Forth and Automation Research at The National Bureau of Standards

William G. Rippey

U.S. Department of Commerce
National Bureau of Standards
Washington, D.C.

Abstract

The National Bureau of Standards (NBS) is conducting a large research project in industrial automation. Automated control of manufacturing processes will be achieved through extensive use of computers. Some of the individual projects implemented in FORTH will be described. Some FORTH techniques used on each project will be highlighted.

The National Bureau of Standards (NBS)

NBS is located in Gaithersburg, Maryland, 24 miles northwest of Washington, D.C. NBS was established by Congress in 1901 as an agency of the Department of Commerce. It is the national measurement laboratory in the physical and engineering sciences. Through the years the research activities of NBS have expanded, by Congressional mandate, into the areas of the fire safety of buildings, computer standards, energy use, materials use, and industrial productivity, to name a few.

The NBS Automation Program

The research in industrial productivity is being carried out by the Center for Manufacturing Engineering (CME). The goals of the research are embodied in the major project, the Automated Manufacturing Research Facility (AMRF). The AMRF is a prototype factory that is being built at NBS. It will perform all of the control and production functions of an automated machine shop. Some of these functions are: generation of CAD/CAM data files describing part geometry, generation of control data for machining, inspection, and materials handling operations, scheduling production of batches, allocation of factory resources, monitoring of factory status, and machining of batches of parts in unattended workstations.

To date, we have installed two NC (Numerical Control) machining centers, one NC lathe, and an industrial robot. Another lathe and 3 additional robots will be installed in the near future. In January, 1983 a workstation consisting of a machining center, robot, automated part fixturing, workstation controller, and a wire guided cart demonstrated unattended production of batches of metal parts.

Our purpose in implementing the AMRF is to test control concepts and to develop techniques in metrology and standardization for the automated manufacturing environment.

Distributed Numerical Control System (DNC)

The NBS DNC system stores part program files for numerically controlled machine tools. The files are images of paper tapes that were used before the DNC was built. The part programs, which control cutter motions during machining, can be downloaded to the machine tools from the DNC via RS-232 links. See Figure 1. An office terminal is used to write and edit high level APT (Automatically Programmed Tool language) programs that are stored locally and then sent over telephone modem link to a remote timesharing service that processes the APT into machine level part programs. Utility words used at the terminal request the timesharing service to transmit part programs to the DNC for storage on

*Received August 1983.

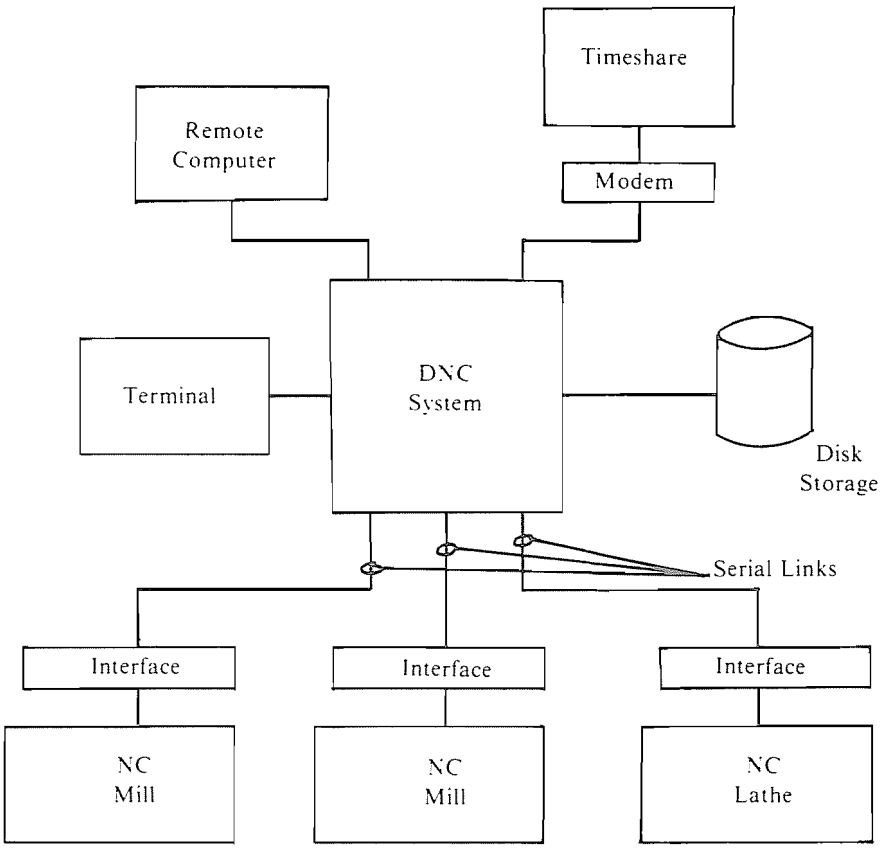


Figure 1. Distributed Numerical Control System

the local disk. A serial link to a development computer permits transmission of FORTH source code to the DNC disk when application software revisions are made.

All DNC functions are supported by a multitasking, multiterminal FORTH operating system that runs on a 16 bit single board computer.

The DNC data base management system indexes the named part programs according to the machines that can run them. This permits use of duplicate names for part programs that run on different machines. For example, the name "gauge-housing" could be used for the two different programs that run on the vertical mill and the horizontal mill. The indexing also prevents inadvertent downloading of a milling machine program to, for instance, the lathe.

Workstation Controller

In the NBS AMRF, part machining, part fixturing, and materials handling operations are performed by automated workstations. The coordination and monitoring of these operations are the functions of the workstation controller. Hardware components of the machining workstation, shown in Figure 2, include an NC machining center or NC lathe, industrial robot, automated fixturing system, materials transfer system, and safety and sensor systems.

During batch production, the workstation controller receives commands from a higher level of factory control, and issues commands to the equipment controllers to direct the sequence of tasks that comprise the "process plan" for a batch of parts. Status feedback from the equipment controllers is monitored so that tasks are initiated in their proper sequence. For example, the robot is commanded to put a part in the machining center's vise by a workstation controller command "TRANSFER Blank

FROM Tray #77 TO Fixture." While robot motion is in progress the robot controller reports a status of "TRANSFER command in progress." When the part has been placed in the vise, the feedback is "TRANSFER COMMAND COMPLETED." The workstation controller then issues a command to the fixturing controller to "ACTUATE, Hold-part". Equipment controllers do not communicate with each other; the task coordination function is centralized at the workstation controller.

The hardware and software systems comprising the workstation controller computer are identical to those of the DNC. A single board computer with common bus peripheral boards runs multitasking FORTH.

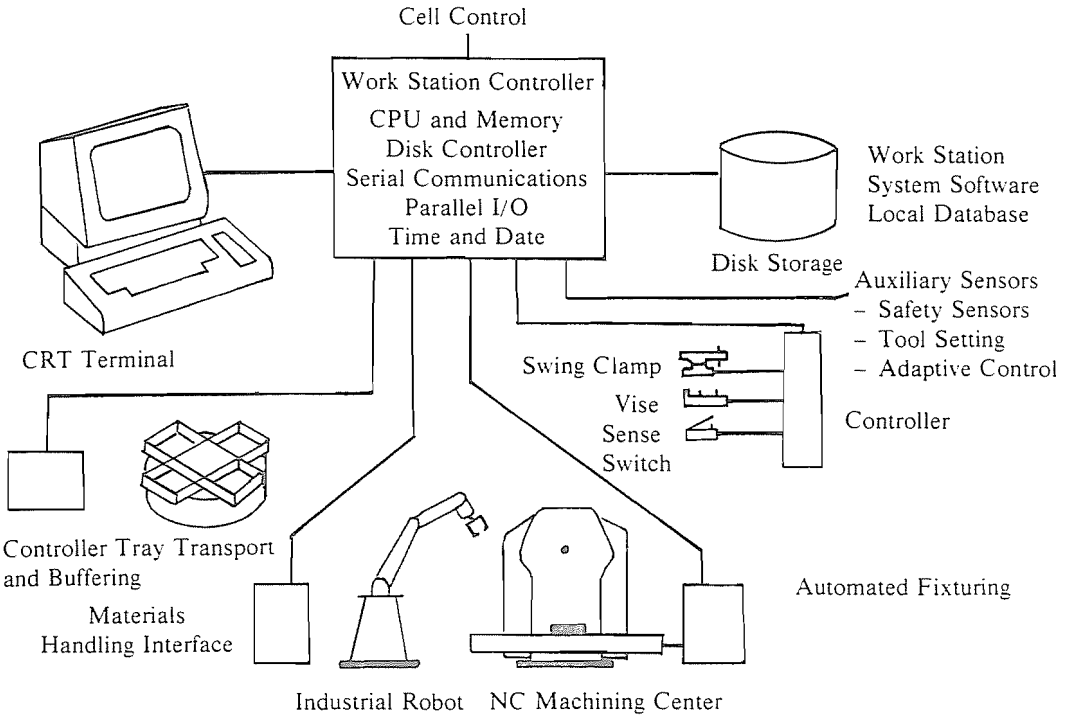


Figure 2. Work Station Control Architecture

Command and status feedback links to the equipment controllers can be serial or parallel signals, depending on the interface supplied by the equipment controller.

The software processes comprising the workstation controller are: 2 levels of real time decision making control processes, remote data base interface, external parallel I O, diagnostics, process plan loader, external command information processor. Processes are coded as separate modules. Each module obtains all of its input parameters from a memory buffer, and deposits all output in another buffer. Interprocess communication occurs solely through these buffers, which are called "common-memory."

All processes in the workstation controller run asynchronously with the round robin task scheduler providing access to the CPU. System resources, such as the disk, parallel I O drivers, and the common memory buffers, are protected by the GET RELEASE words. Processes requiring serial I O are coded as terminal tasks. For development purposes the two real time decision processes are also terminal tasks, even though they use no serial I O directly. The terminal tasks are useful for displaying run time diagnostic messages and for trouble-shooting in a non real-time fashion.

Robot Control System

The group at NBS that is conducting robot control system research is testing generic control concepts involving interprocess interfaces, hierarchical decomposition of tasks to simplify programming, real-time control using sensory interaction, and user interfaces for diagnostics and control system

programming. Their implementation of a robotic control system has been developed into a multiple processor architecture in order to meet the processing demands of controlling a 6-axis robot arm using information from sensors, such as a vision system.

Figure 3 shows a 3 level hierarchical decomposition for the task "assemble pump." Each horizontal line represents a control level within the robot controller. On the diagram, time runs from left to right. The input to level one is the name of a high level task, "assemble." Level one performs its processing and generates programmed commands to level two such as "ACQUIRE," "PLACE IN," "INSERT IN," and "ATTACH." An example of level 2's response to an "ACQUIRE" command is the sequence of outputs "GOTO," "LOCATE," "APPROACH," and "GRASP." Level three responds to input commands by outputing updated trajectory coordinates to the arm. The NBS robot control system uses 6 internal levels to transform high level commands into 20 millisecond updates of joint servo signals.

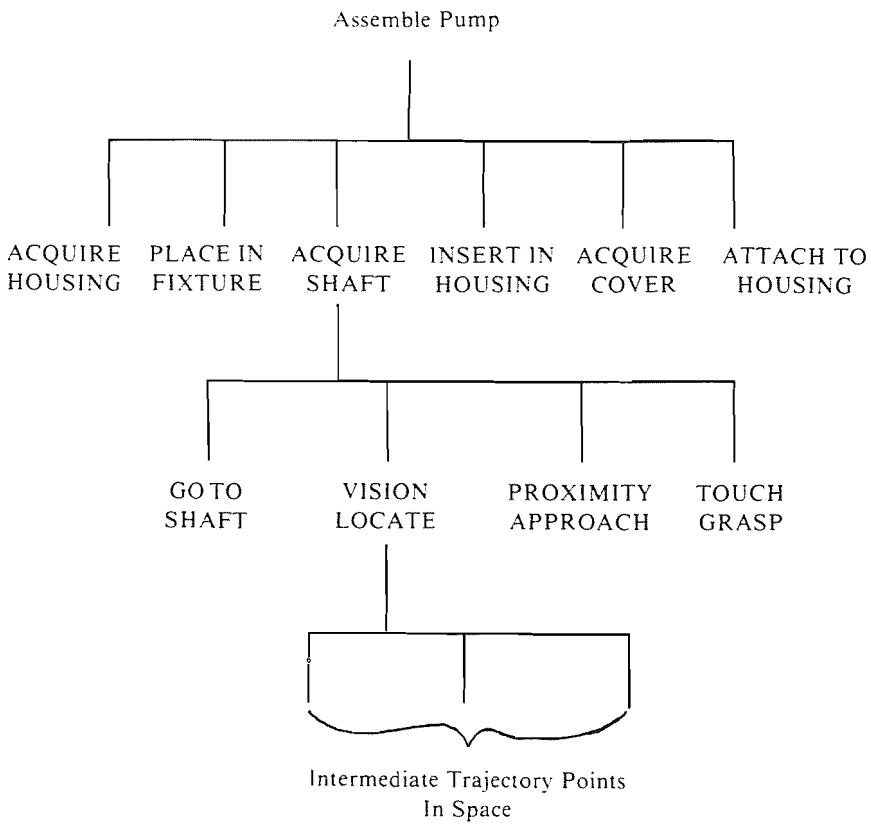


Figure 3

Processes that are separated into modules are sensory information processing, control (one module for each internal level), internal communications, diagnostics, and user interface functions. The model used for each process is in Figure 4. A process obtains data from its input buffer, performs its programmed algorithm, and puts all outputs into another buffer. These three steps comprise a control cycle.

In the robot control system, a communications task moves all output data from buffers to the proper input buffers. Moves of contiguous bytes of memory are desirable to simplify the job of the communications task and to minimize the amount of time needed to transfer large amounts of information. A FORTH implementation of allocating contiguous memory locations for VARIABLE memory is shown in Figure 5. The defining word "OWNER" allocates memory in general buffer space,

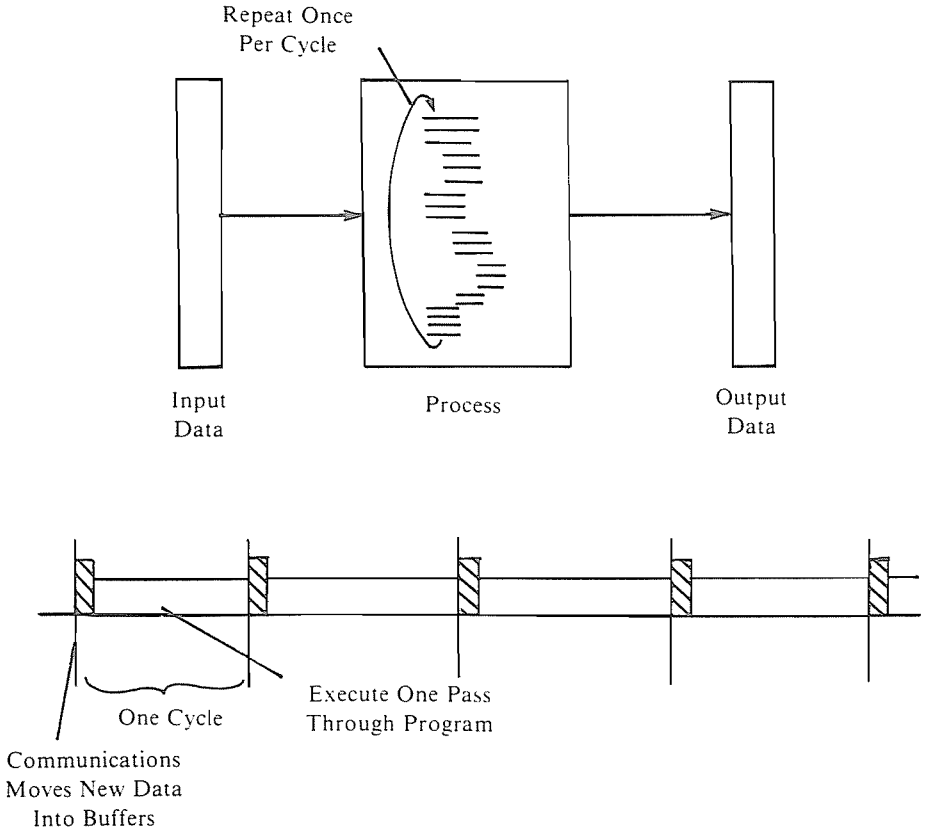


Figure 4

and builds a dictionary entry that contains the number of variables that will use the buffer, and a pointer to the beginning of the buffer. The defining word AVARIABLE allocates no memory, but it leaves a place in the dictionary entry for a pointer that will later be set to a location in a named buffer. When AVARIABLE executes, the pointer is initialized to zero. The word ASSIGN puts the address of the next available word in the named buffer in the pointer entry of the assigned variable's dictionary entry. It also updates a redundant list of the pfas of variables assigned to the buffer. If a variable name is invoked, and it has not been assigned to a buffer, an error message is displayed.

The robot controller implementation has an architecture of several single board microcomputers running on a common bus backplane. Each board runs a separate FORTH operating system.

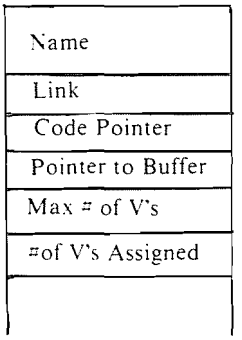
General Observations on FORTH

Two features of FORTH can contribute to efficient writing and debugging of software modules that "talk" to hardware. The FORTH assembly language is easy to use and extend if necessary and assembly language words can be freely interspersed with FORTH words. The capability for an operator to test individual words from the terminal can eliminate having to write special test programs. Initializing conditions and testing the software from the keyboard gives indications about its behavior very quickly.

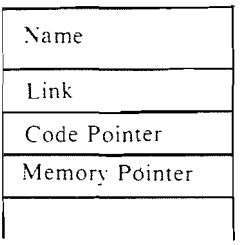
The robotics research group found the three unique character naming convention to be a limitation, but the extensibility of FORTH allowed them to change the dictionary entry and search words, and re-target compile their system to permit 31 unique character length names.

People who have been "raised" on operating system environments that run on multi-user mini-computers tend to complain about the lack of file structures and file editing capabilities in FORTH. Some general purpose packages have been written in-house for text editing and disk block allocation.

```
10 OWNER MOTION-OUTPUTS
     ( 40 BUFF-H ALLOT)
```




```
VARIABLE VELOCITY
VARIABLE ACCELERATION
```




```
TO  MOTION-OUTPUTS ASSIGN VELOCITY

TO  MOTION-OUTPUTS ASSIGN
      ACCELERATION

MOTION-OUTPUTS  MEMBERS
     VELOCITY
     ACCELERATION
```

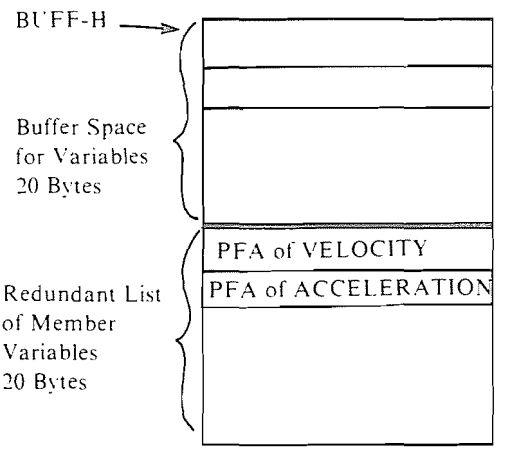


Figure 5. Assign 'VARIABLE' Space to Contiguous Memory in a Named Buffer

Until recently there was no good quality literature to use to bring new FORTH programmers up to speed. Now that Brodie and a few others have published good tutorials for beginning FORTH, a book on advanced FORTH techniques might be helpful to many FORTH users.

The 64k address space has been a limitation to the robotics control group in their multiprocessing applications. Individual processes had adequate space, but the "common memory" areas could not be addressed using conventional FORTH. The group is currently working on solutions to this problem.

FORTH at NBS

The automation research involves about 80 people. About 15 in 2 work groups have used FORTH to implement 5 projects, all on microcomputer single board computers. Other computing hardware in use includes general purpose minicomputers and desktop computers. Other languages in use are Assembler, Fortran, Pascal, C, Basic, and Lisp.

Research groups have relative autonomy in their selection of methods and tools. There is no institutional policy concerning use of software systems at NBS. Both of the work groups at NBS that use FORTH have applied it to real-time control of integrated hardware systems.

FORTH would gain wider appeal among NBS computer users if it could be used for applications other than real-time hardware control. Possible areas in the automation research are graphics, AMRF simulation, automated generation of control data, and data base management.

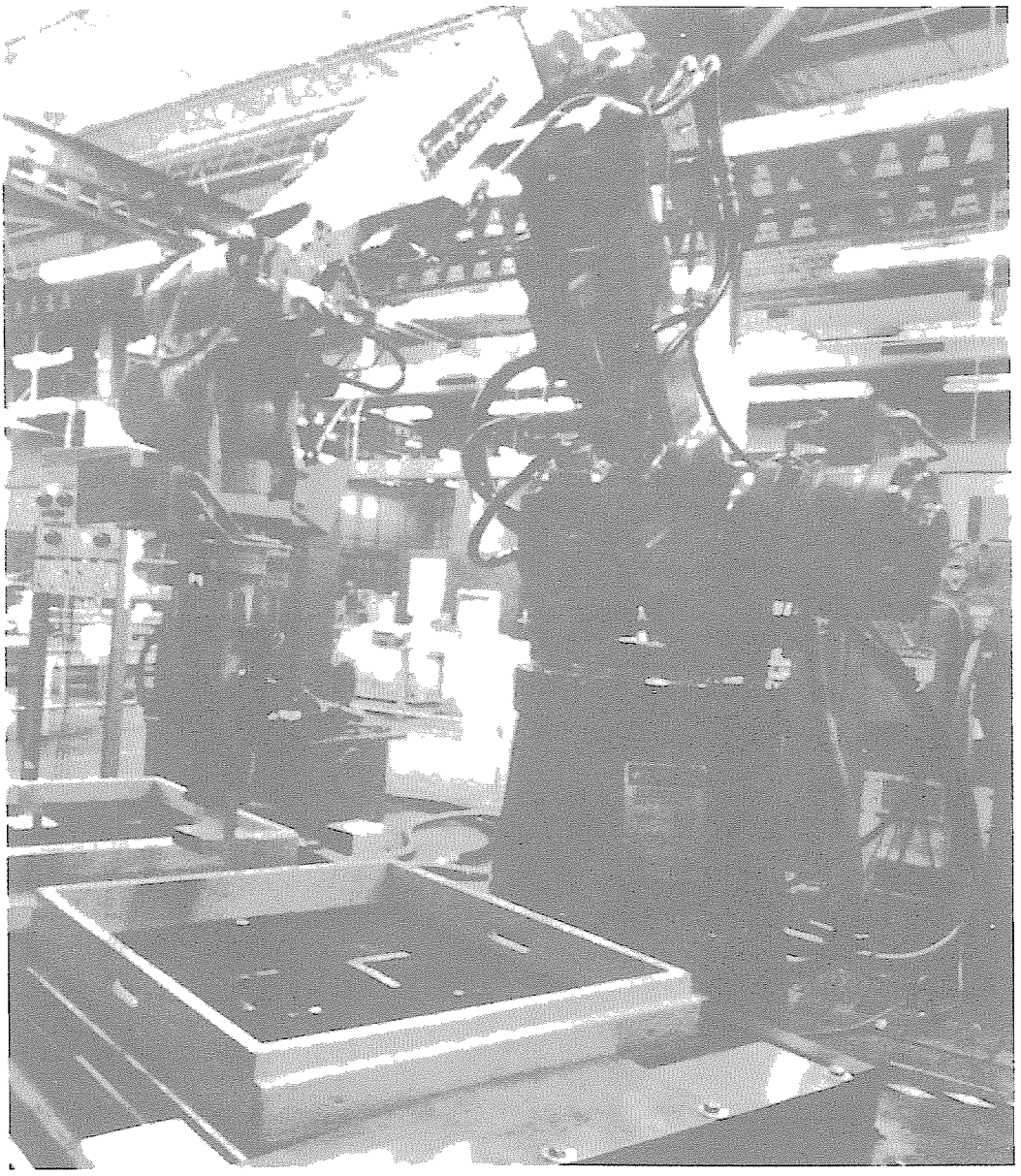


Photo 1. Robot and Cart.

References

1. A. J. Barbera, M. L. Fitzgerald, J. S. Albus, "Concepts for a Real-Time Sensory-Interactive Control System Architecture", presented at the *Proc. 12th International Symposium on Industrial Robots*, June 9-11, 1982. Available from the National Bureau of Standards.
2. H. A. Scott, W. G. Gregory,m "Real-Time Control of a Machining Workstation," presented at the *Numerical Control Society 20th Annual Meeting and Technical Conference*, Cincinnati, Ohio, April 10, 1983. Available from the National Bureau of Standards.
3. R. J. Hocken, "NBS Building Facility for Research in Automating Small Machine Shops", *Industrial Engineering*, April 1982.

Mr. Rippey graduated from Penn State with a BS in engineering science in 1973. He originally worked on power plant instrumentation and has been in automation research at NBS for the past 4 years. The author is currently developing control structures for automated factories at the workstation level.