2010 International Conference on Measuring Technology and Mechatronics Automation

Design and Implementation of Engraving Machine Controller

Yang Lin, Hu Tianliang, Zhang Chengrui
Key Laboratory of High Efficiency and Clean Mechanical Manufacture,
School of Mechanical Engineering, Shandong University, Jinan, 250061, China
s3n@163.com, tianliang.hu@hotmail.com, zhangchengrui@gmail.com

***Abstract*—This paper presents a software architecture for engraving machine control system based on hierarchical finite state machine and multithread programming. Interactive sequence diagram and dataflow chart are described in detail for manual and automatic operation mode, which provides a consistent and flexible way to construct CNC control system.**

***Keywords-** engraving machine; CNC; software architecture*

I. INTRODUCTION

Engraving machine tools are widely used in the manufacturing of wood, stone, glass and metal products. Engraving machine tool control system is more than a CNC, it has the following characteristics:

- The NC program files generated by CAD/CAM software are very large, the file size ranges from several MB to hundreds MB;
- In the NC programs, the freeform curves of the machining parts are usually presents by small line segments. To improve the machining efficiency and guarantee the surface quality, acceleration control is usually very complex. Also look-ahead function is needed.
- With the development of the industry, small batch manufacturing is more and more common in engraving manufacturing. The updating of the engraving machine controller becomes very frequent.

Therefore, general CNC controller is seldom used in engraving machine field. To realize the special functions above, open and flexible controller is needed. PC-Based system can take the advantage of rich software and hardware resources. In the market nearly half of the engraving controllers are PC-based. In these PC-based controllers, Windows system with RT-extension or hard real-time Motion Control card is used, which are usually very expensive[1,2]. Aiming at solving the problems above, in this paper, a novel software design method, which is based on hierarchical finite state machine and multithread programming, is implemented on PC-based Windows platform to provide a consistent and flexible way to construct CNC control system. A real-time Ethernet fieldbus- EtherMC (Ethernet for Motion Control) [3], which is developed by our research group, is adopted. Windows kernel programming is used to guarantee a soft real-time characteristic. With a special designed mechanism of EtherMC, hard-real time ability can be obtained. In the following sections, the hardware architecture of engraving machine control system is described. Subsequence section presents the software structure in details.

II. HARDWARE ARCHITECTURE

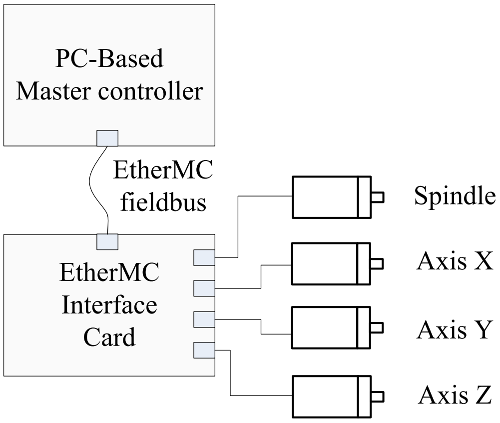

Figure 1. System architecture overview

The system hardware is designed as figure 1. In this architecture, EtherMC, which is developed by our research group, is used as the implementing platform. In this platform, the EtherMC Master is PC-based, and NC kernel is realized in the Master controller. EtherMC has the following characteristics:

- Easy to do reconfiguration. All slaves of EtherMC is connected to the master through a line topology. Slaves can be added or removed from the system easily.
- Standard Ethernet is used in EtherMC. The costs can be greatly reduced.
- One slave node should be configured as the synchronous real-time control node in EtherMC system to give synchronous realtime signal periodically. With this design, only soft real-time ability is needed for the Master controller, hard real-time is not necessary.

These characteristics make EtherMC fit the demands of engraving machine controller. In this design, a four axis interface card is used as the EtherMC slave. This card converts the data commands sent from Master to controlled axes, such as spindle, Axis X, Axis Y and Axis Z.

978-0-7695-3962-1/10 $26.00 © 2010 IEEE
DOI 10.1109/ICMTMA.2010.748

III. SOFTWARE ARCHITECTURE

The control software adopts multi-layer architecture as shown in figure 2, including human machine interface (HMI) server, HMI client, NC kernel, communication library and Ethernet protocol driver.

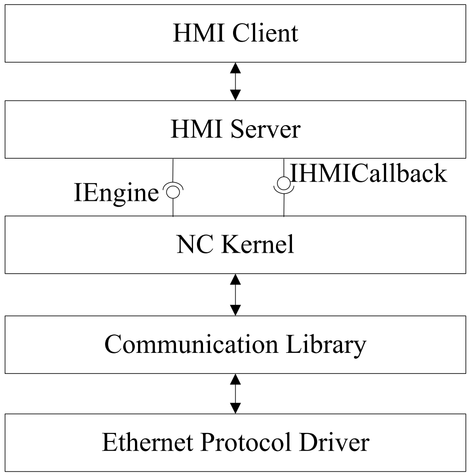

Figure 2. Multi-layer software architecture

NC kernel implements IEngine interface that performs a facade layer which encapsulates the internal structure of NC kernel. HMI server receives commands sent from HMI client, and invokes the IEngine interface method accordingly, which turns the method parameters into a state machine event object and dispatches the event object to state machine. When NC kernel internal data changed, IHMICallback interface will be called to notify HMI server, which then updates HMI client. Ethernet protocol driver provides the real-time ability by kernel deferred procedure call (DPC) of Windows operating system, which has priority over all user level threads [4].

A. NC Kernel

Inside NC kernel, a hierarchical state machine (HSM) takes care of the operating mode of NC, which is on top of the function modules, shown as figure 3. Only in non-running sub state, the system can be switched to another operating mode. HSM processes events sent from user interface or internal function modules and coordinates the global behavior of the system. The HSM invokes the function module accordingly to implement the actual behavior.

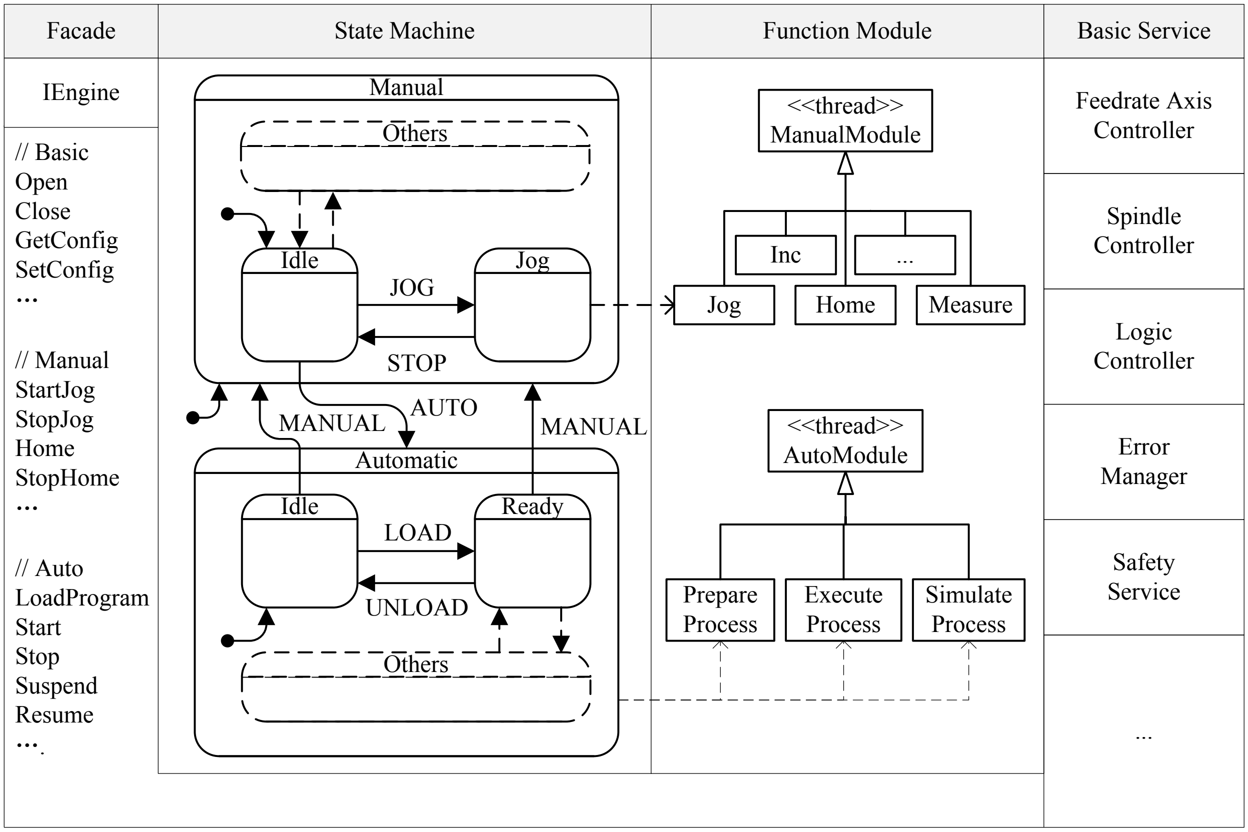

Figure 3. NC kenernal component structure. The dependency between components is from left to right

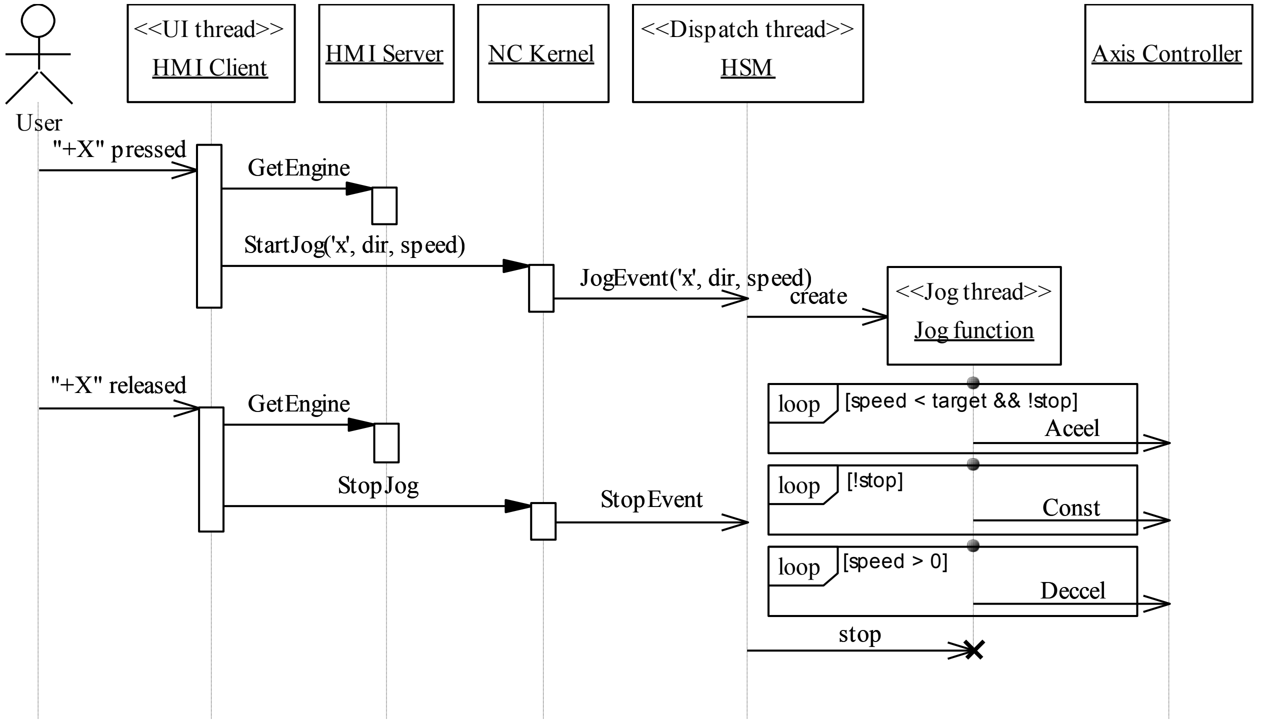

Figure 4. UML sequence diagram for jog operation

State machine engine normally assumed run-to-complete semantics by queuing events and dispatching them sequentially [5]. The state machine engine executes in a separate thread, should not be blocked by the corresponding function module, otherwise it will not response the other events. Therefore, the system function modules, such as automatic running, jogging, homing, tool measuring etc., execute in its own thread. State machine communicates with function module thread by volatile flag variables or operating system native synchronize objects. For example, jog mode has following execution sequence as shown in Figure 4:

(1) User switches to manual mode if system in a suitable state, then the initialize action of *Manual* state transfers from super state *Manual* to sub state *Manual/Idle*;

(2) Positive X axis button is pressed in user interface, in the button's *ButtonDown* event handler, HMI client gets NC kernel's *IEngine* interface from HMI server, and calls the interface method *StartJog* with parameters, including axis, direction and speed. NC kernel packs the method parameters into *JogEvent* object and enqueues it into the event queue of HSM . All above steps are executed in UI thread context;

(3) State machine engine dispatching thread dequeues the *JogEvent* object and dispatches it to active state *Manual/Idle*. System current state transfers from *Manual/Idle* to *Manual/Jog*. The initialize action of *Manual/Jog* creates a new thread to perform actual operation, and sets flag *stop* to *false*;

(4) In the jog function thread, speed is accelerated for each communication cycle reached jog command speed is reached, and then it will runs at the constant target speed. Interpolation data is sent to Ethernet protocol driver through axis controller, which converts the step length to cycle pulses by take count of mechanical and electrical parameters. At each step, flag *stop* is checked. If the flag is set to *true*, the speed will be decelerated to zero;

(5) Whenever user releases the jog button, its *ButtonUp* event handler is executed. As described in step 2 and 3, a *StopEvent* is sent to *Manual/Jog* state, flag *stop* is set *true* to notify jog thread to decrease speed and exit thread.

B. *Automatic mode*

In automatic mode, the NC kernel executes the NC program continually, which contains both one-shot command and modal command. NC interpreter parses the G-code source file and inserts the interpreted commands into a command buffer. A runtime context object is used to record current modal data, such as plane selection code (G17/18/19), interpolation type code (G00/01/02), cutter compensation type code (G40/41/42), and so on. When look-ahead function is applied, the modal data may be different between look-ahead position and actual execution position. Thereby two threads are used to execute the buffered commands: one is the preparation process, and

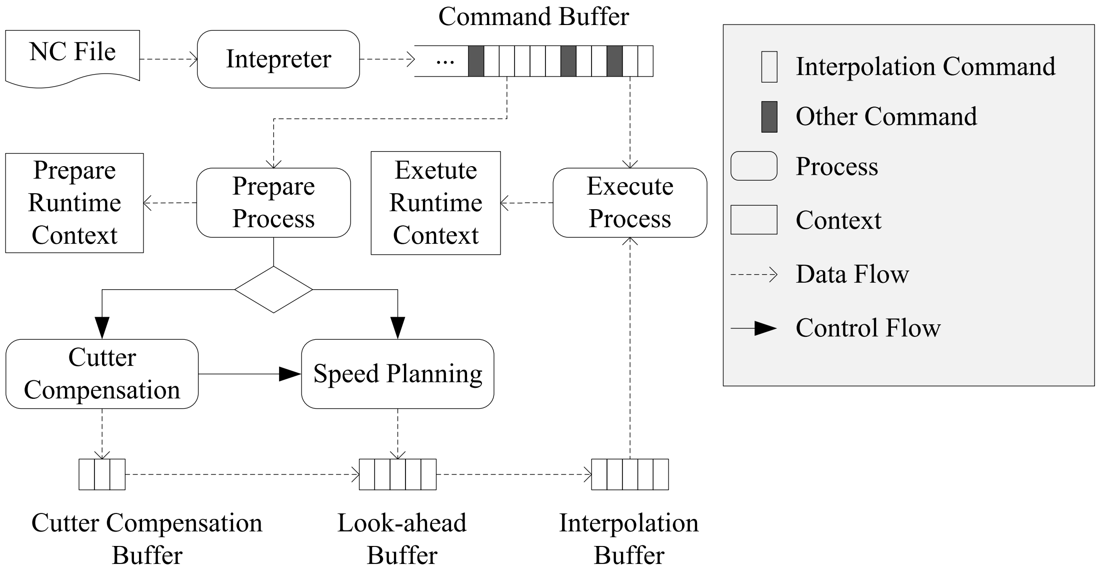

Figure 5. Automatic mode dataflow diagram

another is execution process. Both threads hold its own runtime context object, when execution is suspended and resumed, the runtime context object held by execute process is copied to the object held by preparation process.

Preparation process and execution process execute the commands concurrently as shown in figure 5. When they encounter the modal changing command, the new modal data is stored into runtime context object respectively. While interpolation commands is processed differently. On one hand, when preparation process reads an interpolation command, it translates the coordinate value from workpiece coordinate system (WCS) to machine coordinate system (MCS) according to current workpiece coordinate selection (G54-G59), external workpiece zero point offset, absolute / incremental mode (G90/91) and inch/metric mode (G20/21). If cutter compensation is demanded, the command will be fed into cutter compensation buffer, else into look-ahead buffer. Look-ahead algorithm [6, 7] is applied to the commands stored in look-head buffer, and once interpolation end point velocity is determined, it is moved into interpolation buffer. On another hand, when execute process reads an interpolation command, it dequeues the peer interpolation from interpolation buffer that contains preprocessed interpolation commands. If the peer is not available, the execution process thread will wait until it is enqueued by the preparation process. Then execution process sends the interpolation data to the device periodically.

IV. CONCLUSION AND FUTURE WORK

This paper gives an implementation method of engraving machine controller. In this method, EtherMC fieldbus is used as the hardware platform to ensure the flexibility of the hardware. In the design of the software, hierarchical state machine is used as the control layer for specific function module. It fits well with operation mode of CNC. Multithread programming cooperates with the state machine to process continuous long-run task. This method can obtain good flexibility of the CNC functions. The Future work will focus on system openness to provide interoperability, maintainability, reconfigurability and low development cost under this framework.

V. CONCLUSION AND FUTURE WORK

This paper is supported by National Natural Science Foundation of China (under contract of 50675123) and Natural Science Foundation of Shandong Province (under contract Y2007F47).

REFERENCES

[1] HUANG Ying-hua, FANG Kai and ZHANG Yao-xin. Research and Realization on DSP Based Control Card Applied to High-speed CNC Engraving and Milling Machine. Automation and Instrumentation. 2006 21(2): pp.55-59

[2] CAO Hao-ying, WANG Xiao-chun, ZHANG Bing. The Design of CNC System of a kind of Hi-Speed CNC Engraving and Milling Machine Tool. Modular Machine Tool and Automatic Manufacturing Technique. 2005(12): pp.64-66

[3] Tianliang Hu. Research on the Design Method for STEP-Compliant Open CNC Platform. Ph. D Dissertation of Shandong University. Jinan, China. 2008.

[4] Walter Oney: Programming the Microsoft Windows Driver Model, Second Edition, Microsoft Press, 2002.

[5] Miro Samek: Practical UML Statecharts in C/C++, Second Edition, Newnes, 2008.

[6] Han, G. C., D. I. Kim: A high speed machining algorithm for CNC machine tools. Industrial Electronics Society, 1999. IECON '99 Proceedings. The 25th Annual Conference of the IEEE, 1999.

[7] Hong-Tzong, Y., W. Jun-Bin: Development and implementation for real-time lookahead interpolator by using Bezier curve to fit CNC continuous short blocks. Mechatronics, 2005. ICM '05. IEEE International Conference on, 2005.