Presented at the Summer Computer Simulation Conference, in Vancouver, BC, Canada, July 11-13, 1983.

THE AUTOMATED MANUFACTURING RESEARCH FACILITY OF THE NATIONAL BUREAU OF STANDARDS

Cita M. Furlani, Ernest W. Kent, Howard M. Bloom, and Charles R. McLean
Industrial Systems Division, Center for Manufacturing Engineering, NEL
National Bureau of Standards
Washington, D.C. 20234

ABSTRACT

A major facility for manufacturing research is being established at the National Bureau of Standards (NBS). NBS is addressing the measurement and standards needs for the automation of the small batch, discrete parts manufacturing industries. The Automated Manufacturing Research Facility (AMRF) is being developed to serve as a testbed and demonstration facility in support of research by workers from NBS, industry, academia, and other government agencies. The facility control architecture has been adopted from the NBS robot control system and is a modular hierarchical feedback system. The unique features of the control system of the AMRF are the number of hierarchical levels and the amount of real-time computation and sensory-interaction at each level. Major research areas that are discussed are (1) the development of the manufacturing planning and control software, and (2) the system hardware integration projects. In support of this research, simulation and emulation are being used for design, development and testing. Emulation techniques are being employed to study the operation of the control software of the AMRF. The Hierarchical Control System Emulation (HCSE), presently in use at NBS, is described.

INTRODUCTION

The National Bureau of Standards (NBS) is the United States' measurement laboratory in the physical and engineering sciences. An agency of the U.S. Department of Commerce, NBS was established by Congress in 1901 to help insure the compatibility of measurement standards needed by industry, consumers, the scientific community, and other government organizations. These standards provide the basis for the exchange of goods, the accurate specification of products, quality control methods for production, the equitable enforcement of environmental regulations, and the establishment of adequate guidelines for the protection of public health and safety. NBS is not a regulatory agency, but rather a laboratory used by industry, academia, and government alike as an independent, authoritative source of technical information and advice.

To help reduce or remove technical barriers that impede the prompt introduction or exploitation of new technologies, NBS is working to improve measurement methods, data, and standardization in such areas as semiconductor electronics, materials science, automated manufacturing, and chemical engineering. To accomplish this, the Bureau has, over the years, installed numerous experimental facilities, including a nuclear research reactor, a linear accelerator, and dead weight force generators with a capacity of 4.4 meganewtons. NBS has recently embarked on the design, procurement, and installation of a new Automated Manufacturing Research Facility (AMRF) to support its measurement and standards responsibilities in the decades of the 1980s and 1990s.

Thus, the role of NBS in automated manufacturing is twofold: 1) to provide the basis for measurement assurance, a means by which the dimensional attributes of manufactured products can be traced to national standards, and 2) to assist in the development of those voluntary standards necessary for the successful automation of industry. Basic and exploratory research is carried out, as needed, to support these major functions.

In particular, NBS is addressing the measurement and standards needs for the automation of the small batch, discrete parts manufacturing industries such as those supplying parts for

---

*The Emulation Project is Sponsored by the Air Force/DARPA ITA Project.

*The NBS Automated Manufacturing Research Facility is partially supported by funding from the Navy Manufacturing Technology Program.

aircraft, automobiles, and industrial machinery. These industries produce goods that account for 75 percent of all U.S. trade in manufactured goods. Although this sector of the U.S. economy includes 35 percent of all U.S. manufacturing firms, over 100,000 of these firms (or 87 percent of the total) have less than 50 employees. It now appears that these industries must automate to successfully compete, in both foreign and domestic markets, with Japan, Germany, and other industrial countries which have major government sponsored programs in automation.

The aspect of manufacturing that has been chosen for research is the completely automated small batch manufacturing system, or flexible manufacturing system. It consists of such elements as metal cutting machines, computers, terminals, robots, conveyors, and measuring or inspection systems. Such a system produces a variety of shapes and sizes of parts to be assembled into larger systems; for example, gears, shafts, and cases into transmissions.

The AMRF is the work of a number of groups within three Divisions of the Center for Manufacturing Engineering. The work of all these groups will be briefly described along with a more detailed description of the Hierarchical Control System Emulation (HCSE).

DESCRIPTION OF THE AMRF

The AMRF is being developed to serve as a testbed and demonstration facility in support of research by workers from NBS, industry, academia, and other government agencies. It has become a focus of effort and center for research in automation of small batch machined parts manufacturing. The AMRF is a major facility which supports continuing studies of interface standards for computer integrated manufacturing systems. The facility also functions as an important vehicle for communicating current technological information to other agencies and their contractors.

The AMRF is being designed to handle the bulk of the part mix now manufactured in the NBS Instrument Shop. This part mix has been studied using group technology (GT) concepts and is similar to that found in a typical machined-parts job shop. The AMRF and the research performed on it will address only the manufacture of individual parts by chip forming metal removal. Hence, the unit operations will include only fixturing, milling, drilling, reaming, tapping, boring, turning, facing, threading, cleaning, deburring, and inspection. Currently there are no plans to address such problems as automated assembly, welding, hardening, and finishing. Within this constraint, the intent is to completely automate production from the transfer of near net shape blanks from inventory to the delivery of finished, cleaned, and inspected parts.

The shop floor layout of the AMRF is based on the concept of fully automated workstations, each with a well-defined set of functions and capable of running in a stand-alone mode. A typical machining workstation will include an industrial robot, an NC machine tool, a local storage buffer for tools and materials, a material handling system interface, and a workstation control computer which integrates and coordinates the operations of the components listed.

The current development plan for the NBS research facility calls for seven stations of varying degrees of complexity. These stations and their component systems are representative of general purpose production equipment in common use throughout the United States. The equipment selected also meets the machining needs of the NBS Instrument Shops as indicated by the group technology study. The seven stations are:

(a) Horizontal Machining Station

(b) Vertical Machining Station

(c) Turning Station

(d) Inspection Station

(e) Materials Inventory Station

(f) Transfer System

(g) Housekeeping System

Items (f) and (g), the Transfer and Housekeeping Systems, are not strictly stations since they are non-localized in the facility. From the point of view of the control system, however, they will be treated as stations.

The goal in designing the facility control architecture is to implement a system that provides sensory interaction along with the flexibility of software automation, i.e., the functioning of the system can be changed over a very wide range by varying data rather than by reconfiguring mechanical elements or reprogramming control structures. Thus, the AMRF control system has been adopted from the NBS robot control system and is a modular hierarchical feedback system. The control architecture is shown in

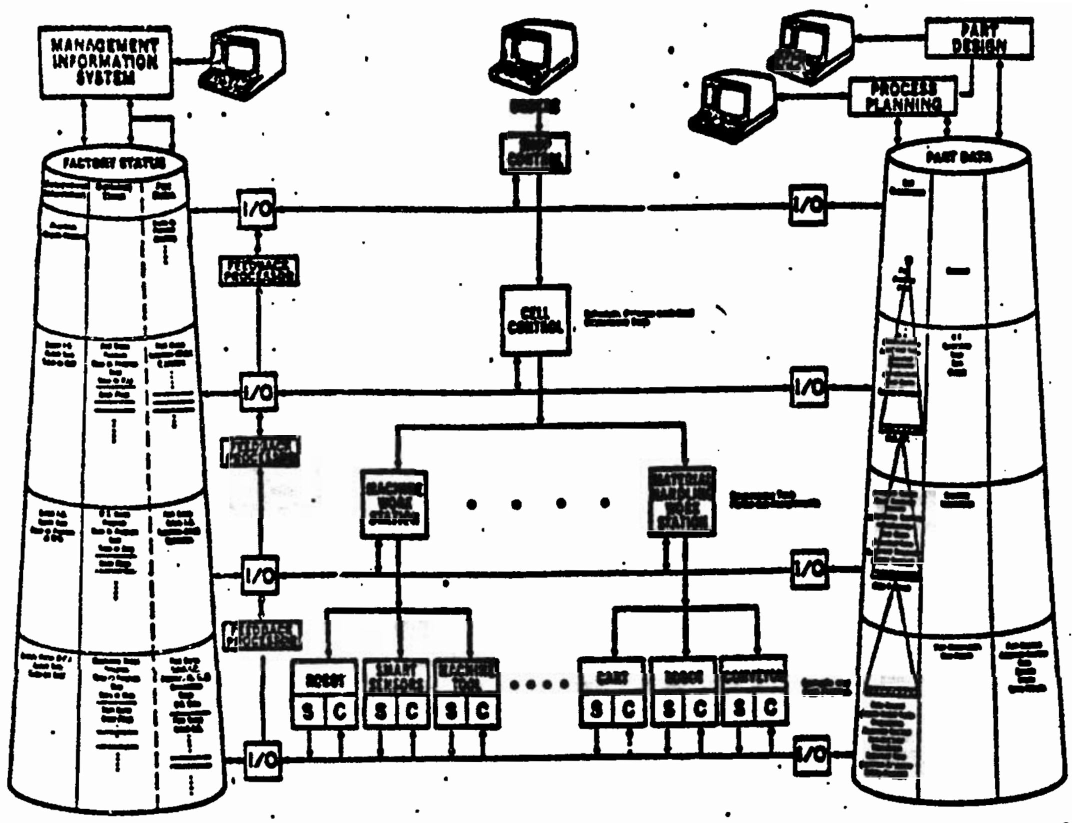


Figure 1. A Hierarchical Architecture for a Factory Control System

(Figure 1). The two distributed hierarchical data bases contain all facility planning and control information. The planning data base contains all data necessary to manufacture the selected part mix, including part dimensions and geometry, desired grip points for robot handling, and tool and material requirements. In the same data base are found the process plans for routing and scheduling, as well as cutter location files needed for performing the various machinery operations. The control data base contains dynamic factory status information, including management information data. Also included is the status of all the control systems, tools, robots, and computers in the hierarchy, as well as each work order in process.

Information is passed from one control level to another and from one computing module to another through the data bases which serve as common memory. The distributed data management system of the AMRF is based on a hierarchy of task decomposition modules and feedback processors that scan the appropriate levels of the data base and extract the information needed by the various control modules. Higher level control modules decompose complex tasks into simpler ones. The simple tasks are issued as commands to controllers at the next lower level. Feedback processors scan the appropriate levels of the data base, and extract and process sensory and status information needed by control modules at higher levels. This processed sensory feedback data is also passed between control systems through the common memory mechanism.

At the top of the hierarchy, orders are placed for machining of parts. At the lowest level of the hierarchy are individual device controllers that drive various machines on the shop floor. The unique features of the control system being implemented in the AMRF are the number of hierarchical levels (seven or eight, each typically with several sublevels), and the amount of real-time computation and sensory-interaction at each level. The control and data management architecture clearly distinguishes the AMRF from "just another FMS" and provides the basis for much of the NBS research activities.

Several research projects are underway to implement the AMRF. Two major research areas discussed in this paper are (1) the development of the manufacturing planning and control software, including support software tools and (2) the system hardware integration projects, including the enhancement of commercially available hardware, such as robots and machine tools, with special sensors and controllers to provide compatibility with NBS control and interface concepts.

MANUFACTURING PLANNING AND CONTROL SOFTWARE

The purpose of this research is to develop an understanding of the modules

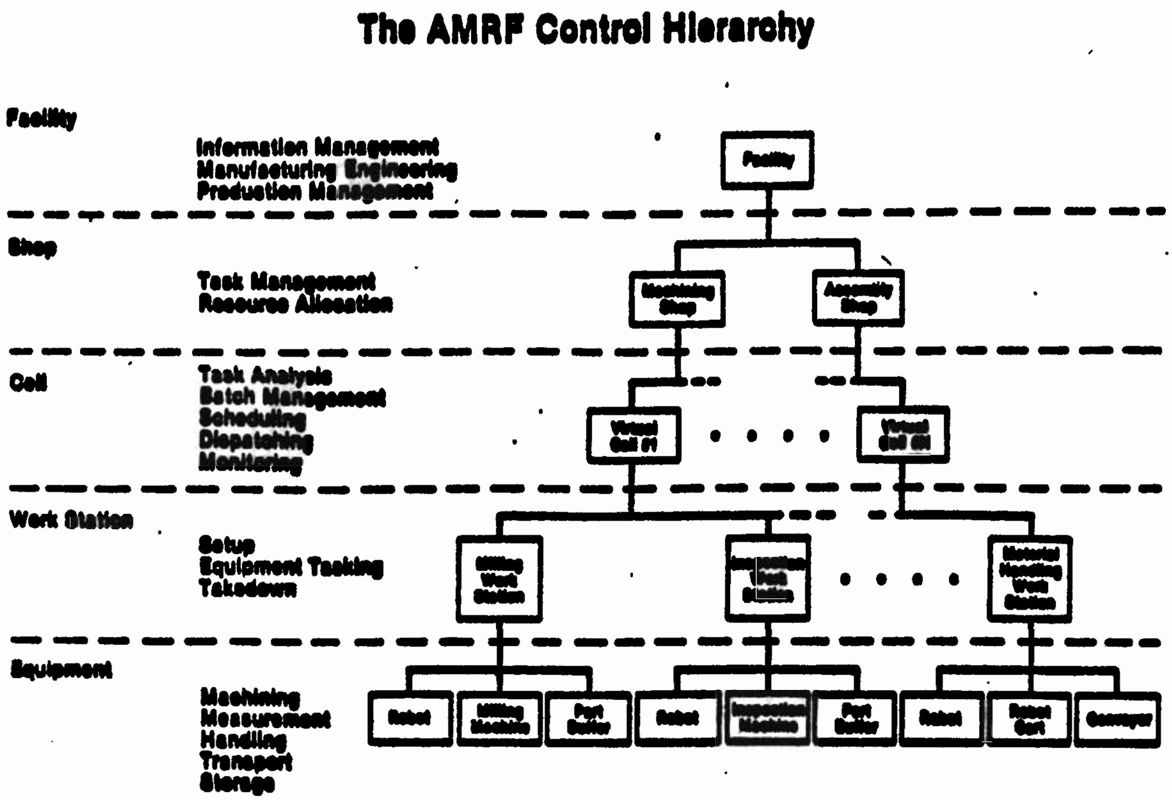


Figure 2. The AMRF Control Hierarchy

that will comprise future computer integrated manufacturing systems. To accomplish this, the NBS approach has been to use structured software development techniques to analyze the requirements of systems, decompose module functions and develop the software architecture of the manufacturing hierarchical control system (Figure 2). Particular attention is being given to the data that must be communicated between modules, and the representation of the information within the system and to the user. These data structures will provide a foundation for the development of potential standard interfaces.

The manufacturing planning and control software research activities are described in terms of four major efforts: Production Control, Distributed Automated Process Planning, Database Systems, and Network Communications.

Production Control

The objectives of the research in production control are to develop (1) a real-time, adaptive, state-table driven, hierarchical production control system for the AMRF that provides the flexibility and adaptive behavior of existing manual control systems, and (2) specifications of potential standard interfaces for modular, multi-vendor integrated manufacturing control systems.

A general-purpose control system is being developed that is transportable and implementable at any level in the AMRF. Major modules of this system include special software for handling the state table structure including an editor, a compiler, a data management system, an execution processor and a testing/debugging system.

The control system processes the state tables in real-time to allow adaptive control at each level. This technique will ensure the speed and reliability that is required of a real-time control system. The state table processing is being integrated into the production control system, a hierarchy of five major levels. The levels are (in descending order): Facility, Shop, Cell, Workstation, and Equipment. Functions performed at the Facility level include manufacturing engineering (design, process planning, production planning), information management (order handling, cost accounting, inventory management, procurement, and performance monitoring), and long-range scheduling. Shop level functions include shorter-range production scheduling, resource allocation, production task management, and support activities management. Cell level functions include production job analysis, reporting, routing, scheduling, dispatching, and workstation monitoring. Support activities include a variety of functions such as material storage, transportation, tool assembly, cleaning and deburring. Workstation level control includes the supervision and coordination of base machining, handling, and measurement operations.

At the cell control level, the concept of the group technology manufacturing cell will be extended by the development of software-based production management systems (virtual cells) whose configuration varies dynamically according to the processing resource rquirements of a batch during the production process. These virtual cells will be responsible for managing the production of parts that have been batched using a group technology classification scheme. The cell will exist solely as a control process for the duration of the production run of the batch. This approach represents a considerable change in the philosophy of cell control as in the past cells have had static control structures that were normally associated with a fixed physical grouping of machinery. The cell controller will maintain a PERT-type network to track progress and predict when it will need to request the assignment of specific workstations from the shop controller in order to process its batch of parts. Workstations will be allocated to cells by the shop level resource allocator only when they are required. Idle workstations will be assigned to a workstation pool cell controller. This structure promises to provide much of the flexibility found in manual job shops while increasing efficiency through automation.

Distributed Automated Process Planning

The objective of this recently initiated subproject is to (1) identify and provide the process planning functions that must be performed in a totally automated facility, and (2) investigate artificial intelligence and expert system techniques for distributing planning functions within the AMRF. Local intelligence will be developed for each control module. Standard techniques will be developed to allow modules at different levels in the hierarchy to access the planning capabilities and knowledge bases of other modules as necessary, to perform distributed planning.

In an automated facility process plans must be comprehensive in that they must specify all the details of how to produce a part; including selection of materials, machine tools, robot grippers, grip points, fixtures, clamping points, inspection procedures, cutter paths and parameters, coolants, and any special material transportation considerations.

The distributed process planning system that is planned for the AMRF will consist of a hierarchical network of planning systems that reflect the control tasks at each level. Systems will have their own particular planning specialties. Consulting relationships between systems, similar to those found between different levels of managers and operators in conventional systems, will be used to obtain answers which fall outside of any system's domain of expertise. This distributed approach should allow for real-time modification of process plans at the appropriate control level and ensure efficient usage of system capabilities as the systems dynamically change with time.

Database Systems

The objective of the AMRF data base work is to (1) define and develop the data structures necessary to support an automated manufacturing system, and (2) utilize data base management techniques to develop flexible data-driven control systems which provide the information required by the AMRF operating in a distributed and real-time control processing environment.

The first subproject is the identification of the data structures which will be required to support the AMRF. An analysis of data requirements is being made for each function and level of control within the AMRF. This includes the data requirements of a typical manufacturing operation, those of an automated facility, and those specific to the research needs of the AMRF. The first prototype database system will operate in an interactive but stand-alone fashion utilizing menu-driven interfaces. The next stage will be aimed at fine tuning database structures, developing the interface specifications between systems and levels of control, generating a workable distribution of the data, and identifying data which for performance reasons must be replicated in low level processors. Data dictionary functions will be heavily employed during the design and implementation phases. An automated data dictionary can be used to control standards, facilitate shared access to data, and allow for distributed systems.

The second database effort is concerned with the development of the software architecture to maintain this data and permit access to it. The database software must support a distributed processing environment. At lower levels of control, namely at workstations and equipment, processes will run on special purpose microcomputer-based systems, and rapid response times are imperative. Commercial data base systems, in particular those that can be extended by higher level programming languages, will be used at higher levels, but may not be suitable at lower levels.

A common data definition language and a common data manipulation language will be defined so that the distributed nature of the database can be made transparent to the control systems that access it. The entire set of databases will be organized into a common logical view in order to provide a data model that describes how each data element is used within the AMRF. Through the implementation of logical views each data user, whether it be a human or control process, will access and will be presented with only the information that is relevant to its/his particular functional task. This logical access to data permits the user to be ignorant of the actual physical organization of data, and permits it to be shared by multiple users.

Network Communication

The objective of this network communication project is to provide a communication link that allows all system developers to interface their terminals to any AMRF computer or control process, and to provide network and local communications mechanisms for control processes within the AMRF.

The network is being implemented using commercially available hardware. Standard network interfaces will be obtained to handle the variety of computers to be linked via the network. The heart of the research effort is the development of the software interface standards that will provide AMRF processes with a means for communicating with (1) each other, (2) the operating systems of control computers, (3) the local communications system interface, the distributed data management system, and (4) the user display devices.

A common communications language will provide a standard method for naming, opening, reading, writing, and closing logical communication channels, i.e. mailboxes. A process management command language would provide common terminology and syntax for naming, creating, moving, suspending, interconnecting, setting priorities, and performing other computer process management functions.

Communications nodes will be associated with one or more processors which share a common physical interface to the network. The computing systems of the node will support functions such as: manufacturing control, data administration, program and communication control. Processes within a node will access each other by reference to logical names, their associated standard interchange mailboxes, and the common local data path.

SYSTEM HARDWARE INTEGRATION

At the lowest level of the manufacturing control hierarchy is the elemental control unit represented by robots, machine tools, carts, etc. Projects at this level address the problem of making the unit more responsive to a changing environment by incorporating sensory feedback adaptive control for improved accuracy and performance.

Research is being performed in the following areas: (a) work station control systems, (b) machine tool metrology, (c) machine tool sensors, (d) real-time robotic control systems, (e) robot vision, and (f) non-visual robot sensors.

Workstation Control System

The workstation control system project involves the integration of automated manufacturing equipment into production work stations for small lot manufacturing. The work involves extending and enhancing numerically controlled machine tools and industrial robot manipulators to form flexible, coordinated, and sensory-interactive systems. Research is focused on the generation and verification of control data, the development of real-time software for synchronization and control, and the development of interface standards and technical guidelines.

The first workstation consists of a horizontal machining center that has been integrated under work station control with a robot, a simple fixturing system, and two loading platforms served by a robot cart outside the workstation.

The coordination and supervision of this equipment is the responsibility of the workstation controller. Essential features of the workstation controller are: unattended operation of the workstation; extensive use of sensors; ability of the program to adapt to changes in machine availability, part designs and batch sizes; and status reporting to higher control levels.

The workstation receives raw materials, modular fixturing components, and robot and NC tooling from inventory, machines batches of parts, and releases all materials to inventory. Coordination of these operations within the workstation is centralized in the workstation controller. As a level in the hierarchical control system of the AMRF, the work station controller receives production commands from the cell controller, decomposes these commands, and issues commands to manufacturing equipment controllers to coordinate their operations. It performs these functions and reports work station status back to the cell controller once each control cycle.

Machine Tool Metrology

Machine tool metrology research is concerned with the improvement of measurement and machining accuracy in existing hardware. An automated system provides greater reproducibility of behavior than a manual system. In fact, if an automated system makes a good part, and if none of the process parameters changes, then it will continue to make good parts. This concept underlies the philosophy of deterministic metrology, which concentrates on making a good part by measuring and adjusting process parameters rather than on measuring a completed part. Specific efforts include (a) improvement of the point-to-point positioning accuracy of a computer numerical control (CNC) machining center through the use of on-line computer modeling and process monitoring, and (b) development of quantitive techniques for evaluating and improving the performance of coordinate measuring machines.

The machining accuracy enhancement research is being performed with a vertical spindle CNC machining center, using a dual minicomputer system and laser interferometer system. The enhanced machine tool will be benchmarked by producing a pair of high precision, high speed, rotary postage stamp perforating cylinders for the Bureau of Engraving and Printing. The successful manufacturing of these cylinders requires a positioning accuracy of ±2 micrometer over a one meter length for the peck drilling of approximately 50,000 holes (1 mm diameter) in the pair of cylinders.

Several approaches to machining accuracy enhancement are utilized. These include (1) static positioning errors: corrections from calibration data are supplied from the computer into the servo loop of the machining center (2) thermal errors: data from an array of thermal sensors are used for quasi-real time position correction. (3) dynamic errors: tool wear, tool chatter, spindle run-out and machine vibration are sensed and corrected, and (4) table deformation and loading: theoretical prediction and experimental verification of the natural frequencies and mode shapes of machine tools are applied.

In coordinate measuring machine (CMM) research, a complete set of kinematic errors for a computer controlled CMM have been measured and incorporated into error modeling software. This model, combined with differential thermal expansion calculations, has been able to predict linear positioning errors within four micrometers for an arbitrary diagonal of the machine. The testing of 3-D ball plates continues as part of our effort to support the development of standard acceptance tests for CMMs.

Machine Tool Sensor Systems

With the advent of fully automated functions, on-line tool sensing is more important than ever before since human operators are not normally present to monitor the process.

One research effort has resulted in the development of Drill-Up, an instrument which was originally designed to avoid breakage of small-diameter drills used on automatic-feed drilling machines with a spindle-retract capability. The instrument determines that breakage is imminent and commands the drilling machine to retract the drill. The input sensor is a piezoelectric accelerometer which is mechanically coupled to the workpiece. Potential drill breakage is determined by time-domain analysis of the accelerometer signal. A calibration routine automatically adjusts to the normal amplitude of the signal. Large amplitude accelerations, synchronous with the drill rotation, have been found to be indicative of improper cutting. Drill-Up's detection method recognizes that scraping in the hole is synchronous with the rotation of the spindle.

This detection method is based on the characteristic deflection of a column with one rigid support on the end opposite an axial load. When the drilling is improper, due to factors such as a worn cutting edge or a hard spot on the workpiece, the material cannot be removed as fast as the drill is being fed into it. When this occurs, the drill deflects as a column and induces a vibration signal as it scrapes on the side or bottom of the hole. If this is allowed to continue, the column will collapse, resulting in drill failure.

A second sensor project is concerned with spindle error testing and analysis. Besides the quasi-static errors of imperfect geometry, load deformations and thermal deformations, a machine tool has errors associated with its dynamic behavior. The most important of these errors are the spindle error motions and the kinematic errors associated with the coordination of axis motion and vibrations, both self-induced and forced. A microprocessor-based system has been developed for spindle error testing and design that uses a position encoder, a high-speed sample-and-hold circuit, and an analog-to-digital converter. In addition, a displacement transducer with wide-bandwidth and extended dynamic range has been developed such that 512 points per revolution may be analyzed at rotational speeds up to 10,000 rpm. Because of the low cost of microprocessors, this system can be dedicated to a machining center for continuous monitoring.

__Robotic Real Time Control System__

The robotic real-time control system addresses three major areas - definition of a robot system architecture to accomplish real-time control, specification of functional module interfaces, and the study of user interfaces for programming and diagnostics. The system architecture partitions the system into functional modules. Interfaces between these modules can be described as well-defined input and output data buffers. These data buffers, which reside in a common memory that is shared by all modules, become the mechanisms for interfacing between modules and therefore for interfacing between the control system and the sensors, robots, supervisory control, etc. __When a user is to__ interact with a system of this complexity, additional functional modules are required. These user interfaces are made through the same type of data buffers and thus allow the use of various programming and display modules.

A sensory-interactive robot control system has been developed at NBS. It has been implemented on two robots: a Stanford Arm and a PUMA 560. It illustrates the following points:

(a) generic task decomposition control modules operating as independent state machines in a hierarchical multi-level system.

(b) a common-memory communications architecture, for a modularly structured system, implemented on single or multiple processors.

(c) common-memory resident data interfaces between the modules of a structured system, to be used as the mechanism for interfacing other sensors, supervisory control, data bases, and different robot hardware.

(d) table-structured mechanisms (such as state-tables) for the representation of data and program control in a convenient form for user interaction and modification.

(e) the use of diagnostic tools to access common memory to provide a real-time trace of user defined variables.

__Robot Vision System__

The robot vision work is being performed in three major research areas: 1) the integration of control theory with image processing systems, 2) real-time gray scale vision processing, and 3) fast acquistion of range data.

A robot vision system for industrial robotics is under development. Television frames and inputs from other sensors are interpreted by a hierarchically organized group of microprocessors. The system uses knowledge of object prototypes, and of robot action, to generate visual expectancies for each frame. At each level of the hierarchy, interpretative processes are guided by expectancy-generating modeling processes. The modeling processes are driven by a __priori__ knowledge, by knowledge of the robot's movements, and by feedback from the interpretative processes. At the lowest level, other senses (proximity, tactile, force) are handled separately; above this level, they are integrated with vision into a multi-modal world model. At successively higher levels, the interpretative and modeling processes describe the world with successively higher order constructs, and over longer time periods. All levels of the hierarchy provide output, in parallel, to guide corresponding levels of a hierarchical robot control system.

Robot vision systems have requirements which are frequently not encountered by computer vision systems developed in other contexts. Among these, are: 1) the need to model the spatial and temporal structure of the environment independently of the requirement to name objects in it, 2) the need to make information from this model available, in real-time, to a control system at many hierarchical levels of organization, and 3) the need for multi-modal analyses integrating many sensory processes. There are also unique aspects of robot sensory environments which may be used to advantage. For example, the sensory system may obtain predictive information about the next viewing position from the robot control system. It is also characteristic of robot applications that most of the system's processing time will be spent on problems of sensory servoing, rather than on object identification. This is because almost all of the images with which it deals are seen in a historical context as members of a set successively altered by object and observer motion. In this respect, the problem domain is very similar to that of animal vision. After objects are initially acquired by the sensory system, its principal job is to provide continuous sensory guidance information to the control system as robot and object orientations change.

As a result of this continuity in the world being sensed, the sensory system can employ many kinds of context-dependent and context-independent knowledge in generating attention processes and expectancies with which to guide processing of incoming data, and thus facilitate real-time operation. In our approach, a best possible model of the external world is maintained by servoing the model's predictions against selected, optimally discriminable, aspects of the data. Information required by the control system is passed to the control system directly from the model, independently of the particular data which is analyzed by the interpretative processes which servo the model to the external data.

The present vision system uses structured projected light sources to quickly obtain minimal information about the six degrees of freedom of an object relative to the robot. The ability of the "6-D" robot vision system to acquire unknown three-dimensional objects, arbitrarily positioned in 3-space has been demonstrated; it can guide the robot in useful interaction with the objects. An object unknown to the robot's software can be placed in a random location with a random 3-D orientation. Using flood flashes, the robot can discover the object. Alternating flood flashes with flashes which project two planes of light, it approaches the object and discovers the 3-D orientation of its largest surface. The robot then positions itself normal to that surface. Using the combined information from both illumination types, it correctly interprets the object's 2-D outline in 3-space and locates its centroid and principal axis. It then attempts to grasp it squarely across its narrowest dimension at the center of gravity and remove it from the field. Because it can directly sense the range to surfaces, it can correctly repeat this process to unload a stack of objects.

In support of the longer term AMRF requirements, a multi-stage gray-scale image processing pipeline is being designed and constructed. It will analyze a continuous stream of consecutive 256 x 256, eight-bit images, through a series of stages, at television frame rate. A variety of arithmetic and Boolean neighborhood operators may be applied to each pixel at each stage, and multiple feedback loops exist between stages. A unique architecture allows processing within the device to be influenced by expected models from above as well as data from below. Ultimately, this device will be used in conjunction with the AMRF robot vision system described above.

Also as part of the vision research, experiments are being conducted using multiple structured light frames in the acquisition of full-frame range images. The NBS system will use eight frames of structured light, and is capable of acquiring a 256 x 256 image of eight-bit ranges from them. Thus, a full-frame eight-bit range image can be acquired in about 0.3 seconds. This device will find application both in robot vision and in inspection tasks.

Robot System Integration

Integrated robot systems are being developed through a number of interrelated projects:

(a) AMRF Robot Integration - Each commercial robot in the AMRF is to be equipped with an NBS Controller, a sensory system, an instumented end-effector (gripper) and a safety system. This research seeks to develop effective techniques for integration of all of the above systems with each different type commercial robot.

(b) Instrumented Gripper - Each work station in the AMRF may require several different types of grippers. The current development effort is a two-fingered pneumatically actuated gripper which can be servoed on finger position and gripping force. This gripper will be used on the horizontal machining center robot.

(c) Safety Systems - A "Watchdog" Safety Computer which will monitor robot joint velocity, acceleration, and position is currently under development. The computer will have independent capability to stop the robot if it exceeds pre-set limits.

(d) Robot Cart - A commercial robot cart is being modified for control with an on-board microprocessor and a command and status R/F communications link.

(e) Robot Performance Measurement - This project develops techniques which can be standardized to achieve automated on-site measurement, analysis, evaluation and control of robot performance.

EMULATION AND SIMULATION IN THE AMRF

The development of a facility such as the AMRF demands new techniques to design and test the facility itself. Software tools, such as simulation and emulation, are being used for design, development and testing. Emulation, a method that has recently proven useful in the design of complex computer systems, is a special form of simulation in which the rules governing the operation of a planned system are duplicated, and the user of the emulated system interacts with it as if he were using the real system. This technique is being employed to emulate the operation of the modular hierarchical control software of the AMRF, while conventional simulation is being used to study the overall control and information flow in the AMRF.

Emulation

The system presently in use at NBS is the Hierarchical Control System Emulation (HCSE), a collection of computer programs written primarily in the high-level PRAXIS language. The HCSE and PRAXIS itself were developed at Bolt, Beranek, and Newman, Inc., to run under the DEC VAX/VMS Operating System. The software provided allows the emulation to follow the structure of the AMRF modular hierarchical feedback control system, with the modules conceptualized as finite state machines (FSM) that interact through a shared time-slice synchronized common memory (Figure 3). This permits the emulation to be used to establish the feasibility of such a system, as well as to assess the computational and communications requirements.

The primary advantage of emulation, as compared to conventional simulation, is that it may serve as a prototype for (parts of) the real system and may be developed before the real system hardware is available. When the emulation speed is equal to actual clock speed, then the emulation can function as a mock-up of the actual shop floor control software, and any subset of the simulated machine functions could be substituted for actual physical devices. In bringing the job shop on-line, new components (including man-machine interfaces) may be tested with existing components and simulations of components which are not yet installed. Thus, an emulation is particularly advantageous when there is a significant delay between ordering and receipt of hardware, where user acceptance is important, and where there are a large number of special situations to test, as in the AMRF.

As the concept of the virtual manufacturing cell is being developed, the emulator is being used as a design aid in the system software development, allowing designers to test out their ideas before implementing the code in its final form. Thus an up-to-date design description of all the control levels in the AMRF is recorded, providing self-documentation.

The HCSE is also being used as a design tool for computer-aided-design (CAD) directed inspection. This project involves integrating the emulator with the geometric modeling and graphics functions of PADL-2 (a constructive solid geometry modeling system developed at the University of Rochester), with a coordinate measuring machine, and with a program development environment including an editor, a data dictionary, and documentation aids. A control hierarchy has been defined and two levels of task decomposition have been programmed with the aid of the emulator. This emulation includes a front end through which commands can be entered, a machine simulation module, and a dynamic graphic display of the emulated inspection machine. Work is proceeding on providing communications with the coordinate measuring machine so that the low-level simulation can be replaced with the actual hardware.

Emulations are also being written for the robot control and vision systems that are being developed at NBS.

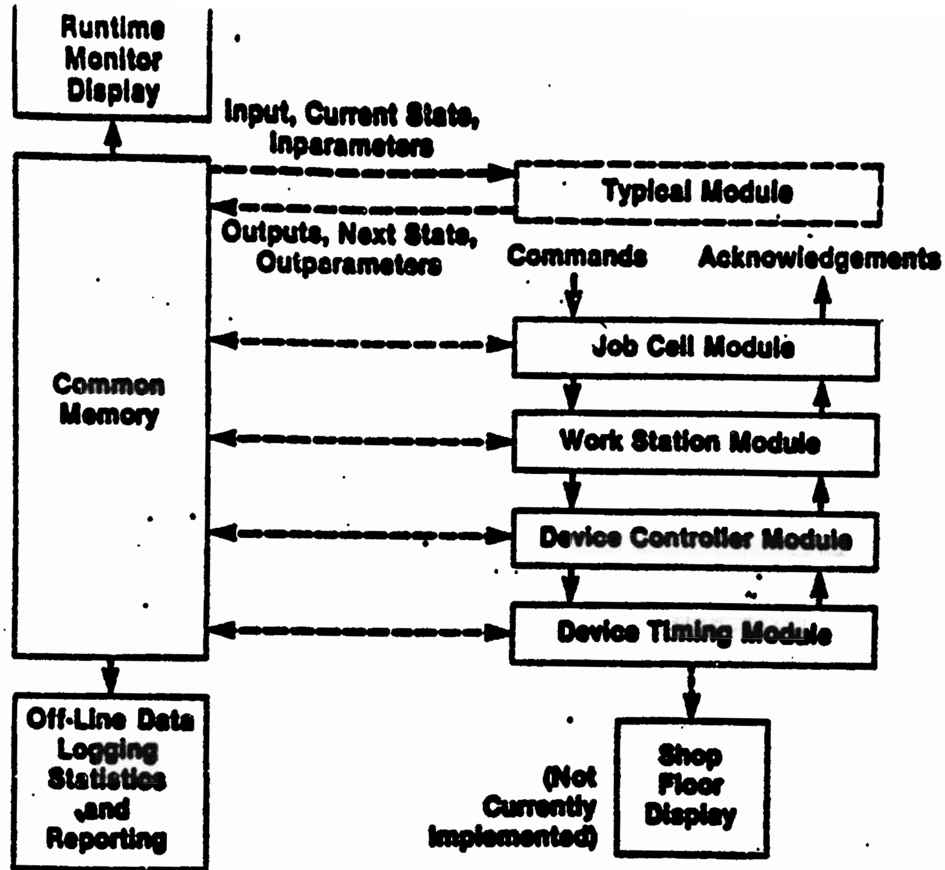


Figure 3. Emulation Structure

Description of the ECSE. The whole of the AMRF is to be emulated with each module (e.g., workstation) on each level distinctly visible to facilitate replacement with the actual hardware. The data base structure will approximate that anticipated for the AMRF and distributed local computational processes will be represented. Communication and computing delays are emulated, along with the allocation of modules to different physical processors.

Both continuous physical processes and decision-making processes are included in the emulation along with the ability to test the results of discrete and continuous sources of error. The level of detail in the emulation is under the control of the user, so that critical operations may be represented with great accuracy while only rough approximations of other subsystems are used.

Data is entered for the emulation entirely by writing module descriptions. Each module has a common format based on a generalized state-machine description with named variables (Figure 4). The first part consists of lines which declare input, output and internal variable names and types. This is followed by a sequence of condition-action pairs that implement the rows of a state table. Procedures written in PRAXIS are last.

Communication between modules, which is transparent to the user, is achieved totally through storage (by name) of common input and output variables in a shared (common) memory. Access to the memory is time-slice synchronized. The pattern and sequencing of input/output transactions between modules may be specified by the user to define a hierarchical relationship of the control system modules.

In order to run the emulation, the modules are combined into subsets which are translated into executable form and run as independent processes. These processes are synchronized through common memory. The run-time display runs as another process, allowing the user to monitor the real-time progress of the emulation. The user can synchronize the actual rate of progress of the emulation through the run-time display to achieve single-cycle operation, wall-clock synchronization with variable time-scaling, or free-running (maximum-speed) emulation. The user can select the variables from common memory which are to be displayed,

Figure 4
Finite State Machine (FSM) Format

```
//name  MODU  AME
//input  INPUT ARIABLE  TYPE
   ...
//inparameter  INPUTPARAMETER  TYPE
   ...
//internal  INTERNALVARIABLE  TYPE
   ...
//outparameter  OUTPUTPARAMETER  TYPE
   ...
//output  OUTPUTVARIABLE  TYPE
   ...
//preprocess  STATEMENT
   ...
//postprocess  STATEMENT
   ...
//conditions  CONDITION1;CONDITION2;...
//actions  STATEMENT1;STATEMENT2;...
[condition-action pairs]
//multimatch  STATEMENT
//nomatch  STATEMENT
//procedures
procedure PROCEDURE_1()
   ...
endprocedure (PROCEDURE_1)
   ...
[more procedures]
procedure  PROCEDURE_N()
   ...
endprocedure (PROCEDURE_N)
[end of file]
```

and may stop the emulation to record "snapshots" of common memory at any time.

Prior to running the emulation, the user may select variables for 'logging' purposes. Upon completion of a run, these logging files are processed to produce summary statistics that show the values taken by each logged variable and the amount of time spent at each value, along with a sequential logging file.

__Example__. Figure 5 shows a very simple two module example to illustrate the way finite state machines are written. Module COUNT1.FSM resets to -10 and thereafter increments (like a simulated integrator or counter) on each tick until it is reset again. Module COUNT2.FSM observes the output of COUNT1 through common memory and issues a "RESET" command to COUNT1 when its count reaches +10 (like a reset controller). These two modules are combined into a single emulation/simulation process which will only work correctly if the two modules successfully communicate via common memory.

The data dictionary output for these two modules is shown in Figure 6. The current state of each module (CURS) is always available from common memory, even if it has not been explicitly mentioned. The data dicionary allows the user to rapidly check that modules are consistent in their naming conventions and that minor typographical errors have not occurred in variable names. These errors may otherwise go undetected, with the emulation running but without the correct connection being made through common memory.

Each time the value of a variable changes in common memory, it is recorded in the logging file (Figure 7). This is an excellent aid in determining logical errors in communicating between modules. The logging file is the only diagnostic which contains the actual values assumed by variables, as these are not readily determined until run time.

Summary statistics are available that show the values taken by each variable in common memory, as well as the amount and percentage of time spent at each value (Figure 8). This figure shows, among other facts, that there were 210 changes in the integer variable "COUNT", which ranged from -10 to 11, and that the variable "COMMAND" spent about 10% of the emulated time at the value "RESET".

__Emulation of AMRF Control Structure__. This emulation follows the AMRF control hierarchy (Figure 2), specifically demonstrating the concept of the virtual cell. A job cell is created each time that the production of a new part (or batch of similar parts) is scheduled to begin. The job cell acquires the resources for each stage in the production of the part, assures that the correct sequence of subtasks is performed to machine the part, and then disappears once the part has been returned to inventory. Three workstations are available in this example: an inventory workstation, a transportation workstation, and a milling workstation. The inventory workstation controls a carousel, the transport workstation controls a cart, and the milling workstation controls the transfer of trays, a robot, and a vertical milling machine.

The job cell in this example first retrieves a specified piece from inventory, transporting it to the milling workstation. It then takes control of the milling workstation which activates the tray transfer. The robot picks up the part and places it on the milling

Figure 5. Two Module Example Using the ECSE

```
//name count1   | resets to -10 and then increments on each tick
//              | until reset
//
//input command string   | command from count2
//
//output count integer   | counting variable
//
//conditions command = "UP"
//actions count := count +1      | do the counting
//
//conditions command = "RESET"
//actions count := -10
//
//multimatch nexts := "MULTI"
//nomatch nexts := "NOMATCH"


//name count2   | observes count1.fsm output through common memory
//              | and issues a reset command to count1 when the
//              | variable count >= 10
//input count integer     | counting variable
//
//output command string   | command to count1
//
//conditions first_entry
//actions nexts := "RUNNING"
//
//conditions curs="RUNNING"; count < 10
//actions command := "UP"
//
//conditions curs="RUNNING";  count >= 10
//actions command := "RESET"
//
//multimatch nexts := "MULTI"
//nomatch nexts := "NOMATCH"
```

Figure 6. Data Dictionary Listing

```
COMMAND      STRING
    Written by : COUNT2
       Read by : COUNT1
    comments:
      COUNT1 -  command from count2
      COUNT2 -  command to count1

COUNT      INTEGER
    Written by : COUNT1
       Read by : COUNT2
    comments:
      COUNT1 -  counting variable
      COUNT2 -  counting variable

COUNT1_CURS      STRING
    Written by : COUNT1
       Read by :
    comments:

COUNT2_CURS      STRING
    Written by : COUNT2
       Read by :
    comments:
```

Figure 7. Sequential Logging File

```
0: 0: 0.00    COUNT1_CURS  NOMATCH
0: 0: 0.00    COUNT2_CURS  RUNNING
0: 0: 0.10    COMMAND      up
0: 0: 0.20    COUNT            1
0: 0: 0.30    COUNT            2
0: 0: 0.40    COUNT            3
0: 0: 0.50    COUNT            4
0: 0: 0.60    COUNT            5
0: 0: 0.70    COUNT            6
0: 0: 0.80    COUNT            7
0: 0: 0.90    COUNT            8
0: 0: 1.00    COUNT            9
0: 0: 1.10    COUNT           10
0: 0: 1.20    COUNT           11
0: 0: 1.20    COMMAND      RESET
0: 0: 1.30    COUNT          -10
0: 0: 1.40    COMMAND      up
0: 0: 1.50    COUNT           -9
0: 0: 1.60    COUNT           -8
0: 0: 1.70    COUNT           -7
0: 0: 1.80    COUNT           -6
0: 0: 1.90    COUNT           -5
0: 0: 2.00    COUNT           -4
0: 0: 2.10    COUNT           -3
0: 0: 2.20    COUNT           -2
0: 0: 2.30    COUNT           -1
0: 0: 2.40    COUNT            0
0: 0: 2.50    COUNT            1
```

Figure 8. Summary Statistics

```
There were a total of 241 transitions.
Page faults  last value = 74
Direct I/O  last value = 3
Buffered I/O  last value = 5
CPU Time (10 msec units) last value=312
Elapsed Time  last value = 00:00:22.74
Total common mem writes, last value=919
Total common mem reads, last value=928
COUNT  last value = 9
    Number of transitions = 218
    Minimum -10 Maximum 11
COMMAND  last value = up
    Number of transitions = 21
    Other values
     Duration    %    value
     0:00:02.5  10   RESET
     0:00:22.3  89   up
COUNT2_CURS  last value = RUNNING
COUNT1_CURS  last value = NOMATCH
```

machine. The piece is then machined into a part, and these steps are repeated in reverse order.

Figure 9 shows four instances of emulated time as the two virtual cells appear and control different work-stations. The five modules at the bottom of the structure contain simulations of the hardware in the system. The actual hardware can be substituted in place of the simulations, with the emulation running in real time.

Future Version of the Emulator. The emulator is presently being redesigned to improve its use as a design tool. An interactive state-table editor will be the user interface to the emulator. This full-screen editor will be in tabular form and will allow the user to input and modify state-tables in text format. The editor will have checks built in to prevent inconsistent statements. The editor will output the state-transistion matrix to an interpreter. The interpreter will implement the common memory, data base interfaces, and network communications functions that are necessary for the AMRF control systems. These functions will be built from primitives that are previously defined as part of the language.

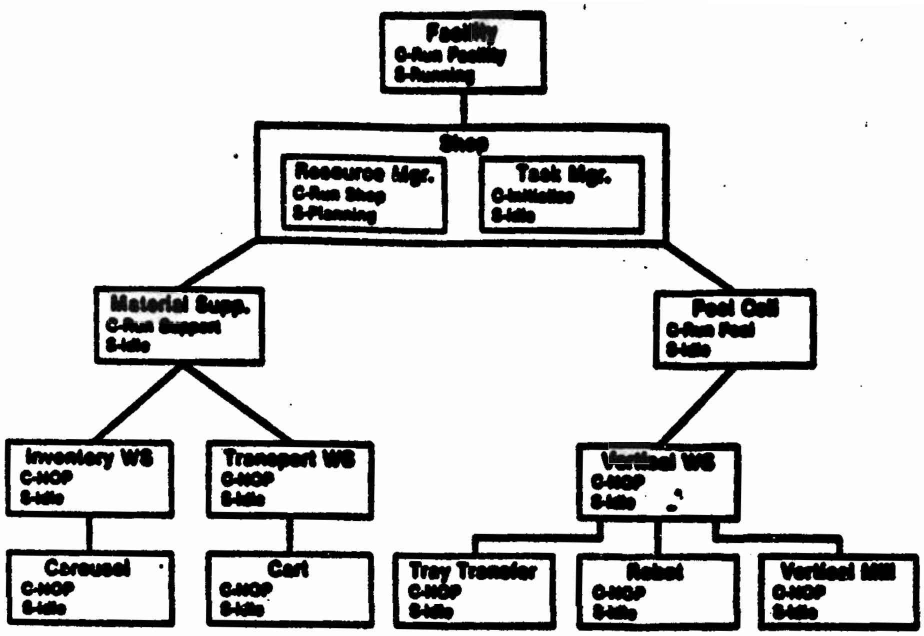
The interpreter will permit the user to dynamically build and interchange state-tables. It will be possible to "collect" state-table programs to implement a model. A library of predefined modules or routines will be implemented. These action routines will conform to standards that will identify what they will read and what they will write. These routines will permit the simulation of processes in any language that uses a standard calling routine. After compilation, these link-edited libraries will be installed in a 'shareable library' which will permit the interpreter to dynamically invoke the action routines on demand from a finite state machine. The emulator is being designed to be transportable, but speed for real time control is not necessary as the state tables that are created with this design tool can be transcribed into the ECSS for real time control of the AMRF.

Simulation

Models of the basic control and information flow of the AMRF will be developed using conventional simulation techniques. Simulation techniques that include combinations of continuous, discrete, and network models will be needed to describe the AMRF. SLAN II, a simulation language suitable to manufacturing environments, is currently being used to study the material transport system. Algorithms for job scheduling, routing, cart movements, etc., will also be tested using these models. Model building tools will include graphics input as well as data acquisition interfaces to data bases to acquire information such as failure rates from operating facilities. Interfaces to computer-aided-design systems will be devised for input to the simulations. The facility managers will be able to use the models to determine the answers to "what if" situations. Graphical presentation tools will include the animated display of the positional flow of objects in the simulations.

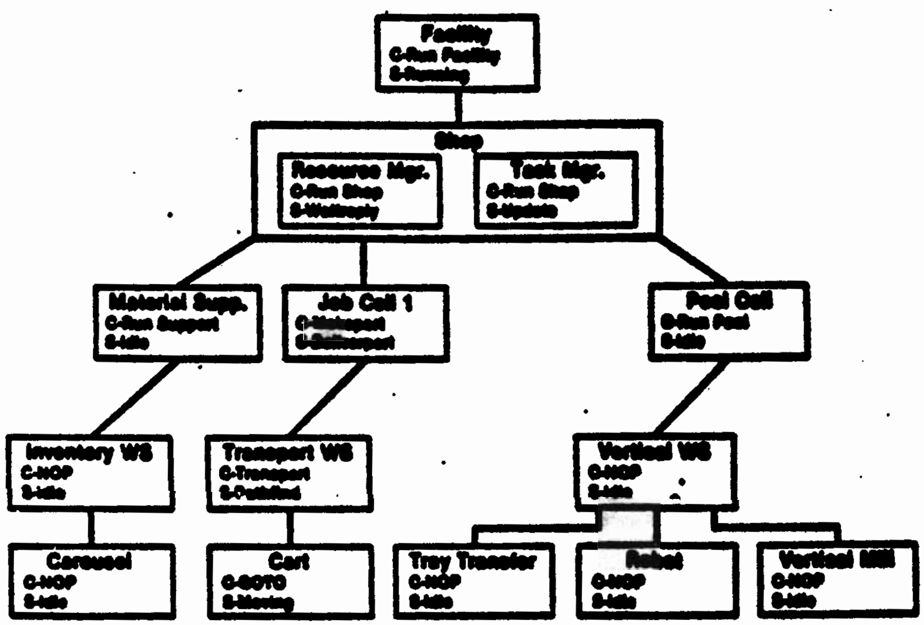
CONCLUSION

This paper has described the research efforts at the National Bureau of Standards to construct an Automated Manufacturing Research Facility. The AMRF, like other Bureau facilities, will be made available to university and industrial groups for nonproprietary research in manufacturing engineering.



SNAPSHOT AT EMULATED TIME OF 30 CLOCK TICKS

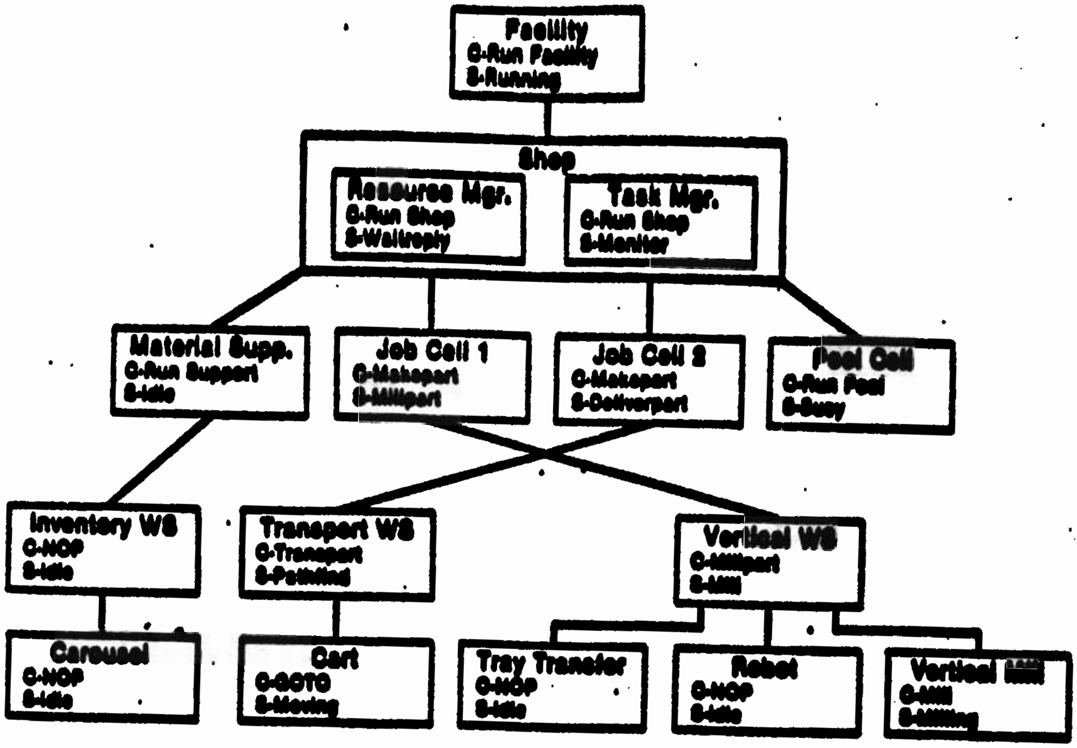
INITIALIZATION OF THE FACILITY



SNAPSHOT AT EMULATED TIME OF 152 CLOCK TICKS

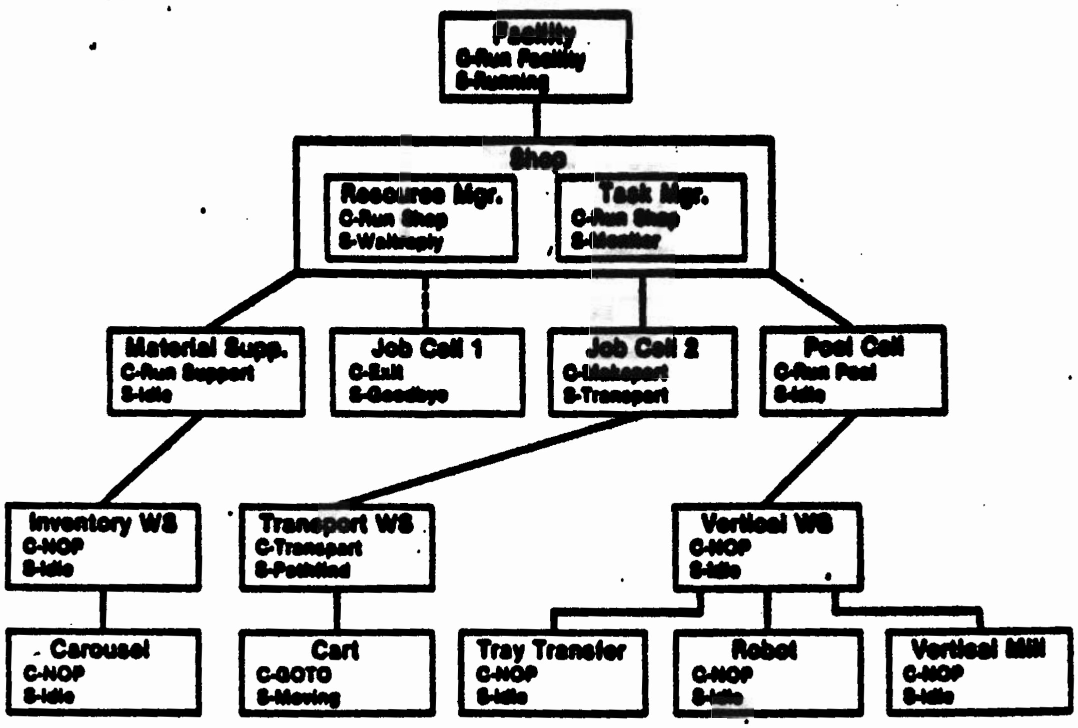
FIRST JOB CELL CONTROLLING TRANSPORTATION WORK STATION

Figure 9. Emulation of AMRF Control Structure



SNAPSHOT AT EMULATED TIME OF 358 CLOCK TICKS

FIRST JOB CELL CONTROLLING VERTICAL MILL WORK STATION
SECOND JOB CELL CONTROLLING TRANSPORTATION WORK STATION



SNAPSHOT AT EMULATED TIME OF 769 CLOCK TICKS

FIRST JOB CELL HAS EXITED
SECOND JOB CELL CONTROLLING TRANSPORTATION WORK STATION

Figure 9 (Continued). Emulation of AMRF Control Structure

ACKNOWLEDGMENT

This paper has been prepared to summarize the preliminary results and integrated reports by many NBS staff and AMRF project team members, too numerous to list and properly credit here.

The NBS Automated Manufacturing Research Facility is partially supported by funding from the Navy Manufacturing Technology program.

The Emulation Project is sponsored by the Air Force/DARPA ITA Project.

REFERENCES

[1] Simpson, J.A., Hocken, R.J., Albus, J.S., The Automated Manufacturing Research Facility of the National Bureau of Standards, Journal of Manufacturing Systems, Vol. 1, No. 1, 1982.

[2] Albus, J.S., Barbera, A.J., Nagel, R.N. Theory and Practice of Hierarchical Control, Twenty Third IEEE Computer Society International Conference, September, 1981.

[3] Johnson, T.L., Milligan, S.D., Fortmann, T.E., HCSE User's Manual, National Bureau of Standards, NBS-GCR-82-413, October, 1982.

[4] Johnson, T.L., Milligan, S.D., Fortmann, T.E., HCSE Applications Guide, National Bureau of Standards, NBS-GCR-82-410, October, 1982.

[5] Johnson, T.L., Milligan, S.D., Fortmann, T.E., HCSE Programmer's Manual, National Bureau of Standards, NBS-GCR-82-414, October, 1982.

[6] Johnson, T.L., Milligan, S.D., Fortmann, T.E., of BBN, Inc., Cambridge, MA 02238, Bloom, H.M., McLean, C.R., and Furlani, C., of U.S. Dept. of Commerce (NBS), Washington, D.C. 20234, Emulation/Simulation of a Modular Hierarchical Feedback System, The 21st IEEE Conference on Decision and Control, Orlando, FL, December 8-10, 1982.

[7] McLean, C.R., Bloom, H.M., Hopp, T.H., The Virtual Manufacturing Cell, Fourth IFAC Symposium, Information Control Problems in Manufacturing Technology, Gaithersburg, MD, October 26-28, 1982, McGregor and Werner.

[8] Hopp, T.H., Lau, K.C., A Hierarchical, Model-Based, Control System for Inspection, First International Symposium on Automated Integrated Manufacturing, San Diego, CA, April 5-6, 1983.

[9] J. S. Albus, E. W. Kent, M. Nashman, P. Mansbach, L. Palombo, M. Shneier, A 6-D Vision System., Proc. SPIE Technical Symposium, Crystal City, VA., May 1982.

[10] E. W. Kent, Hierarchical Control of Robot Vision by Internal Models. Proceedings, Control Engineering Conference, Chicago, Il, May 24-26, 1983.