New York University
A private university in the public service

Courant Institute of Mathematical Sciences
Robotics Research Laboratory

715 Broadway, 12th floor
New York, N.Y. 10003
Telephone: (212)998-3464

email: greenfel@acf8.nyu.edu

1 July 1988

MOSAIC

A MACHINE-TOOL'S OPEN-SYSTEM ARCHITECTURE, INTELLIGENT CONTROLLER

by: Israel Greenfeld

1 Scope

This brief note describes and discusses new ideas concerning machine tool controllers. In the center of focus stands the notion of the *Machine-tool Open-System Architecture Intelligent Controller (MOSAIC)*, which is suggested as a possible future substitute for the current controllers. MOSAIC will most likely be based on a low end computer such as a workstation or a PC, that will provide the machine tool with an advanced and flexible environment for programming and communicating.

This new approach is being now developed at NYU - Robotics Laboratory based on the experience gathered at the NYU Machining Cell, working on a Kitamura machine tool with a Fanuc controller, supported by the cadcam system Anvil-5000. This approach ties up with other concepts developed at NYU - Robotics with the *Machine Tool of the 21st Century* in mind.

2 Background

Today's automatic machine tools, and similar computerized manufacturing machines and robots, are controlled by Computerized Numerical Controllers (CNC) and Programmable Logic Controllers (PLC). These controllers are very widespread in industry and are found on almost any automatic machine tool. Some controllers are unique to a particular machine, and some are general purpose devices that will fit to a variety of machine tools just by setting parameters. The latter ones are predominantly manufactured by specialized companies such as Fanuc. The CNCs and PLCs are ruggedized for the industrial environment and are usually mounted directly on the machine in an upright position. They perform the servo control of the motion axes and provide an operating and programming environment unique to machining and to similar production processes.

The CNCs and PLCs are usually very limited in terms of operating and programming flexibility and in terms of communicating with external computers. They cannot accomodate non-machining devices such as workholding accessories, force sensors, vision sensors, and other devices that are likely to appear in the near future in the machining arena. They are programmed in a very low level machine code which is not practical for direct programming. A large number of post processor systems is available which translates the output of a particular cadcam system to the machine code format acceptable by a particular controller. The CNCs can read and execute machining files sequencially as sent, block by block, but they can't be easily controlled from an external computer. For example, you can't interrupt a machining process from a remote terminal unless you have installed a specific hardwire connection. They can write an output to an external computer in a batch process, but they cannot easily supply to it continuous status data such as axes' positions.

The current CNC and PLC hardware configuration is characteristic of the tradition oriented machining industry. A typical controller, although using the most advanced electronics, may still include a solid state memory for storing programs, a reader/puncher for paper tape programs, and an operator's panel loaded with buttons, switches and dials.

3 Functions of Current Controllers

The current machine tool controllers typically support the following user accessible functions:

(1) *Servo Control* of the motion axes, with very accurate path and positioning, and with backlash and resilience compensation.

(2) *Machining Code* programming language to perform low level tasks such as linear and circular interpolation, spindle control, tool loading, mode selection and canned cycles.

(3) *Macro Code* programming language which provides a general purpose, low level programming environment. It allows management of variables, including machine parameters, basic program control functions and output functions.

(4) *Offsets Management* for tool wear and geometry compensation, and for work coordinate settings.

(5) *Operating System* which controls the user's interaction with the operator's panel, and allows the management of parameters and troubleshooting. The parameter settings relate to the characteristics of the control loops, communication ports and protocols, operator's panel options, programming and editing functions, etc. They define the configuration of the specific machine on which the controller is installed.

(6) *Operating Panel* for editing and executing programs, for manual operation and for all other operating functions. Some controllers also provide a graphical monitor to facilitate programming.

(7) *Communication Ports*, mostly for the RS232 protocol. The number of available ports is usually very small.

4 Functions of Future Controllers

The machining environment of the "21st century" will entail a heavy load of devices installed on the machine tool. Typical devices may be automatic jigs and fixtures for workholding, dextrous manipulators which will substitute the machinist hand, vision systems for watching the process to identify parts and events, force sensors to monitor machining stress, touch or optical probes for dimensional gauging and part localization, and other devices. These devices will be part of the machine-specific configuration, and should be controlled locally rather than from an external controller. A full integration of all the components will be essential and will require an open architecture versatile system, with a universal and flexible operating system that provides a good programming environment and interfaces.

The machine shop floor will be controlled by higher level computers which manage the flow of manufacturing jobs, parts and materials and handle the status information supplied by the machines. A lot of effort is spent today on establishing universal communication protocols (like MAP) and systems for the future factory [3-7]. Much of the difficulties in the way to this goal lie in the close, even secretive, nature of current controllers. Future controllers should be open and flexible to overcome this communicative obstacle.

In view of this scenario a wishfull list of future controller functions and characteristics may contain the following:

(1) *High Level Machining Language* for machining programming, that would make the machine code transparent to the user. Existing machining languages such as APT and Compact may be supported. More advanced, higher level, languages may be developed to simulate the more intuitive Cam environment that cadcam systems provide.

(2) *General Purpose Programming Language* such as C or Pascal, in which the applications may be written and directly interfaced with the machining environment. In fact, it may be the programming media for the previously mentioned machining language.

(3) *Machining Operating System* which will provide the high level operating functions needed to control the machining process and the various devices, and to handle the interfaces and communications. It will manage the status and configuration data and the programs' run. It may be considered as a shell invoked within the supervising general purpose operating system.

(4) *General Purpose Operating System* such as Unix or OS/2 which will support the interfaces and communications and will provide a good programming environment. The real time component of the system will handle the control loop in conjunction with the machining operating system.

(5) *Operating Terminal* will be the controller's workstation, that will completely emulate the operator's panel in a schematic/iconic representation. The terminal may be on a desk, away from the machine. Depending on the workstation's configuration, the terminal may be used to serve a few machines simultaneously as a user's interface and a supervisor.

(6) *Interfaces* may include a bus such as the IEEE 488 and the PC Bus with a multitude of ports for the input and output from the controller to the machine and its adjacent devices. For outside communications advanced systems such as EtherNet should be supported.

5 Examples of Applications

A few applications that are being considered at NYU - Robotics for the future machining environment, that cannot be easily implemented on current controllers, are listed below:

(1) *Dexman*
Dexman is a dextrous manipulator attached to the machine tool, which performs the tasks of the machinist's hand. The challenge: to move Dexman simultaneously in its own axes and in the machine tool axes.

(2) *Machining Stress Control*
Machining forces are continuously measured during cutting, and the feedrate and spindle speed are controlled to conform to a particular force specification. The challenge: to include such a provision in the control loop.

(3) *Automatic Workholding System*
The jigs and fixtures are positioned and controlled in a flexible manner. The challenge: to synchronize with the machining operations, the vision system, etc.

(4) *Vision System*
The vision system is used to locate and identify parts and jigs and to monitor the general activity for special events. The challenge: to use the vision data with the controller's operations.

(5) *Probing System*
The probe provides the controller with part localization information and gauging data. The challenge: full interaction between the gauging and machining.

6 Configuration

It is premature to indulge in detailed configurations, and therefore my description should be regarded as raw ideas and examples, rather than rigorous suggestions.

Generally speaking, MOSAIC may be configured as either a local machine controller or a supervisor for several machines. In particular, functions such as machine tool control should reside on the machine, while the user's interface may be away. Control of the machine's specific accessories (Dexman, vision, probe, etc.) may be performed on the MOSAIC itself, which seems to be difficult to achieve, or it may reside on a supervisor controller. In any case, MOSAIC should be designed to be able to interact in real time with all the accessories.

6.1 Hardware

Availability and price are primary considerations in the selection of hardware. The hardware configuration may consist of a standard PC or a workstation, with disk storage, a graphic monitor, and an expansion bus. Additional control processors, servo control, and I/O boards will be installed on the bus. A universal control and instrumentaton bus may be selected, such as the IEEE 488 (GPIB), PC Bus or VME Bus. All these are well supported in the market [8-15].

Other configurations may be considered, such as the SPARTA system [16] and [17] from IBM, which is based on signal processors mounted on a PC Bus, and provides a very high control performance.

6.2 Software

The software configuration may consist of a general purpose environment and a machining-specific environment. The general purpose operating and programming environment may be based on existing systems such as Unix and C. It will handle communications, file serving and the programming environment.

The machining operating and programming environment would have to be written. Apt like languages may be considered as a first step, but a new, broader and more flexible machining language, is needed. It should include such provisions as the ability to change the control laws during motion, the ability to branch, the ability to interpolate complex curves, etc. This language may be written in a general purpose language like C, and thus be compilable and retain all the advantages of a universal language.

The machining operating system should be designed to accommodate all the various operations and systems related to the future machine tool. It should include provisions for the operation of the machine-specific accessories, and should be able to synchronize all planned and unplanned events. It may be written in the general purpose operating and programming environments and thus retain their flexibilities.

A few control and instrumentation general purpose software systems, such as Labview TM [8] for the Macintosh, supporting the IEEE 488, are appearing in the market, and may be considered as a preliminary solution for the low level control.

7 Research Goals

Three main research areas are associated with MOSAIC:

(1) *Machining Language*
Research in a new versatile machining language, possibly compilable.

(2) *Machining Operating System*
Research in a new machining operating system that will incorporate all the activities in the future machine tool, and enable communication with the factory systems.

(3) *Systems Integration in Machine Tools*
Research in the integration of accessory equipment with the machine tool operating and programming environment.

8 Research and Development Outline

Although the control system is not an issue for research, it is going to be a very difficult part of the design and integration of the prototype of MOSAIC. Much of this is based on experience in machine tool servo control, which is not easy to get. I hope to be able to start with existing systems to minimize this difficulty. A collaboration with industry may be advisable.

Research in languages and operating systems should be conducted in conjunction with the CIMS departments that specialize in that. Specification of needs should be originated in the Robotics Lab.

Research in the integration of the MOSAIC system and the integration of applications such as vision, dexman, probe and fixturing seems to be the main research effort suitable for the Robotics lab.

Acknowledgment

I would like to thank my friend Jehuda Ish-Shalom for the valuable information and insights that he has given me, and for his willingness to collaborate on this project.

References

The reference list contains mostly articles from the popular engineering magazines, which I have found to be very informative on the industry trends. Not all the articles are directly related to this discussion. The list is sorted to the following groups:

(1) Industry strategies [1,2].

(2) Manufacturing systems configuration and networking [3-7].

(3) Industrial controllers software and hardware [8-15].

(4) Other research programs [16,17].

References

1. Joseph McNotty, "Checking the Pulse on the US Machining Center Market," *Manufacturing Engineering*, pp. 64-65 SME, (May 1988).
2. Gregory T. Farnum, "Mazak Targets Europe," *Manufacturing Engineering*, pp. 70-72 SME, (March, 1988).
3. James G. Ames, "Which Network is the Right One ?," *Manufacturing Engineering*, pp. 56-60 SME, (May 1988).
4. R. C. Baron, "Picking the Varied Fruit of Open Systems," *Manufacturing Enginnering*, pp. 52-54 SME, (May 1988).
5. David J. Parrish, "Opening a Dialogue Between FMS and CIM," *Mechanical Engineering*, pp. 70-76 ASME, (May 1988).
6. R. Lee Martin, "ENE '88i: Users Unite," *Manufacturing Engineering*, pp. 75-78 SME, (May 1988).
7. Peter J. Valce, "How to Justify the Cost of DNC," *Manufacturing Engineering*, pp. 67-69 SME, (May 1988).
8. Paul G. Schreier, "Controllers, Software Ease IEEE Instrument Use on Any PC," *Personal Engineering and Instrumentation News*, pp. 35-48 Personal Engineering Communications, Inc., (March 1988).
9. John Rosford, "Language Choice, Programming Techniques Optimize 488 Controller Throughput," *Personal Engineering and Instrumentation News*, pp. 55-60 Personal Engineering Communications, Inc, (March 1988).
10. Richard A. Eckhardt, Matthew F. Kubitsky, and Robert C. Molloy, *The PC Systems Handbook for Scientists and Engineers*, CyberResearch, Inc., New Haven, CT (1988).
11. Leland Teschler, "How PCs are Shaping the Cell Control," *Machine Design*, pp. 86-92 Penton, (February 25, 1988).
12. , "Linking PCs and Machine Tools," *Machine Design*, p. 179 Penton, (December 10, 1987).
13. David T. Robinson, "Industrial Controllers," *Machine Design*, pp. 103-106 Penton, (May 26, 1988).
14. Rick Furr and Glenn Dorsey, "Tips for Applying Servo Positioning Systems," *Machine Design*, pp. 70-73 Penton, (March 24, 1988).
15. Jacob Tal, "Circuit Boards Simplify the Route to Motion Control," *Machine Design*, pp. 124-128 Penton, (March 10, 1988).
16. Jehuda Ish-Shalom and Peter Kazanzides, "Sparta: Multiple Signal Processors for High-Performance Robotic Control," *Proceedings of the 1988 IEEE International Conference on Robotis and Automation*, IEEE, (April 24-29, 1988).
17. James U. Korein and et al., "A configurable System for Automation Programming and Control," *Proceedings of IEEE Conference on Robotics and Automation*, IEEE, (April 1986).