A Testbed for Research on Smart Machine Tools

Robert B. Jerard, Barry K. Fussell, Min Xu, Chad Schuyler
Department of Mechanical Engineering
University of New Hampshire, Durham, NH 03824

Abstract
In this paper we describe the development of a general testbed for research on "Smart Machine Tools." The hardware and software system components are described, along with the results of our investigation of a particular implementation of a smart machining system. We describe how sensors, models, information technology and computational resources can be combined to create the foundation for a Smart Machining System. The intent of this paper is describe a flexible and expandable testbed which could be used to explore a variety of alternative approaches. A test case is described which illustrates how the system can automatically select safe and efficient feedrates.

Keywords:
Open Architecture, Smart Machining, Machining Models.

1 INTRODUCTION

"Smart Machine Tools" is an emerging topic of interest and is the focus of recent industry initiatives [1, 2]. The December 2002 National Institute of Standards and Technology (NIST) workshop on "Smart Machine Tools" brought industry, academic and government agencies together to identify capabilities and needs for smart machine tools. The characteristics of a "Smart Machining System" (SMS) include:

1. The system learns from experience and future performance is improved based on stored knowledge and science based simulation,
2. Machining conditions are automatically selected to produce parts of the desired quality with maximum efficiency.
3. Models and sensors work synergistically to improve both the machining process and the accuracy of the models themselves through on-line calibration,
4. A high level language is used to communicate the part requirements, control the machining process, describe the physical components and store the history.

Taken together, these qualities allow the system to produce parts of the desired quality with the first part produced and every subsequent part.

2 SYSTEM COMPONENTS

2.1 Open Architecture Controller

A commercially available Open Architecture Controller (OAC) purchased from MDSI [3] serves as the foundation of our system. The selection of the MDSI control was based on the following factors:

1. We wanted a Windows PC based controller that would both control the CNC and run our applications simultaneously with no adverse effects on the machining process.
2. The system must have the ability to obtain position information, i.e. x, y, z, in real time, along with other sensor information such as slide and spindle motor power and slide velocities. It is critical that there be excellent synchronization between simulation models and real time measurements.
3. It must be possible to change the feedrate command signal in real time in response to optimization issues as well as real time changes in the process, such as tool wear.
4. Application programs must be able to run on the PC while the NC machine is running, to pass sensor and position information to the simulation model programs and to return feedrate, spindle speed and position commands to the NC control program in a timely fashion.

There are also a number of commercially available OAC controllers (e.g. MDSI, Fanuc, Siemens, Okuma). There are also a number of efforts at developing OAC standards [4-6]. We selected the MDSI primarily for reasons of cost and simplicity and it has met our needs.

2.2 Hardware Components

A schematic of the system components is shown in **Figure 1**.

1. **CNC Machine**: A FADAL VMC 40 was originally purchased in 1989 with a proprietary control and was retrofit in 1999 with an MDSI OAC. The VMC 40 was replaced with a FADAL EMC in 2006 and the existing MDSI controller is now used to control it.
2. **PC Computer**: Intel Celeron ® 2.4GHz CPU, 512MB RAM. One advantage of using the MDSI Open-CNC is that it is easy and relatively inexpensive to upgrade standard computer components with faster processing and more memory. We have upgraded the PC twice since 1999 without any significant disruption.
3. **Data Acquisition Board**: Computer Boards PCI-DAS 6402/16. The PCI-DAS6402/16 analog and digital I/O board offers 64 single-ended or 32 differential 16-bit analog inputs with sample rates up to 200 kHz.
4. **Force Sensor**: Kistler 9257B 3-Component Piezoelectric Dynamometer.
5. **Spindle Power Sensor**: Load Controls Inc.[7] Universal Power Cell (UPC). The UPC provides an analog output of 0-10 Volts proportional to spindle motor input power. The time constant is about 25ms. The LCI sensor non-invasive and easy to install. The sensor provides a clean and accurate (0.1%)

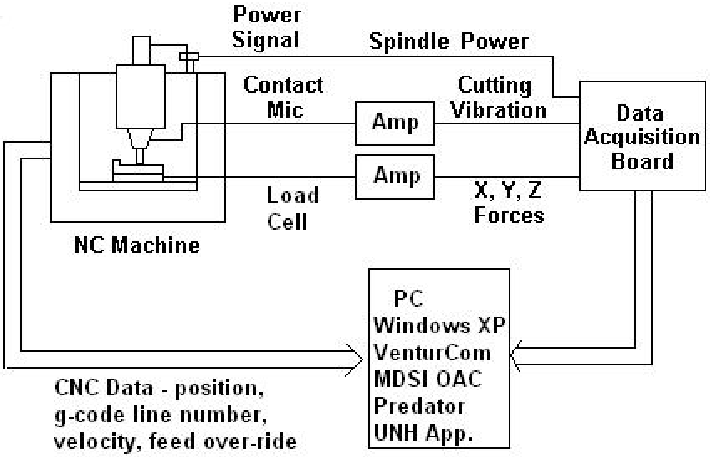

Figure 1 Hardware/software components of Smart

measurement of average motor spindle power input. The Baldor spindle motor efficiency versus power curve is used to estimate mechanical output power of the motor. The cost and non-invasive nature of the sensor make it ideal for the shop floor environment.

6. **ACG C411 Contact Condenser Microphone** – Microphones have been used in machine tool research and applications to detect chatter [8]. The ACG C411, originally designed as a contact microphone for stringed instruments, can be easily mounted either on the spindle or on a test part, is relatively inexpensive (<$200) and non-invasive.

2.3 Software Components:

1. **Operating System**: Windows 2000/XP. It takes less than 5% of the CPU capability of the PC to control the CNC, leaving plenty of excess capability to run our applications.
2. **Open Architecture Control Software**: MDSI OpenCNC. [3] A key ingredient to this is the use of shared memory where both the MDSI controller and our application can simultaneously access vital information like the current G-code line number, axes position, velocity and spindle speed [9].
3. **Real-Time Extension**: VenturCom RTX. [10]. The Windows operating system is not specifically designed for real-time operation so it requires some care in using it for CNC control. Reliability and safety are paramount concerns. A real-time command structure is required so that the motion control always receives the highest priority.
4. **Software Development Environment**: Microsoft Visual C++ 6.0. All of our application software is written in C++ providing speed, flexibility and portability.
5. **NC Simulation and Verification**: Predator Virtual CNC 4.0 SDK [11]. Predator Virtual CNC is a G-Code-based CNC simulation and verification application that simulates the CNC manufacturing process off-line. Predator's Feed Rate Analysis (FRA ATL V6.0) ActiveX Template Library (ATL) is a SDK that provides the cutting geometry information, such as material removed and contact area between the tool and the workpiece.
6. **UNH Application Software** –We have developed software modules for on-line calibration of machining models, wear analysis, runout analysis and feedrate optimization. Research methods are described in detail in numerous other publications [12-20]].

2.4 Information Technology: For a system to learn from experience there must be a systematic way of representing the part specifications, machine tool capabilities, cutting tool characteristics and process history. Conventional CNC controls rely on "G-codes" [9] which are inadequate in most respects. Considerable effort is being expended on improved technologies using the STEP standards. STEP-NC [21] is a new model for data transfer between CAD/CAM systems and CNC machines and is intended to replace G-codes.

Since STEP-NC is still a developing standard, we chose to develop a somewhat simpler language for our research. We developed NCML as an XML based dialect for representing machining processes [14, 15]. NCML represents conceptual process plans in a macro format that includes size tolerance information.

3 SYSTEM IMPLEMENTATIO

Figure 2 is a block diagram of our Smart Machining System [19]. In our vision the Designer (1) supplies a part description (2) in NCML format. The cost estimator (3) provides a bid which includes a breakdown of cost by machining features, thus enabling the designer to understand the relationship between design choices and costs. The Tool Path Planning (5) module compiles the macro conceptual process plan into individual toolpaths. The Machine Process Capabilities are also input to this module (Path A) to choose the proper strategies for the individual tool paths. The strategies include the choice of Unit Machining Operations (UMOs) [22] to make a given feature, tool choice, depth of cut, finish cutting allowance, etc. These strategies can be stored in Strategy Templates (4).

Speeds and feeds are set by using Process Models (6). The chosen feedrates for a given strategy depend on both the required accuracy (described by the tolerance information in the NCML file) and the Machine Process Capabilities. The OAC (7) commands the cutting tool motion to remove material from the workpiece (8) and continuously collects data to calibrate the machining models and dynamically estimate the current process capabilities of the machine. Sensors (9) monitor the process and provide feedback to the Control module (10)

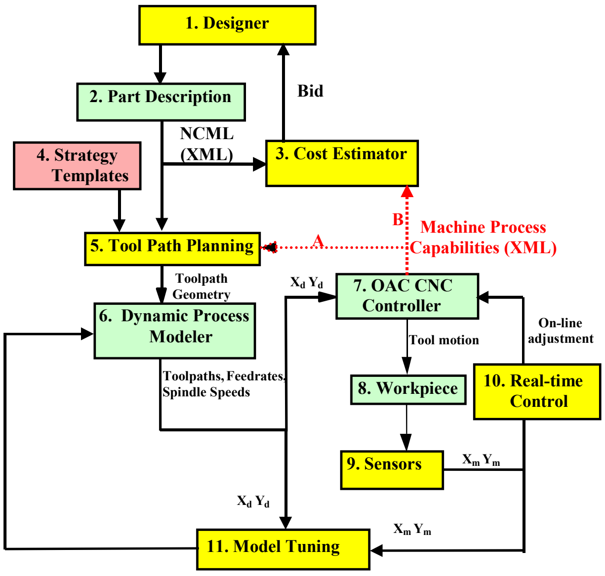

Figure 2 Structure of a Smart Machining

which adjusts feedrates to compensate for process degradation such as tool wear.

.

The sensor data is compared with process modeler data for Model Tuning (**11**). In this way, sensory input is interpreted based on the expectations of the process model. Furthermore, deviations between expected and measured data can be used to tune the model to improve its accuracy. Note that we have not yet done any meaningful research on Strategy Templates (**4**) and Tool Path Planning (**5**).

4 FEEDRATE SCHEDULING

This paper describes how the system can be used to perform feedrate scheduling. Other publications describe on-line calibration methods [18, 20], runout estimation [23] and tool condition monitoring [19]. The choice of feedrates affects both part quality and process efficiency.

Feedrate selection is achieved in our SMS by analyzing the existing program and adjusting feedrates to achieve uniform quality and maximum efficiency [12, 13, 20]. Our program implements feedrate adjustment by accessing the shared memory of the MDSI controller to adjust the feedrate override parameter. The feedrate adjustment is the equivalent of adjusting the feedrate override pot on the front panel of the CNC control panel, except that the "hand" turning the dial is our software automatically analyzing and adjusting.

An industrial partner supplied us with tooling and stock for a "best practice" part program developed by skilled, experienced programmers for a high volume production part. Our program reduced peak forces by 40% (eliminating a broken tool issue) while simultaneously reducing production time by 7% as shown in **Figure 3**. [18, 19].

This example illustrates how the SMS can combine sensors and models to improve the process during production runs. In this way, the model is used to optimize the cutting process (Block 6 in **Figure 2**) and observation of the cutting process is used to improve model accuracy through on-line calibration (Block 11 in **Figure 2**). It is important to note that these synergies can only be achieved with an OAC in which experimental measurements are synchronized with the model estimates. It is therefore critical that there be a tight communication between the machine controller and application programs. In our case, we achieve this through the shared memory capability of the MDSI.

5 SUMMARY

A testbed has been assembled at UNH for performing research on Smart Machine Tools. It uses a commercially available OAC from MDSI, geometric modeling software from Predator along with a number of modules developed in our lab. A high bandwidth Kistler load cell, LCI power sensor and AKG contact condenser microphone provide measurements of cutting forces, motor spindle power and tool vibrations respectively. On-line calibration allows the SMS to fine tune model parameters which can then be used to improve production efficiency as the machine "learns" its own capabilities. An XML database (NCML) is used to

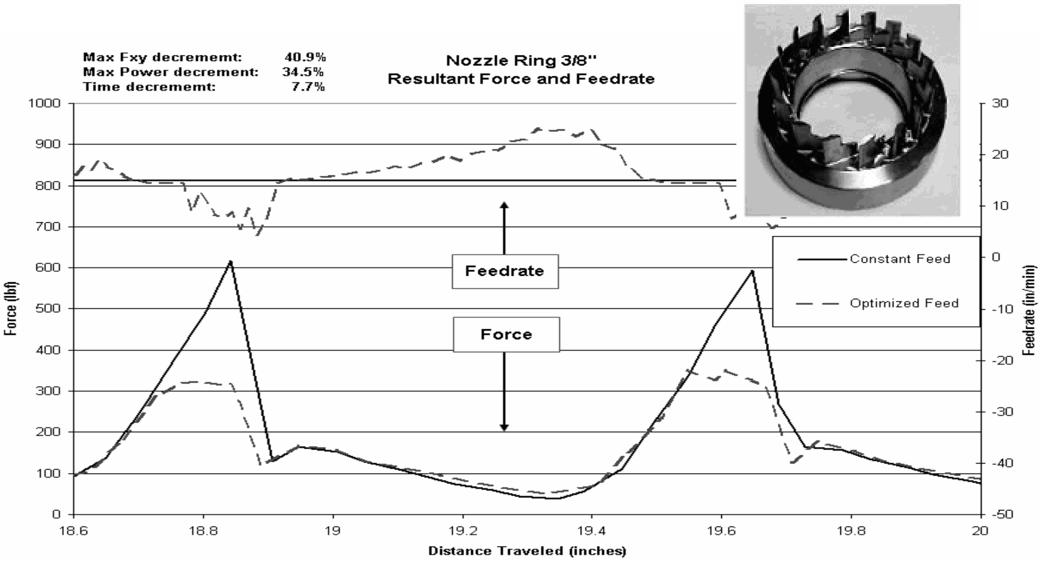

Figure 3. Total Resultant Force and feedrate profiles for 3/8" tool cutting two blades of the nozzle ring part, original constant & variable feedrates shows a 40% decrease in peak forces and a 7% decrease in cutting time [xxx]

represent part programs, machine tool characteristics and store machining history [17].

6 ACKNOWLEDGMENTS

The support of the National Science Foundation under grants DMI-9713906, DMI-9872575 DMI-0322869 is gratefully acknowledged.

7 REFERENCES

1. NCMS, National Center for Manufacturing Science, http://ncms.org/
2. NIST, (2002) National Institute of Standards and Technology, The Smart Machine Tools Technology Workshop, Dec. 9-11, Gaithersburg, MD. http://www.mel.nist.gov/proj/sms.htm
3. MDSI Inc., Ann Arbor, MI, http://www.mdsi2.com/
4. OMAC, Open Modulate Architecture Controllers, http://www.omac.org/
5. OSACA, Open System Architecture For Controls Within Automation Systems, http://www.osaca.org/
6. OPC, OLE for Process Control, (note that OLE stands for Object Linking and Embedding, http://www.opcfoundation.org/.
7. LCI: Load Controls Incorporated, http://www.loadcontrols.com/.
8. Schmitz, T.L., (2003) "Chatter Recognition by a Statistical Evaluation of the Synchronously Sampled Audio Signal," Journal of Sound and Vibration, 262(3):721-730.
9. Jerard, R B. , Barry K. Fussell, Min Xu, Chad Schuyler, Donald Esterling, "Development of a Testbed for Research on Smart Machine Tools, ", Proceedings of 2006 NSF DMII Grantees Conference, St. Louis, July 24-27, 2006
10. VenturCom http://www.vci.com/
11. Predator Software Inc., Predator Virtual CNC, http://www.predator-software.com/virtualcnc.htm
12. Fussell, B.K., R.B. Jerard and J.G. Hemmett, (2001) "Robust Feedrate Selection for 3-axis Machining Using Discrete Models," ASME Journal of Manufacturing Science and Engineering, Vol. 123, No. 2, pp. 214-224.
13. Fussell, B.K., J.G. Hemmett and R.B. Jerard, (2003) "Modeling of cutting geometry and forces for 5-axis sculptured surface machining," Computer Aided Design, vol 35, pp. 333-346.
14. Jerard, R.B. and Okhyun Ryou, (2002) "NCML: A Data Exchange Format for Internet Based Machining," 2002 Japan USA Symposium on Flexible Automation (JUSFA), July 15-17, 2002 Hiroshima, Japan.
15. Jerard, R.B. and O.H. Ryou, 2006, "NCML: a data exchange format for internet-based machining," International Journal of Computer Applications in Technology (IJCAT), Vol. 26, No. 1/2.
16. Jerard, R.B.; Fussell, B.K.; Xu M.; Yalcin, C.: Process Simulation and Feedrate Selection for Three-axis Sculptured Surface Machining, International Journal of Manufacturing Research, 1(2), 2006, 136-156.
17. Schuyler, C.K., (2005) "Feedrate Optimization and Tool Condition Monitoring of Flat-End Milling Operations Utilizing Spindle Motor Power, M.S. Thesis, UNH ME Department.
18. Schuyler, C.K., M. Xu, R.B. Jerard and B.K. Fussell, (2006) "Cutting power model-sensor integration for a smart machining system," Transactions of the North American Manufacturing Research Institution/SME Volume 34, NAMRC 34, Marquette University, May23-26.
19. Xu, M., C.K. Schuyler, B.K. Fussell, R.B. Jerard, (2006) "Experimental Evaluation Of A Smart Machining System For Feedrate Selection And Tool Condition Monitoring," Transactions of the North American Manufacturing Research Institution/SME Volume 34, NAMRC 34, Marquette University, May23-26.
20. Yalcin, Cuneyt, Barry K Fussell, Yigit Erzan, and Robert B Jerard, (2004) "Real-Time Calibration Of Cutting Force Models For CNC Milling", Proceedings of 2004 Japan – USA Symposium on Flexible Automation, Denver, Colorado, July 19-21.
21. STEP-NC, ISO/DIS 14649-1, Industrial automation systems and integration –Physical device control-Data model for computerized numerical controllers : http://www.step-nc.org/.
22. Choi, B and R. Jerard, (1998) Sculptured Surface Machining - Theory and Applications, Kluwer Academic Publishers, ISBN 0-412-78020-8
23. Jacobson, D, B.K. Fussell and R.B.Jerard, 2006 "Tool Runout Estimation Using Feed Force Spectral Components", submitted to the 2006 ASME International Conference on Manufacturing Science and Engineering, Oct 8-11, Ypsilanti, MI.