Proceedings of the 21st International Conference on Automation & Computing, University of Strathclyde, Glasgow, UK, 11-12 September 2015

Reconfigurable software architecture for a hybrid micro machine tool

Wenbin Zhong, Wenlong Chang, Luis Rubio, Xichun Luo*
Department of Design, Manufacture and Engineering Management
University of Strathclyde
Glasgow, G1 1XJ, UK
*xichun.luo@strath.ac.uk

***Abstract*—Hybrid micro machine tools are increasingly in demand for manufacturing microproducts made of hard-to-machine materials, such as ceramic air bearing, bio-implants and power electronics substrates etc. These machines can realize hybrid machining processes which combine one or two non-conventional machining techniques such as EDM, ECM, laser machining, etc. and conventional machining techniques such as turning, grinding, milling on one machine bed. Hybrid machine tool developers tend to mix and match components from multiple vendors for the best value and performance. The system integrity is usually at the second priority at the initial design phase, which generally leads to very complex and inflexible system. This paper proposes a reconfigurable control software architecture for a hybrid micro machine tool, which combines laser-assisted machining and 5-axis micro-milling as well as incorporating a material handling system and advanced on-machine sensors. The architecture uses finite state machine (FSM) for hardware control and data flow. FSM simplifies the system integration and allows a flexible architecture that can be easily ported to similar applications. Furthermore, component-based technology is employed to encapsulate changes for different modules to realize "plug-and-play". The benefits of using the software architecture include reduced lead time and lower cost of development.**

Keywords-hybrid micro machine tool; reconfigurable software architecture; finite state machine; component-based technology

I. INTRODUCTION

Conventional machining techniques have been increasingly challenged by some highly engineered mechanical products such as gas turbines, advanced automotive systems and heavy off-road equipment, which rely on high strength materials [1]. Although the recent improvements on tool materials have improved the machinability on advanced materials such as aeroengine alloys, structural ceramics and hardened steel to some extend [2,3], the productivity and manufacturing cost remain problems to be solved. Nevertheless, hybrid machining processes have the unique advantage of being able to machine the difficult-to-cut materials with high material remove rate, while achieving fine surface finish and reduced tool wear. Figure 1 shows the machined surface topography of hardened steel (53HRC) by ultrasonic-assisted diamond machining and the diamond tool tip after the machining. It can be seen that good surface quality (Ra of 4.38 nm) and negligible tool wear were achieved. There are a number of hybrid machining processes having been developed, which can be classified as assisted and combined hybrid machining processes [5].

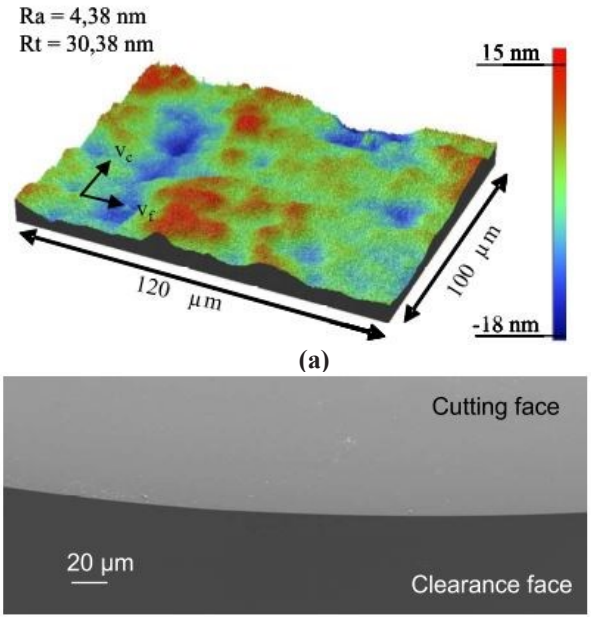

Figure 1. Ultrasonic-assisted diamond machining on hardened steel: (a) Machined surface topography; (b) SEM image of the monocrystalline diamond tool after machining [4]

However, there are a number of issues in the development of hybrid machines that are fundamentally different from the development of conventional machines. The reason lies in that hybrid machines need to incorporate more components from different vendors to achieve the desired functionalities. A typical configuration of a hybrid machine is that a multi-axis machining control system works with a non-conventional machining controller as well as many advanced on-machine sensors. Therefore, a reconfigurable software architecture that enables hybrid machines to react to changes rapidly is highly desirable. Unfortunately, there are no generic approaches for the design of software for such highly customized hybrid machines presently, which is largely due to the lack of standards in sensor interfaces, signal processing algorithms and control systems. Great efforts are required to maintain and upgrade the machines because of the low reconfigurability of the software architecture. In recent years, open architecture controllers,

which are PC-based solutions with a homogenous and standardized environment, are widely witnessed among industry. They provide the possibility of continuously integrating new advanced functionality into the control system from hardware side. Figure 2 summarizes the trend of development of open controller architecture and its components.

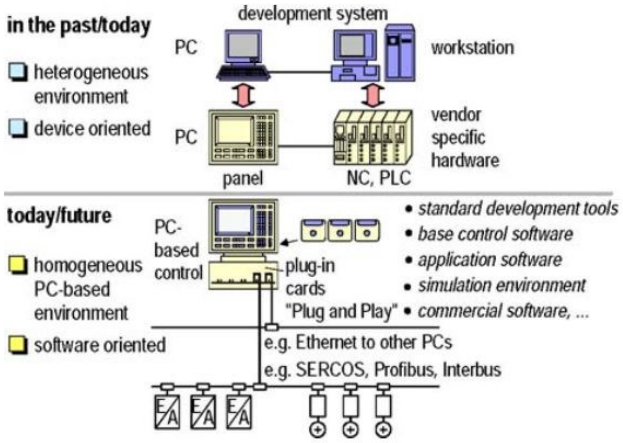

Figure 2. PC-based, software-oriented Control Systems [6]

Nor and Cheng successfully developed a PC-based control system for a five-axis ultra-precision micro-milling machine — 'Ultra-Mill' [7]. Some third party sensors were connected to the system based on open controller architecture. However, the software architecture was based on the CNC controller used in the application. The software integrated development environment (IDE) was also provided by the CNC vendor. Such device-oriented architecture with proprietary hardware and software components decreases system integrity as well as reconfigurability.

In this paper, the requirement for the reconfigurability of control software architecture for hybrid micro machine tools is investigated. The focus will be on the design of software architecture for a hybrid micro machine tool with laser-assisted 5-axis micro-milling capability, integrated with a material handling system, on-machine metrology and on-line force measurement. Two key technologies used in the implementation of the architecture are discussed. Finally three reconfigure scenarios are provided to illustrate characteristics of the software architecture and its advantages over device-oriented architectures.

II. SYSTEM DESCRIPTION

Ultra-precision micro-milling machine tools are widely used to manufacture 3D complex micro-components at sub-micron accuracy and nanometer surface finish. This project aims to develop a hybrid micro machine tool to push the limits of ultra-precision micro-milling, viz. enhancing the machining performance on hard materials and increasing the productivity. The machine consists of a 5-axis micro-milling system, a laser–assisted machining system, a material handling system, and two sensor systems, including an interferometer for on-machine metrology and a dynamometer for on-line cutting force measurement.

Each sub-system is developed or provided by a different vendor with its own hardware and software and can function independently. However, as a whole, they should be coordinated to collaborate seamlessly to achieve the goal — hybrid machining. In this application, the open controller architecture is adopted. A PC acts as the coordinator, while all the systems can exchange data with the PC through communication interface or peripherals, as shown in Figure 3. A software system running on Windows operating system is required to manage and process the data from each sub-system as well as providing the human-machine interface (HMI).

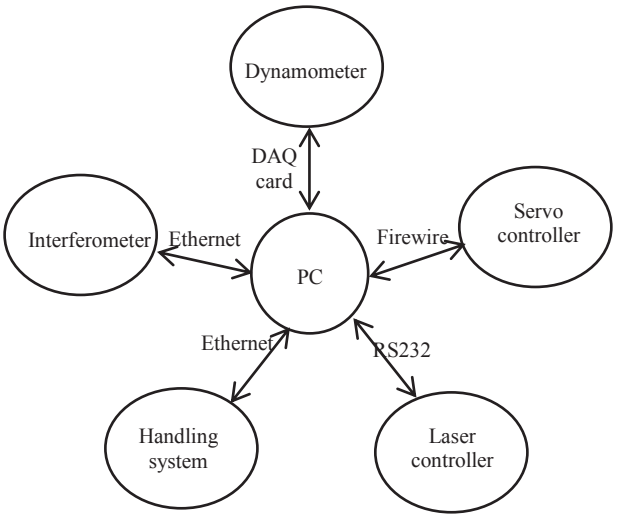

Figure 3. System connection diagram

A reconfigurable software architecture allows the modification of functionality in a very dynamic way. There are five characteristics of reconfigurability, which can be summarized below [8]:

- Modular. Decentralized structures are supported. All components are encapsulated as vendor-neutral modules.
- Interoperable. Components can cooperate in a consistent manner and can interchange data in a defined way.
- Portable. Components can be integrated in different environments without any changes, while maintaining their capabilities.
- Scalable. Topology of the architecture can be modified depending on the user requirement.
- Extendable. Functionality can be enhanced. A variety of components can run on the platform without any conflicts.

Modular and interoperable are the critical characteristics of reconfigurability. They ensure the rest three characteristics. To fulfill the criteria of being reconfigurable, the following principles should be taken into consideration:

- Decoupling. Too much coupling results in an inflexible architecture.

- Vendor-neutral. This feature ensures that a variety of components can be incorporated with minimum effort.
- Standard-based. The data exchange and application interface (API) should be standardized, so that a component can be distributed rapidly.

III. SOFTWARE ARCHITECTURE DEVELOPMENT

From the control perspective, a typical machine tool has three control layers, the hierarchy comprises servo control, process control and supervisory control from bottom to top. These control layers interact with machine, process and product respectively, and lead to three control loops. Each control loop is running upon a group of hardware, and controls some parameters within this layer, while a bottom layer will get instructions from its top layer. The control scheme can help look into the design of the software architecture with the reconfigurable design principles described in section 2.

Firstly, the scheme covers all the machine functions with very clear data flow. For easy cooperation of different components, the data, which contains control messages and sensor feedback, can be packaged as events. All the possible events are defined in a protocol that all components on the platform should follow. As a result, the software architecture becomes event-driven; Secondly, all hardware that services for the machine can be placed in one of the layer. After encapsulation, the function of the hardware can be a vendor-neutral component in the architecture; Finally, the coupling is remarkably reduced with the event-driven mechanism and vendor-neutral components. Therefore, in this software architecture, each sub-system, for example servo controller, can be regarded as one unified module with specific events in/out interfaces, and can be encapsulated into a component. During the process, FSM is used in implementing the event-driven mechanism, while component-based technology is used for modularization.

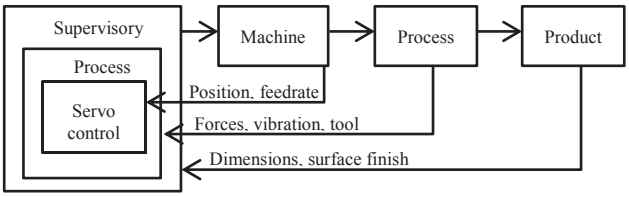

Figure 4. Machine, process and product interactions in the control loops

A. FSM Model

FSM can be observed widely in software industry but is rare in the machine tool software design. It can change from one state to another when specific event is triggered. In this application, FSM is implemented in both component level and architecture level. Figure 5 illustrates the work principle of a FSM. When an event is received, FSM will call the API defined in the knowledge base to process it, with the state change following. The FSM itself may trigger events, which will be dispatched to other modules. The type of events to be received and the type of events can be posted are agreed using the protocol. An unknown event will cause an exception. The knowledge base along with the FSM is of the greatest importance. Typically, there are two types of APIs within the base. The first type is hardware abstract APIs, which are directly related with hardware and ensure the proper operation of the hardware; Another type is advanced algorithms for data processing, for example, the workpiece surface data from the metrology sensor is processed to analysis the surface quality, the results can be used to optimize the machining parameters.

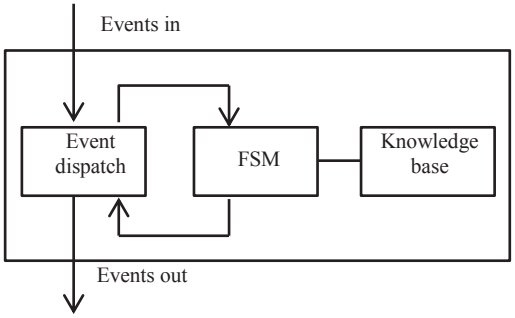

Figure 5. FSM work principle

Each FSM model for a sub-system comes with standardized data exchange, which is defined as events. The events implementation varies with the programming language to be used, a structure can be used when programming with C/C++, the structure may have the following definition:

```
typedef struct {
        DWORD dest;
        UINT message;
        WPARAM wParam;
        LPARAM lParam;
        DWORD time;
} EVENT;
```

In this example, `dest` identifies the destination of the event, `message` specifies the ID of the event, `wParam` and `lParam` specify some additional information of the event, `time` indicates the time when this event was posted. The available event ID and exact meaning of additional information are explained by the protocol. A new task is created to let one FSM work, which leads to parallel software tasks in the architecture, and makes the model more independent.

B. Modular

From the software perspective, a component is a reusable piece of software that serves as a building block within an application. It is also language-neutral that can be used in a variety of languages, which makes it easier to integrate. Essentially, the software is built with a set of components, with events flowing from one component to another. Besides the sub-systems described in Section 2, HMI is essential for the architecture. HMI, as its name suggests, is responsible for the interaction with operators. It collects command inputs from operators and dispatches them to the corresponding sub-systems, as well as receiving update events to update the display. All the sub-systems are based on a FSM model presented before, and

they are encapsulated into several components or modules as shown in Figure 6. There are several component based technologies, of which Windows COM technology is used in this application.

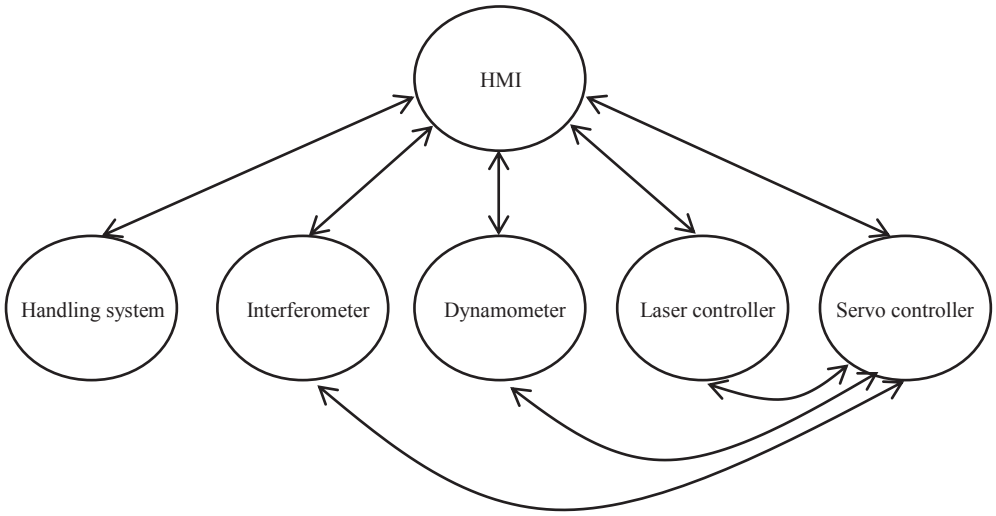

Figure 6. Modules and events flow in the architecture

IV. CASE STUDY

The case study includes the software reconfiguration in three scenarios: change a component, add a component and add a feature.

A. Change a Component

This case considers when a new hardware should replace the old one for better performance, for example, a new type of laser is to be installed. In this case, a new component is created by implementing the same component interface, states and state transitions, then it can replace the old one without causing any problem, the only change needed to be made is that adapting the hardware abstraction API to the new controller. The good scalability is achieved.

B. Add a Component

This case looks into the condition that if a new module should be added. For example an acoustic emission sensor is needed to detect tool breakage. In this case, the protocol defining the events should be extended, and the related FSM model should be expanded to accommodate the new events. The extendibility is improved due to the architecture.

C. Add a Feature

Another common case is that a new algorithm is developed to improve the machine performance when the machine has already been delivered. For this scenario, the quick solution is that adding the algorithm in the related component and adding an event at which the algorithm will be called.

V. CONCLUSION

A software architecture has been developed for a hybrid micro machine tool. It uses FSM technology for hardware control and data flow. The coupling between different modules is minimized. The data exchange between different components is standardized to events. The architecture encapsulates modules using component-based technology, which improves the reconfigurability of the software system. Results from this paper provide a general methodology for designing the software architecture for hybrid machine tools, especially these incorporating several subsystems from different suppliers.

ACKNOWLEDGMENT

The authors gratefully acknowledge the financial support of EPSRC (EP/K018345/1).

REFERENCES

[1] B. Lauwers, F. Klocke, A. Klink, a. E. Tekkaya, R. Neugebauer, and D. McIntosh, “Hybrid processes in manufacturing,” CIRP Ann. - Manuf. Technol., vol. 63, no. 2, pp. 561–583, 2014.

[2] E. O. Ezugwu, “Key improvements in the machining of difficult-to-cut aerospace superalloys,” Int. J. Mach. Tools Manuf., vol. 45, no. 12–13, pp. 1353–1367, 2005.

[3] E. O. Ezugwu, J. Bonney, and Y. Yamane, “An overview of the machinability of aeroengine alloys,” J. Mater. Process. Technol., vol. 134, no. 2, pp. 233–253, 2003.

[4] B. Bulla, F. Klocke, O. Dambon, and M. Hünten, “Ultrasonic Assisted Diamond Turning of Hardened Steel for Mould Manufacturing,” Key Eng. Mater., vol. 516, pp. 437–442, 2012.

[5] S. Z. Chavoshi and X. Luo, “Hybrid Micro-machining Processes: A Review,” Precis. Eng., vol. 41, pp. 1–23, 2015.

[6] G. Pritschow, Y. Altintas, F. Jovane, Y. Koren, M. Mitsuishi, S. Takata, H. van Brussel, M. Weck, and K. Yamazaki, “Open controller architecture–past, present and future,” *CIRP Ann. - Manuf. Technol.*, vol. 50, no. 2, pp. 463–470, 2001.

[7] M. K. M. Nor and K. Cheng, “Development of a PC-based control system for a five-axis ultraprecision micromilling machine ‘Ultra-Mill’ and its performance assessment,” *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.*, vol. 224, pp. 1631–1644, 2010.

[8] Y. Koren, U. Heisel, F. Jovane, T. Moriwaki, G. Pritschow, G. Ulsoy, and H. Van Brussel, “Reconfigurable Manufacturing Systems,” *CIRP Ann. - Manuf. Technol.*, vol. 48, no. 2, pp. 527–540, 1999.