SELF-SUSTAINING, OPEN-SYSTEM MACHINE TOOLS

by

Israel Greenfeld
Associate Research Scientist

Fred B. Hansen
Assistant Research Scientist

Paul K. Wright
Professor of Computer Science

Robotics Research Laboratory
Computer Science Division
Courant Institute of Mathematical Sciences
New York University
719 Broadway, 12th Floor
New York, NY 10003

ABSTRACT. The theme of this research is the creation of "self-sustaining, open-system machine tools" for small-batch manufacturing, with the vision of the "machine tool of the future" in mind. The self-sustaining machine tool entails a rich environment of design and planning capabilities, expert systems, dextrous manipulators and sensors. As an open-system, it has communications ability, a modular configuration and a universal architecture. Specifically, a prototype design and manufacturing environment based on this theme is described and the development of an open-system controller for machines is discussed as a necessity for the implementation of the proposed diversified environment.

INTRODUCTION. With the eventual goals of reducing labor costs and increasing quality in U.S. small-batch manufacturing industries, and acknowledging the critical role of machine tools in manufacturing, new approaches to the design of future machine tools should be conceptualized. We envisage two basic descriptors for the next generation of machine tools: **self-sustaining**, and, **open-system**. These two complementary features should be considered, balanced and integrated in the construction of future machine tools. A variety of sub-features evolves from these design criteria, as illustrated in Figure 1.

The rising power of standalone universal computer workstations, which have a considerable local computing power and a very strong communications network support, provides an opportunity for the creation of advanced machine tools. Such computer workstations enable more local performance, in terms of sensory and manipulative devices, design and planning capability and computing power. Also, they especially encourage a universal open architecture that can enhance communications and flexibility. This represents an opportunity for the machine tool industry to benefit from the innovations and advances in the computer industry.

In order to be self-sustaining a machine tool will be equipped with dextrous manipulators dedicated for the continuous needs of machining, such as chip clearance and part relocation. In future commercial systems a variety of sensors will provide vision, touch, force and temperature senses, with the task of recognizing events, in-cycle inspection and optimizing the machining parameters. A rich supporting design environment is essential, including:- Cadcam for part and tool path design; expert systems and libraries of technical information for an optimal and efficient design; and an operating environment with process planning capability.

In order to be an open-system a machine tool will be equipped with a general purpose computer, such as a PC or a workstation, that will control the axes of motion and the various devices, and will manage programs and data locally. Communications will use networks that are universally accepted in the computer world, and thus resources may be shared as needed. The machine should be adaptable to the changing environment and tasks, in terms of its controller's computer configuration, and in terms of the mechanical construction.

Several research institutions and industrial organizations have been developing research methods that are concerned with creating unattended, intelligent machine tools. Some of the individual components of such unattended machine tools include: expert systems for setup planning [1]; appropriate machine control architectures [2]; sensors for machining [3]; and automated fixturing devices [4]. These describe the theoretical aspects of the work, typical practical implementations, and some simple demonstrations [5,6].

Our research is concerned with the integration and implementation of the complete manufacturing cycle, as well as with the development of some novel devices that allow reliable automation of small batches. We see this laboratory development as a mechanical analog of the MOSIS system [7]. Within the bounds of a local area network, we envisage that designers will ship their files to a machining environment where the automated mechanical system will generate the part and return it to the designer, similar to the spirit of 'desktop' manufacturing [8].

Sections 2 and 3 of the paper describe, from an implementational viewpoint, the general laboratory environment and some of the new hardware and software developments that comprise the system. Section 4 contains a conceptual discussion of the advanced controller that is suggested for future machine tools.

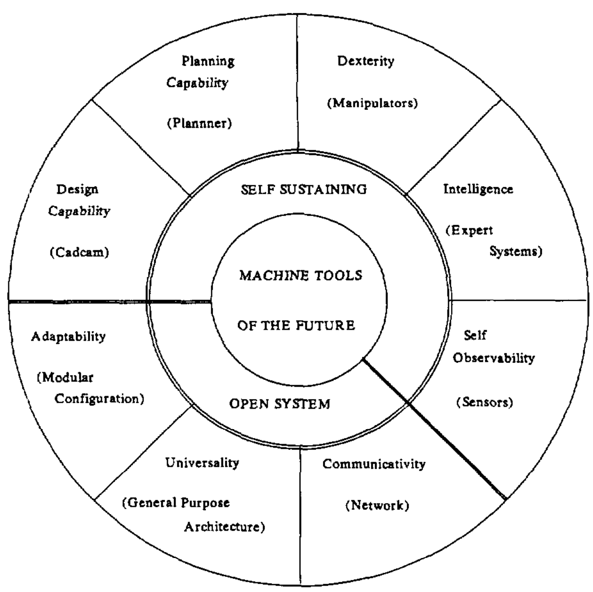

Figure 1. Features of Self-Sustaining, Open-System Machine Tools

LABORATORY OVERVIEW. The current laboratory environment is captured in Figure 2 and described in [9,10].

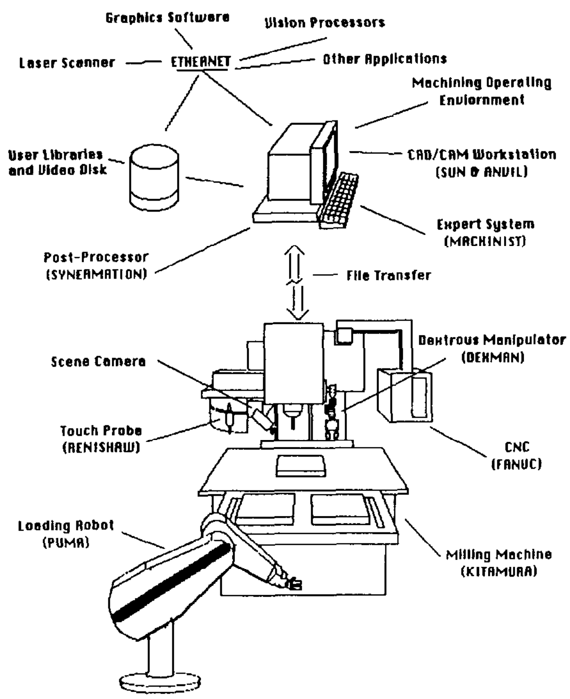

Figure 2. Hardware and Software Overview

The machining cell is built around a machining center that is supported by a Cadcam system, a computer network and an operating environment. The cell is currently being enhanced with robotic, vision and expert systems. The current configuration includes the following components:

- Kitamura MyCenter 1/485 machine tool, 3 axis, 2 pallet.
- Fanuc 10M controller.
- Sun 3/60C color workstation, networked to the laboratory's Ethernet.
- MCS Anvil-5000 Cadcam software, with 3D design/drafting, solids, IGES and NC.
- Renishaw MP9 touch probe, with macro routines.
- Synermation MFNC01 post processor software.
- Dextrous manipulator (Dexman).
- Scene monitoring cameras.
- Cell operating environment software.
- Cadcam cell utilities.
- Users' libraries, including Anvil's users group, CarrLane and Metcut.

This configuration currently provides basic design and manufacturing capabilities. To demonstrate these capabilities let us observe a typical design and manufacturing sequence for an arbitrary part:

The designer designs a part on the Cadcam system using wireframe, surfaces and solids construction methods. An expert system provides support in making machining and planning decisions. A number of engineering databases enable the designer to browse through libraries of downstream manufacturing data on tools, fixtures, standard parts and manufacturing processes. The technical information may be read onto a Sun-window and used in real time for design. Video disk databases, currently in development, contain images of real parts and devices to be displayed. The designer, or a manufacturing engineer, then prepares the machining tool paths (Figure 3) on the Cadcam system using the part's geometrical model. An expert system helps in planning the process setup. A number of programming tools and NC macros (Figure 4), such as the probe routines described later, are available for custom applications. A machining output file, in APT format, is created and translated into the Fanuc controller's machine code, using the semi-custom post processor. The post processor has special procedures to translate probe programs.

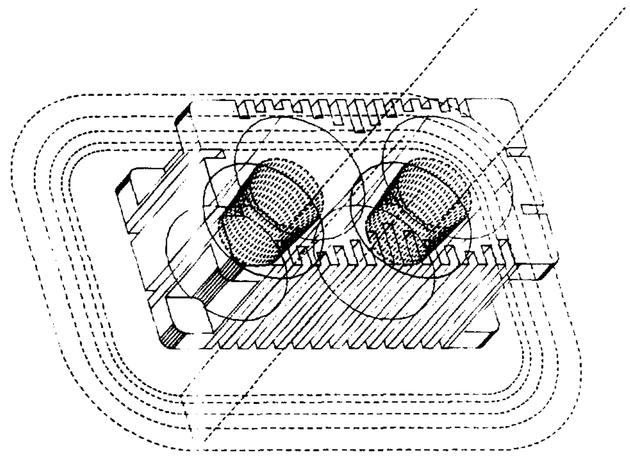

Figure 3. A Sample Component: the Throttle from the Engine of a Speedboat, with Tool Paths

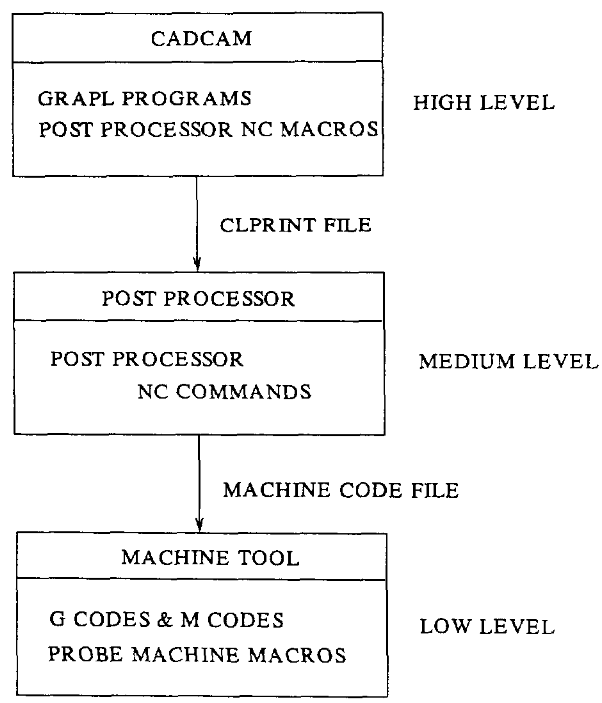

Figure 4. Levels of Programming

The machining and probing files are handled and merged by the cell operating environment, and then sent to the machine tool for execution. The operating environment also controls the operation of other devices, such as pallets and the probe. Upon completion of a machining step, the part is gauged with the probe while still on the machine tool, using a program prepared on the Cadcam. The results are read back to the Sun computer, and serve for a quality control report and for part localization. The dextrous manipulator (Dexman) is mounted on the machine tool and selects tools from a special magazine. It performs functions that would normally be carried out by a human machinist, such as part relocation, chip clearance and carrying a CCD camera for tool wear inspection.

NEW SOFTWARE AND HARDWARE.

TOUCH-TRIGGER PROBE ROUTINES. The touch-trigger probe provides the machine tool with a touch sense, supplying data for inspection and for automatic update of work coordinates and tool offsets. The probe operating cycles (also known as In-Cycle-Gauging), include tool setting, tool wear monitoring, workpiece identification, workpiece set up and workpiece inspection. The probe is shank mounted and is stored in the machine's tool magazine. It is automatically loaded on the machine spindle for a gauging cycle.

The accuracy of the probe depends on the accuracy of the machine tool, the calibration of the probe, the approaching speed and direction and on other factors [11]. In our installation a repeatability of 0.0002" and an accuracy of 0.0005" are estimated under nominal conditions. This level of accuracy may enable gauging of a large variety of mechanical parts (although excluding very high precision parts).

In-Cycle-Gauging is an essential feature in the implementation of unattended manufacturing, and in particular in the manufacturing of one-of-a-kind parts. This kind of production entails flexible fixturing systems and requires in-line quality control. The position of the part on the machine tool table will typically be inaccurate. In-Cycle-Gauging provides inspection and correction in a single set up and supplies the spatial position data needed for part localization.

The operation of the probe involves a variety of routines. The low level probe motion and gauging functions are driven by machine macros which are resident in the controller's memory. The medium and high level codes, developed during the project, are special Grapl programs and post processor NC macros [10]. The Grapl programs provide an Anvil-like user environment that eases the specification of parameters and enables the selection of features on the part's model. The probe paths are created automatically by the post processor NC macros, called from the supervisory Grapl program.

A probe path includes the probe loading and starting, the protected rapid motion prior to gauging, the positioning of the probe at a gauge-start location, and the invocation of a standard gauging cycle. The measurement results are recorded and sent back to the host computer for further use.

The system provides three major probe applications: dimensional inspection, setup calibration and part localization. Dimensional inspection is executed after completion of a machining step, without altering the part setup. Standard gauging functions are included such as measuring bore, boss, web, pocket, top surface and side surface, as shown in Figure 5. If a tolerance of a gauged feature is exceeded, a tool offset or a work offset may be updated for use later on subsequent parts or passes. A pass may be repeated if necessary.

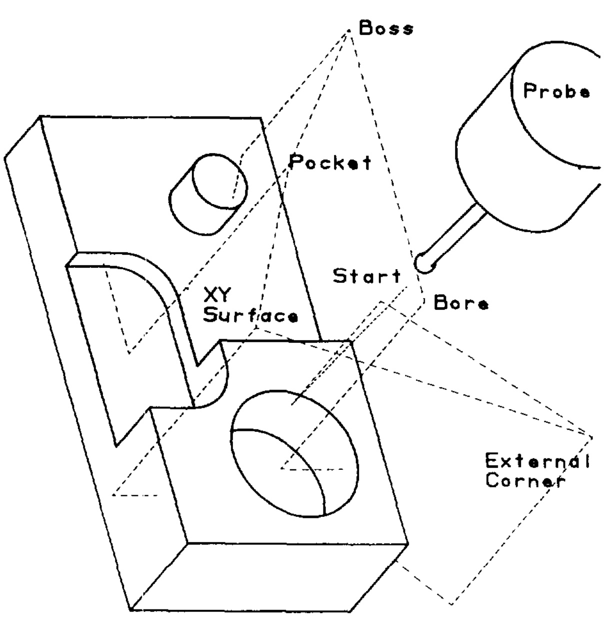

Figure 5. Probe Path on a Sample Part

Setup calibration is done prior to fixturing the part, by gauging the workholding reference devices, such as the fixed face of a vise, and updating the relevant machine coordinate system. Two additional functions, datum in external corner and datum in internal corner, are used for this. This type of compensation is possible in translation only.

Part localization is done after fixturing the part and prior to machining, by gauging the part at a few locations and determining its misalignment (translation and rotation) with respect to the machining coordinate system. Such inaccuracies in workholding are typical of one-of-a-kind machining. For prismatic parts, the six point gauging method is used: three points on a primary plane, two on a secondary plane, and one on a tertiary plane. Refer to Figure 6 for an example. The data is sent to the host computer, and a transformation matrix is computed and applied to subsequent machining files.

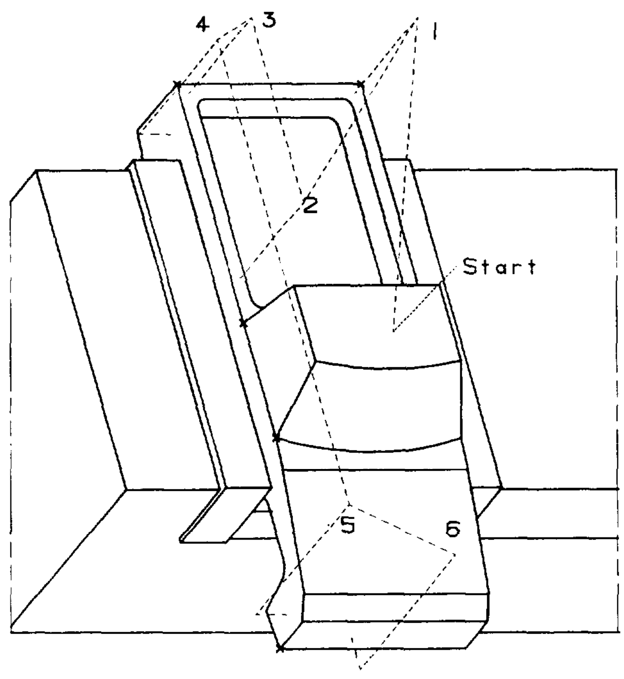

Figure 6. Spatial Localization of a Model Truck

AN EXPERT SYSTEM FOR SETUP PLANNING. The designer of a part is usually required to exhibit some knowledge of machining skills and judge the best way of setting the part up and the various cutting sequences. However, this setting up and ordering process is, today, generally the function of the machinist rather than the designer. To avoid this "split of duties" we have been developing an expert system that begins to build the machinists' setup rules into the design phase.

In the work so far, the program - called **Machinist** - comprises approximately 180 OPS5 rules. Experienced machinists have judged the program to make better plans, on average, than a journeyman with five years of experience, (within the limited domain of rectangular parts, of the type shown in Figure 3, that are machined on a three axis machine.) **Machinist** is a high level process planner that groups features into setups and orders these setups. It does not divide features into all their component operations, plan cutter part or choose feeds and speeds - however, the standard Anvil-5000 NC package, in conjunction with a computerized machining handbook such as Metcut's, can accommodate such needs.

The expert system merges two constraints, namely -

- The squaring-up process
- The feature interaction considerations

This merging process is shown in Figure 7. On the right hand side, the planning software must create an orthogonal block, otherwise the features will not be placed accurately relative to each other. This squaring process always has to occur whatever the part geometry. This was concentrated on in the earliest work, and the various squaring-up plans are shown in [1].

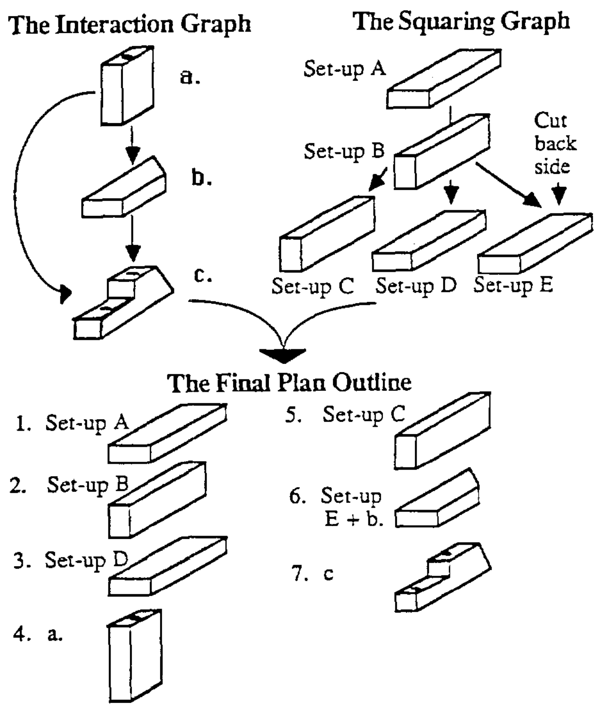

Figure 7. Merging the Squaring-Up and Feature Interaction Graphs

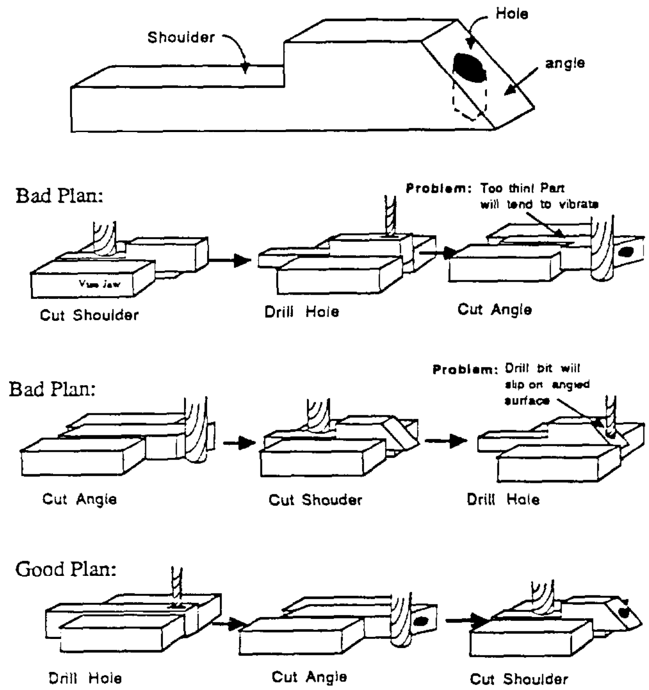

Figure 8. Examples of Planning Showing Two Bad Plans and the Final Good Plan

It is the merging of the features into the squaring up plan that is the challenging aspect of human craftsmanship and, consequently, of the creation of an expert system rule-set that can mirror the deep human experience. The reason this is difficult is that cutting one feature may make it very hard or impossible to cut subsequent ones. This is best explained by referring to Figure 8. This shows two "bad" plans above the "good" plan. The bad plans are ones that a human machinist would avoid based on experience: one of these plans causes too much vibration in the last step and the other causes the drill point to skid on the surface of the chamfer. The good plan carefully orders the setups in such a way as to avoid these problems. The way in which the setup sheet is presented to either a novice machinist or an automated machine tool is thus shown in Figure 7. When the squaring-up is combined with the feature interactions, seven setups are needed to complete the part.

The software's 180 OPS5 rules contain many heuristics, learned from two experienced machinists. In many cases the program uses built-in, generalized patterns to identify the interactions quickly. For example, for the hole/angle interaction, the program has a pattern that matches any drilled hole or depression that enters a non-orthogonal surface. If the pattern is matched, then it puts a restriction on the plan; namely, that the hole must be cut before the non-orthogonal surface. Similarly, for each additional pattern, there is an associated operator that tells the program how to avoid the interaction. These are termed "feature interaction graphs". Although many of these have been identified, one important aspect of the continuing research is to expand the protocol analyses with the machinists, and obtain more feature interaction graphs that reflect a wider range of typical parts.

A MACHINING OPERATING ENVIRONMENT. A supervisory machining operating environment, developed during the progress of this project, resides in the host computer (a Sun workstation), and controls the activities in the machining cell. The system uses the Unix operating system and the C language as the tools for writing the cell commands and utilities. Communication with the machine tool is done via dual RS232C serial lines. The host computer is connected also to other computers in the laboratory via Ethernet, thus providing a high degree of flexibility in computer resources and communication.

The machining operating environment manages the cell communications, the preparation of machining files, the execution of standard operations such as pallet handling and periodical calibration, the management of the controller's tools, coordinates and configuration, the transformation of coordinates, inspection reports' preparation, and other tasks. The system also manages the libraries of machining files (through Unix) and provides safety features and an authorization scheme. The system is highly flexible due to the use of Unix and C. New commands are currently added as the needs expand.

Machining processes are programmed as Unix script (batch) files, using the operating environment commands along with Unix commands. Thus, a complete process sheet may be programmed into a script, having as its input the data supplied by an expert machinist system such as **Machinist**. The Unix script file also enables control functions, which allow conditional branching during the process, e.g., the output of a probing cycle may be analyzed, and a decision may be selected from a set of available courses of action, to determine if a corrective tool path is needed, or if the process should be aborted, and so forth. Complex decisions and large numerical calculations can be invoked from the script as external processes, and the results returned to the script.

THE DEXTROUS MANIPULATOR (DEXMAN). The role of the dextrous manipulator shown in Figures 9 and 10 is *i)* to relocate a part on its fixture for each new setup during machining, and *ii)* to utilize a variety of miscellaneous accessory tools needed for cleaning and viewing parts and cutting tools. As shown in the drawing in Figure 9, the dextrous manipulator is a three-degree-of-freedom, hydraulically-actuated manipulator with a parallel jaw gripper. It uses a wrist and a gripper of an IBM RS1 robot. The swing arm shown at the top of Figure 9, is mounted on the spindle housing of the machine tool, and swings the manipulator into the working space when needed. Figure 10 shows the dextrous manipulator turning over a part in the vise, and the relationship between the dextrous manipulator and the machine tool itself. The dextrous manipulator can also be loaded, with some modifications, on the spindle itself from the cutting tools magazine of the machine tool. The jaw of the gripper contains force and optical sensors that help to modulate the grasping force and to check that the part has not, for some reason, been dislodged from the gripper.

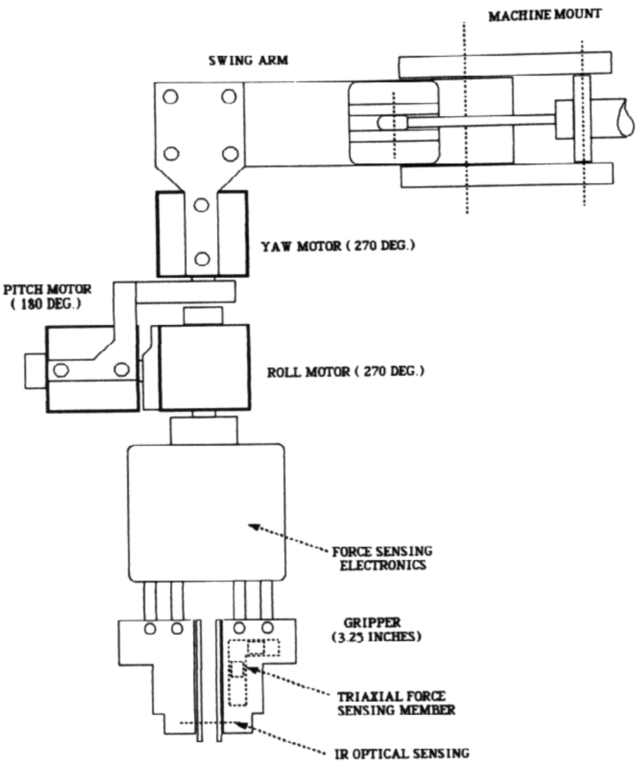

Figure 9. A Dextrous Manipulator for In-process Machining Manipulations

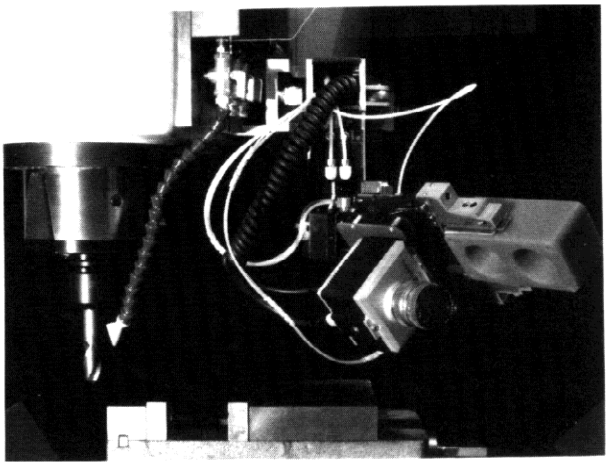

Figure 10. The Dextrous Manipulator Mounted on the Machine Tool Spindle Housing - Turning Over a Part

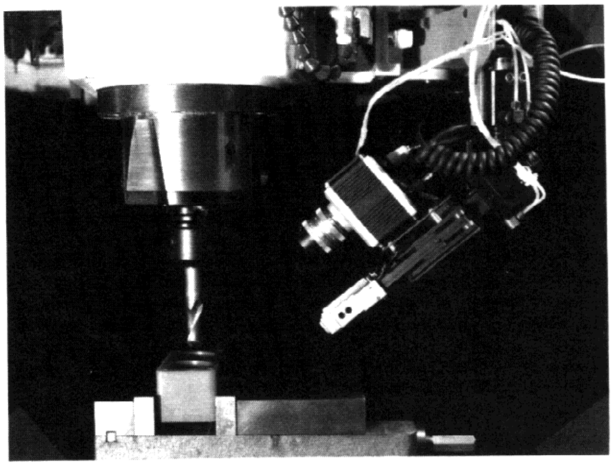

Figure 11. The Dextrous Manipulator Holding a CCD Camera Inspecting a Cutting Tool

The dextrous manipulator is being setup, at the time of writing, for the above operations, and during this initial phase, it is being configured for remote control, using the data acquisition and control systems **Asyst** (on an IBM-PC) and **LabView** (on an APPLE-Macintosh). Future advances in machine tool controller technology, as discussed in the next section, will make it possible to concurrently control the dextrous hand and the machine tool motion, to achieve the higher coordination needed for some operations.

We are currently constructing a small magazine of ancillary tools that can be picked up by the dextrous manipulator. These can be used in tasks that, although appearing rather ordinary, are crucial elements for the successful machining of one-of-a-kind parts. The magazine contains an air-wrench for tightening fixtures, and, as shown in more detail in Figure 11, a CCD camera that is connected to a Vicom vision system for analyzing the state of a worn tool between passes.

The magazine also contains an air-gun to blow away chips, and a brush to further aid chip-clearance and to remove tangled chips from the end of drills. Chip clearance obviously becomes an important issue in unattended machining. In our implementation work, as much care as possible will be taken in the design phase to create controllable chips. For example, this will include the selection of free-machining alloys, the use of mechanical chip-breakers and contoured inserts to control chip length and curl, and the selection of peck-drilling cycles. Fixturing devices must be designed and selected to minimize blind corners and/or recesses where chips might lodge. It is difficult to imagine that current computer vision systems can be setup to provide a quick glance for the possibility of lodged chips, in the same way as a machinist does. The above *a priori* precautions, and thorough air and brush cleaning will thus be critically important but, even so, may not eliminate every single chip from the fixture.

The possibility of a 'stuck' chip places additional emphasis on touch-trigger probing to ensure mating of part-surfaces against fixture-surfaces, and, in critical situations, it will create the need for simple sensors in fixtures to detect uneven part-mating. If these devices reveal that chips are lodged then the part will have to be removed and cleaning repeated or, as a last resort, the system will have to be closed down awaiting a human operator. Chips appear mundane: but we are reminded of the story that begins "for want of a nail a horseshoe was lost". A renewed interest in the physics of chip formation, as it relates to the control of chip length and curl for unattended machining could now, more than ever, be a fruitful area for research.

SENSORY ENVIRONMENT. It has been shown [5] that computer vision is a viable inspection tool for examining the flank wear land on milling cutters, and for judging the percentage of remaining tool life. With the aid of the dextrous manipulator a vision system can get a clear view of these worn areas. Vision works well in this situation because the camera does not have to search for the position of the cutting tool in the scene - it is always in the same place at the end of the spindle. Also, the tool is stationary and the front-lighting of the scene is relatively easy to control. In addition, because the mechanics of tool wear have been developed over several years, there is a well established model-base of the way in which the tool deteriorates and how it can be analyzed.

In the future, it may be appropriate to incorporate other sensors, e.g. dynamometers, thermo-couples, and torque sensors into the environment, thus enabling adaptive control of the machining process. Reviews of the various sensor types and references to other work are given in [3-6]. Realizing that machining is a multi-dimensional problem and that human machinists use a multitude of sensors, investigators have more recently focussed on sensor-fusion strategies [5]. For example, Balakrishman and colleagues [12] present a turning tool holder with a combined acoustic-emission/force-measurement sensor package. Their results show that this combination of sensors is a more reliable tool monitoring system than either one of the sensors individually. Their pattern-recognition analysis of the continuous and transient chip formation signals leads to a positive identification of tool breakage with a reported 97% successful classification rate.

So far, our experience has shown that these "real-time" sensors are difficult to apply in the specific case of one-of-a-kind machining operations. The reason is that most real-time machining sensors (used either alone or as a sensor fusion package as in [12]) require a supporting knowledge base that learns, over a series of parts, the safe, or normal operating range for the sensor data. With one-of-a-kind parts, uncharacterized work materials, and new tools and fixtures, it is difficult to establish this safe operating range for the sensor. In addition, only by offering a more open architecture, can machine tool controllers supply proper provision for a flexible integration of real-time machining sensors. This is the subject of the next section.

So far, our response has been to minimize the use of such real-time machining sensors and set cutting speeds and feed rates conservatively low with the assumption that the tool will not break or rapidly wear during one particular pass of the tool. Currently we rely on the vision system to provide information on tool wear, and on the touch-trigger probe for dimensional inspection of the part.

A DISCUSSION ON OPEN-SYSTEM CONTROLLERS. Today's Computer Numerical Controllers (CNC's) have their roots in Programmable Logic Controllers (PLC's). Thus, while CNC's are rugged and adequate for servo control of motion axes, they are very limited in terms of programming flexibility and in terms of communications with external computers and devices. They cannot accommodate non-machining devices such as workholding accessories, force sensors, vision sensors, and other subsidiary devices. 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.

Such machines are currently programmed in a very low level machine language which is tedious to use and hard to keep error-free. Thus, machine-specific post processors are needed to translate a Cad output to the machine code format acceptable by a particular controller. The CNC's can read and execute machining files sequentially as sent, block by block, but they cannot be easily controlled from an external computer, for purposes of coordinating operations with the machine's accessory devices.

We suggest that a reconfiguration of the CAD/CNC machine tool environment should be implemented [13], involving a more open-architecture system, based on mainstream well-established computer technology such as the PC or the workstation. Such an architecture will provide a more efficient and user friendly environment for operation and programming, the ability to integrate various devices with the machine tool's construction and operation, and the ability to communicate more tightly with Cadcam systems and factory wide management systems. Also, with the current price-performance trends in the general purpose computer industry, a reduction in the controller's price should be expected. The three main research areas associated with this open-architecture system include:-

i) The development of a *Real Time Operating System for Manufacturing*, suitable for the very high speed control required for machining operations. Current general purpose real time operating systems do not provide the response time essential for maintaining the speed, accuracy and safety features inherent to machine tools. The operating system should interface to industry standard operating systems such as Unix or OS/2, that will provide high level management, filesystem operations, communications, and a good programming environment. We plan to extend the work already done on SAGE (NYU) [14], a real time supervisory operating system for robot control, and to investigate the possibility of using other systems such as Condor (MIT) and Sparta (IBM) [15].

ii) The development of *Advanced Machining Language* for machining programming, that will reside in the open-system controller. While existing languages such as APT and Compact will be supported, a more flexible language is needed. It will include provisions for real time control needed for the operation of accessory devices in conjunction with the machining process, a more direct connection to Cadcam systems, and a flexible interface for user applications.

iii) The integration of the *Open-System Machine Tool Controller*. It will include the installation of new devices and sensors, such as those described in this paper. These devices are part of the machine-specific configuration, and will be controlled locally by a prototype of the open-system controller. The prototype controller will be based on a high-end PC or a Sun workstation, with additional control processors plugged into a shared bus, and running the real time operating system. This configuration provides a universal and modular installation, with all the components sharing the same high level operating system and programming environment, communication facilities and other computer resources.

These ideas are now being conceived under the working name 'MOSAIC' (Machine-tool's Open-System, Advanced, Intelligent Controller) [13]. The research is intended to investigate the broad implications of such a controller on machining languages and operating systems, as well as on integration with machine tools and devices.

CONCLUDING REMARKS. The combination of powerful design and planning tools, supporting technical data, real time gauging and inspection, and hardware flexibility, yields a much greater chance of creating a process that leads to "the first part right the first time." This paper presents the initial installation of a machining cell based on these principles. Important components are *i)* the touch-trigger probe for in-cycle part localization and inspection, *ii)* the expert system for setup planning, *iii)* the dextrous manipulator for workpiece handling and viewing, and *iv)* the machining operating environment. We have emphasized the idea of a self-sustaining, open-system machine tool, controlled by an advanced workstation based on current general purpose computer technology.

ACKNOWLEDGEMENTS. This work is supported by Office of Naval Research Grant N00014-82-K0381 and National Science Foundation Grants DCR-8320085, and DMC-8611150. We thank John Deragon, Jim Fehlinger and Elias Pavlakos for their assistance with programming and acknowledge our collaboration with David Bourne and others at Carnegie Mellon University.

REFERENCES

(1) C. Hayes and P.K. Wright, Automated Planning in the Machining Domain, *Knowledge Based Expert Systems for Manufacturing*, American Society of Mechanical Engineers, 1986, PED-Vol. 24, pp. 221-232

(2) D.A. Bourne, CML: A Metta Interpretor for Manufacturing, *AI Magazine*, 1986, Vol. 7 No. 4, pp. 86-96.

(3) J. Tlusty and G.C. Andrews, A Critical Review of Sensors for Unmanned Machining, *CIRP Proceedings*, 1983, Vol. 32 No. 2, pp. 563-575

(4) F.B. Hazen and P.K. Wright, An Autonomous, Self-Loading Machine Tool, *Proceedings of the Manufacturing '88 Conference*, American Society of Mechanical Engineers, Atlanta, GA, 1988

(5) P.K. Wright and D.A. Bourne, *Manufacturing Intelligence*, Addison-Wesley, Reading, MA, 1988

(6) J.A. Simpson, R.K. Hocken and J.S. Albus, The Automated Manufacturing Research Facility of the National Bureau of Standards, *Journal of Manufacturing Systems*, 1984, Vol. 1 No. 1, pp. 17-32

(7) USC, *MOSIS User Manual, Release 3.0*, University of Southern California, Los Angeles, CA, 1988

(8) D.P. Colley, Instant Prototypes, *Mechanical Engineering*, July 1988, pp. 68-70

(9) I. Greenfeld, *NYU Machining Cell Users' Manual*, NYU - Courant Institute of Mathematical Sciences, New York, NY, 1988. (to be published)

(10) I. Greenfeld, *Probe Programming Users' Manual - NYU Machining Cell*, NYU - Courant Institute of Mathematical Sciences, New York, NY, June 1988, Technical Report No. 384

(11) J.A. Balash, T.W. Coffield and R.E. Monley, *An Evaluation of Contact Sensors for Unmanned Machining*, Carnegie Mellon University, Pittsburgh, PA, 1985, (summer project)

(12) P. Balakrishman, H. Trabelsi, E. Kannatey-Asibu and E. Emel, A Sensor Fusion Approach to Cutting Tool Monitoring, *Proceedings of the 15th Conference on Production Research and Technology*, University of California at Berkeley, January 9-13, 1989

(13) I. Greenfeld, MOSAIC - A Machine-Tool Open-system Architecture Intelligent Controller, *NYU - Courant Institute of Mathematical Sciences*, New York, NY, July 1988, A Technical Note

(14) L. Salkind, The Sage Operating System, *NYU - Courant Institute of Mathematical Sciences*, New York, NY, August 1988, (to be published)

(15) J. Ish-Shalom and P. Kazanzides, Sparta: Multiple Signal Processors for High Performance Robotic Control, *Proceedings of the 1988 IEEE International Conference on Robotics and Automation*, IEEE, April 24-29, 1988