control system changes, the overall control logic and behavior state will change greatly, so a higher requirement for CNC system is put forward[6]. However, the current programmable control system relying on external implementation obviously cannot meet this requirement.

In order to improve the reconfigurability of CNC system, a design of CNC system for WEDM machine tool based on active object computing model is proposed. The integration of embedded platform and machine tool control system greatly reduces the hardware cost of the system. At the same time, the idea of state machine is introduced into the software design of machine tool numerical control system, which reduces the coupling between the modules of the system and realizes the reconfiguration of the system.
2. Design of Active Object-based Computing Model

Active object computing model is the product of the combination of event-driven framework technology and hierarchical state machine. Each active object has its own control threads and realizes a more meaningful encapsulation\(^7\). As the most effective technology of event-driven system, active objects use the behavior of Run to Complete (RTC) to process events, and asynchronous communication is used between different objects. Figure 1 shows a streamlined process of running an active object, which includes a state machine, an event processing thread, and an event queue.

![Active Object Computing Model](image)

Figure 1. Active Object Computing Model

2.1. Hierarchical state machine

Hierarchical state machine is based on finite state machine, so the concept of finite state machine is introduced first. Finite State Machine(FSM) is a mathematical model for describing the state of a system and its transfer and action between different states. The system shows different working states at different stages. These states are not overlapping and the number of states is limited. At any time, the system must be in a certain state. In the current state of the system, when an input event occurs, the system will respond to the event according to the current state and the input of the system, resulting in a definite output behavior, and may accompany the state migration. FSM is formally defined as follows:

\[
M = M(A, S, Z, S_0, f, g)
\]  

In Formula:

- \(A\) —— Finite Set of Input Characters
- \(S\) —— A Finite Set of Internal States
- \(Z\) —— A Finite Set of Output Characters
- \(S_0\) —— Initial State in \(S\)
- \(f\) —— State transition function from \(SXA\) to \(S\)
- \(g\) —— Output function from \(SXA\) to \(Z\)

The classification of state machines has different rules. From the point of view of practical application, the state machine is divided into Moore type and Miller type according to whether the output and input conditions of the state machine are related. The output of Moore state machine is only related to the current state, while Mealy state machine is not only related to the current state, but also depends on the system input condition\(^8\).
Finite state machine can solve small-scale problems better, but when the target system is large and the interaction between modules is complex, the explosive and complex characteristics of finite state machine itself will be amplified. The system has a variety of modules and complex behavior, and the finite state machine (FSM) obviously cannot meet the requirements, so the hierarchical state machine (HSM) is introduced.

Hierarchical state machine uses behavior reuse to reduce repetitiveness. The core point is to process events in a hierarchical way. The system is divided into two levels: high and low. Low-level state machine processes specific events. It is the behavior subject of the system. When low-level state machine cannot process an event, the system will call more. A higher level state machine handles the event.

2.2. Event Asynchronous Delivery Mechanism

Event dispatching mechanism undertakes the task of event communication among different objects, which can be said to be the communication core of lightweight framework. Inter-object event information flow can be divided into two modes: direct distribution and "publishing-subscribing" mode with middle-level mode. Event direct dispatch requires full understanding of consumers among active objects. This dispatch method requires strong coupling among components of the system, which reduces the flexibility of the system.

In this system, the "publishing-subscribing" mode is built by setting up the middle layer, which plays the role of communication bus and solves the coupling problem caused by direct event dispatch. Consumers and producers in the system do not need to know the existence of each other, or even the type of events that consumers need. Event dispatching tasks are undertaken by the middle layer. After producers produce events, they send them to the middle layer. The middle layer judges the objects needed and sends them to the target objects according to the events themselves. Event dispatching mechanism is shown in Figure 2.

![Figure 2. Event Asynchronous Delivery Mechanism](image)

3. Software Design and Elements

The software design of control system includes the implementation of hierarchical state machine planning, the design of man-machine interaction interface, parameter acquisition, input, processing, instruction transmission and so on. Because the WEDM machine tool passes 3B code instruction, the system realizes 3B decoder. STM32F407 chip is selected to use Cortex-M4 core, working frequency 168MHz, with enhanced DSP processing capacity and extreme speed, while the interface is more abundant. The system extends 16M SPI FLASH chip and mounts 16G SD card by SOID driver. It is used for interface display and system information access and expands 1M byte SRAM. Based on this hardware, the system develops drivers and transplants UCOS III system and EMWIN graphical interface library. The software structure of the system is shown in Figure 3.
Because UCOS III system has no file access module, FATFS file system is chosen to improve file access efficiency. The file system is written entirely in C language. The hardware platform is independent and the file storage efficiency is high. FATFS file system provides users with a series of application interface functions, such as f_open, f_close, f_read, f_write and so on. It provides users with convenient file reading function[9].

3.1. Hierarchical State Machine Task Planning
State machine planning focuses on classifying system units according to their working status. Firstly, the mechanism and working principle of the machine tool are analyzed. The WEDM machine tool is composed of three parts: electrical system, mechanical system and working fluid system. The mechanical system includes bed, motor, wire conveying mechanism, worktable and lubrication system, which is the basis of the whole machine tool. The electrical system includes control circuit, pulse power supply, machine tool circuit and so on. The working fluid system is used to cool electrode wires and workpieces and remove electro-corrosion products. It is composed of working fluid pump, flow control valve, filter core, working liquid box, etc[10].

According to the working process of machine tools, the system is divided into hand dynamic S0, initial state S1, automatic state S2 and stop state S3. The machine tool is in the manual dynamic S0 after starting. Under S0, the system can accept the manual control of motor movement, control the direction of electrode wire operation, and complete the tool setting function of the machine tool. After the tool setting is completed, the machine tool enters the initial state S1. In the initial state, the system determines the processing trajectory according to the input requirements, including single processing, multiple processing and taper processing trajectory. Processing trajectory includes setting parameters of electrode wire cut-in point, etc. The trajectory is generated and then transferred to the auto-running state S2, which includes motor, parameter setting of coolant system and processing function. After processing, the electrode wire returns to the original point, and then the system enters the stop state S3. If the workpiece needs to be replaced, the system enters the S0 state; otherwise, it enters the initial state S1.

3.2. Lexical analyzer and technical points
The function of lexical analyzer is to read in the character stream of the source program and make them into meaningful morphemes and output lexical units in a certain sequence. From the point of view of software logic, the lexical analyzer provides the unique interface Get_token() for morpheme acquisition to the upper processing process. The process of morpheme acquisition is encapsulated so that the upper software does not care about the specific process of lexical analysis[11]. Because of the limited hardware resources of embedded system, it takes a lot of time to process a large number of characters in the process of code conversion. In order to reduce the time overhead of processing single line code, a double pointer buffer scheme is adopted in the lexical analyzer, which is shown in Figure 4.

Figure 3. Software structure of control system
The capacity of each buffer is determined. In this paper, the buffer is set to 30 characters. Each time a line of instructions is read in, the system reads all one line of instructions into the buffer at a time. EOF is a special character that indicates the end of the source file. In order to save time for word segmentation, the program itself maintains two pointers, lexemeBegin and forward. LexemeBegin pointer: It points to the beginning of a morpheme and needs to determine the end of the morpheme. Forward pointer: It scans forward and increments bit by bit until it finds the complete morpheme position that matches a pattern. The lexemeBegin and forward pointers point to the starting position of the next morpheme to prepare for the reading of the next morpheme. When the lexemeBegin pointer points to the EOF character, it indicates that the current buffer content has been processed. The specific implementation procedures are as follows:

```c
char* get_token(char* des)
{
    //Internal memory block selection
    u8 sramx=0;
    //Double Pointer Buffer Scheme
    char* pBegin = des;
    char* forward = des;
    //Characters end with "\r\n"
    while( *forward != '\r' )
        forward++;
    // Length of a single morpheme
    int len = forward - pBegin;
    //Request storage space
    char *t = mymalloc(sramx ,len +1);
    mymemcpy(t ,pBegin ,len ); // Storage
    t[len] = '\0'; //Add 0 at the end
    return t;
}
```

4. Realization of WEDM Control System

4.1. Hardware Architecture of WEDM System
The WEDM CNC system is an embedded CNC system based on UCOS III 3.04. The system uses EMWIN graphical interface library to build human-computer interaction interface, which greatly improves the user's experience. The machine tool control system uses ARM+DSP as the main multi-CPU structure[12]. The STM32F407 chip on ARM board is the main CPU, which is responsible for information acquisition and processing, human-computer interaction and system management. The ARM board is connected with the DSP through PC/104 industrial bus, i.e. the four-axis motion control
The upper computer uses UCOS III operating system for multi-task management, while the lower computer uses four-axis motion control card for tasks such as motion control.

Figure 5. Hardware structure of control system

4.2. Software and Hardware Platform of WEDM System

According to the theory of state machine, the working state transition diagram of machine tool is planned, and the control system software of WEDM machine tool is developed on the basis of it. The control system adopts the structure of master and multi-CPU to realize machine tool control. The system has friendly man-machine interface, low overall cost and high reconfigurability. Figure 6-7 is the physical diagram of the software and hardware platform of the control system; Fig. 9 is the main board of the control system and the carrier of the system; Fig. 10 is the test screen of the decoder, and the input 3B instruction is converted into G code instruction, which is being tested and run; Fig. 11 is the monitoring interface of the machine tool operation parameters.

Figure 6. Control system motherboard

Figure 7. Machining parameter monitoring interface of machine tool
5. Concluding remarks
In this paper, the active object computing model is used in the development of WEDM control system. The software framework of the control system for CNC WEDM machine tool is built, and each module of the system is realized.

(1) The state machine is used to exchange events and information between machine tools in order to achieve the purpose of state migration, and the coupling degree between the modules of the system is low. When new modules need to be added, state diagrams and state migration conditions can be modified accordingly, without large-scale adaptation or rewriting of the system, which improves the reconfigurability of the system.

(2) The task planning of the system is clear, and the structure of each active object is clear, which avoids a large number of repetitive functional programming in the process of system development and improves the portability of the system.

References
[1] Zhu Hongan. Development Trend of WEDM in China [J]. Science and Technology Information, 2010 (17): 139.
[2] Bi Lei. Design and Implementation of CNC System for Wire Cutting Machine [D]. Anhui: Hefei University of Technology, 2010:1-66.
[3] Zhu Jinming, Cui Xuehong, Meng Jieli. Teaching reform of NC WEDM programming and control technology under the background of new subject construction [J]. Journal of Guangxi University for Nationalities (Natural Science Edition), 2018, 24 (3): 107-112.
[4] Chen Tingcheng, Zhang Shuyuan, Chen Shouqiang, et al. Windows-based automatic programming system for CNC WEDM drawing [J]. Journal of Xihua University (Natural Science Edition), 2014, (1): 8-11.
[5] HITOS, WASHBURNJ, WANGS, et al. Specifying reconfigurable control flow for open architecture controller [J]. Proceedings 1998 Japan-USA Symposium on Flexible Automation, 1998, 7 (2): 659-666.
[6] Yang Zhibang, Chen Delin. Modular design of reconfigurable machine tool control [J]. Manufacturing Automation, 2013, 35 (10): 149-151+156.
[7] Zhu Bing, Mei Hong, Yang Fuqing. Event-driven Active Object Model [J]. Journal of Software, 1996 (03): 145-149.
[8] Kevin Ferland. Discrete Mathematics, Second Edition [M]. Taylor and Francis: 2017:04-12.
[9] Transplantation of Zhang Tao, Zuo Jingping, Ma Hualing. FatFs on 32-bit microcontroller STM32 [J]. Electronic Technology, 2010, 47 (03): 25-27.
[10] Chen Xiuzhen. Structural design of WEDM machine tools based on modular design theory [J]. Modular machine tools and automatic processing technology, 2009 (07): 103-105+109.
[11] Kartik Gupta, V. Krishna Nandivada. Lexical state analyzer for JavaCC grammars [J]. Software: PracticeAnd Experience, 2016, 46 (6).
[12] Dong Bolin, Ju Yi. Research on 6-DOF Robot Control System Based on Quantum Framework [J]. Modular Machine Tool and Automated Machining Technology, 2015 (12): 85-89.
[13] Dong Haitao, Zhuang Shujun, Chen Bing, Chen Youping. A reconfigurable CNC system based on ARM+DSP+FPGA [J]. Journal of Huazhong University of Science and Technology (Natural Science Edition), 2012, 40 (08): 82-87.