A CONTROL SYSTEM FOR AN AUTOMATED MANUFACTURING RESEARCH FACLITY

by James S. Albus, Anthony J. Barbera, M.L. Fitzgerald, Ernest Kent, Charles McLean, Harry McCain, Howard Bloom, Leonard Haynes, Cita Furlani, Edward Barkmeyer, Mary Mitchell, Harry Scott, Donald Bloomquist, Roger Kilmer

Presented at:

Robots 8 Conference and Exposition
Detroit, Michigan
June 4, 1984

ABSTRACT

A hierarchical architecture for real-time planning and control has been implemented in the first Cell of an Automated Manufacturing Research Facility at the National Bureau of Standards. Three workstations (A horizontal milling, a turning, and a materials handling workstation) have been implemented. The horizontal and the turning workstations have robots, and the horizontal has a 6-D robot vision system interfaced with a RCS (Real-time Control System) robot controller. A communications network, a distributed data base and a simulator/emulator have also been implemented.

A CONTROL SYSTEM FOR AN AUTOMATED MANUFACTURING RESEARCH FACILITY

James S. Albus, Anthony J. Barbera, M.L. Fitzgerald, Ernest Kent, Charles McLean, Harry McCain, Howard Bloom, Leonard Haynes, Cita Furlani, Edward Barkmeyer, Mary Mitchell, Harry Scott, Donald Bloomquist, Roger Kilmer.

INTRODUCTION

The concept of a hierarchical control system for an automatic machine shop has been described in a number of previous publications. [1,4,5] The control hierarchy shown in Figure 1 has been designed for an Automated Manufacturing Research Facility (AMRF) now being built at the National Bureau of Standards. [8,13] The computational activities performed at each level of this hierarchy are shown on the left of this diagram. Figure 2 is a diagram of the part of the control hierarchy which has been integrated to date. Three workstations have been implemented with varying degrees of complexity.

TURNING WORKSTATION

The turning workstation represents the first degree of complexity. The turning workstation consists of a workstation controller, a turning center, a turning center controller, a robot, a robot controller, and a roller table materials loader/unloader. The workstation controller provides closed-loop control over all the elements in the turning center workstation. Software for the workstation controller is written in PL/M, a structured language.

The workstation controller receives commands from the cell controller via an RS-232 link. These commands define which items are to be made and how many of each. The workstation controller returns status information indicating that the commands have been received and that the commanded process either is executing, is completed, or has failed. The workstation controller can request part programs for the turning center from the AMRF database. These programs are then downloaded into the turning center controller via an RS-232 link for execution.

The workstation controller also can send commands to the turning center controller through a keyboard input simulator which was designed and constructed at NBS. This allows the workstation controller to perform all the machine tool keyboard entry functions that a human operator must otherwise perform to load and unload parts and set tooling. This interface approach is somewhat slow because the turning station controller requires 80 milliseconds to accept each character. The delay is due to a controller system keyboard entry routine which checks to see that two keys are not depressed simultaneously.

In the present configuration, the robot controller in the turning workstation can only be programmed in the teach mode, and the workstation controller can only command the robot controller to execute one or another of the taught programs. The robot controller returns a DONE status when the robot program execution reaches an end-of-program statement. The workstation controller also can read a tray-arrival signal from the load/unloader and can issue a LOCK-TRAY command to pneumatically actuate locating pins beneath the incoming tray.

MATERIALS HANDLING WORKSTATION

The materials handling workstation consists of an inventory request terminal, a robot cart controller, a cart radio frequency communication modem, and a robot cart with an on-board microcomputer. The materials handling workstation controller is implemented in the Hierarchical Control System Emulator on the AMRF VAX-11/780 research computer.

The materials handling workstation controller receives commands from the cell controller of the form: STARTUP, REPLENISH, RESET, and SHUTDOWN. It responds with status reports of EXECUTING, DONE, or FAIL.

When the materials handling workstation controller receives a REPLENISH command from the cell, it accesses the AMRF database to retrieve a kitting order, verifies that the order can be filled with the stock currently in inventory, and displays to the human operator at the inventory terminal the required layout of the tray to be loaded. The operator uses this display to determine which raw material blanks to place in what sectors of the parts tray. In the near future, a similar set of commands and displays will be implemented for tool trays to indicate what types of tools should be placed in what sectors of the tool tray. In the more distant future the human operator at the inventory request terminal will be replaced with an automatic storage and retrieval system which will have sufficient artifical intelligence to load part and tool trays with the required materials and tools and respond with the appropriate status reports.

The workstation controller decomposes its input commands into cart commands of the form: INIT-CART, SET-TRACK, GOTO-MAGNET, GOTO-STATION, LOAD, UNLOAD. Upon the receipt of the appropriate command, the cart negotiates a network of guide-wire pathways to pick up and drop off trays of parts at the designated load and unload stations.

The cart reports status of the form: READY, FAIL, UNRECOGNIZED COMMAND, MOVING, TRAY-ON-CART, TRAY-AT-STATION, SAFETY-STOP, MAGNET-STOP, STATION-STOP, GUIDEWIRE-LOST. The workstation controller interprets the status feedback from the cart to determine if its commands have been successfully completed. If it finds that the cart path is blocked or there is some other simple problem, it will ask the human tender to intervene. If that is not possible, the workstation will report a failure to the cell.

HORIZONTAL MACHINING WORKSTATION

The horizontal machining workstation is the most sophisticated workstation currently implemented in the AMRF. It consists of a workstation controller, a horizontal machining center, an industrial robot, a sensory-interactive robot control system, an automated fixturing system, high level machining controller, a fixture controller, a roller table load/unloader, a safety system, a wrist mounted vision system, and gripping force sensors on robot finger tips.

The workstation controller receives commands from the cell of the form MAKE-BATCH. It decomposes MAKE-BATCH commands into sequences of commands:

> to the robot of the type ACQUIRE, MOVE, RELEASE, or TRANSFER PARTS;
>
> to the machine tool controller of the type EXECUTE PART PROGRAM, POSITION TABLE, FEEDHOLD;
>
> and to the fixture controller of the type: LOCK, UNLOCK, OPEN, HOME.

Most of these commands carry parameters which define the locations of parts, tools, and fixtures. The workstation controller has a communications interface to the AMRF database which it uses to retrieve resource plans and control programs for making the different parts which make up the batches. The workstation controller coordinates all activities within the workstation through communication with various equipment level controllers including the NBS built robot controller, high level machining controller and the fixturing controller.

The workstation controller receives status reports from each of the equipments it controls of the form:

> EXECUTING, DONE, FAIL.

It returns status reports to the cell of the form:

> COMMAND VALID, LOCAL/REMOTE-MODE, READY, SETUP, PRODUCTION, TAKEDOWN, NOT-READY, BATCH-COUNT, PART-BUFFER-STATUS, TRAY-IDENTIFICATION.

The robot controller is a Real-time Control System (RCS) [2,3] which consists of four levels of task decomposition. It has an interface to a vision processor [15,16] which answers questions about the identity, position, range, orientation, and surface normal vector of parts located in various sectors of the parts trays. Both the vision system and the robot controller contain databases which describe the geometry of the parts. They use this information to select grip points and define parameters for the vision system. In the near future, these databases will be interfaced to the AMRF database so that part geometry information can be accessed automatically from the CAD information contained in the facility database .

The RCS robot controller also is interfaced to a watchdog safety computer [17] which monitors the performance of the robot so as to detect excessive velocity or acceleration. The safety computer has stored in it a set of forbidden volumes, which it prevents the robot end-point from entering. The watchdog computer currently monitors the position values of the robot joints as well as the status of the RCS robot controller and uses a HOLD-SET input to the manfacturer's robot controller to stop the robot if an unsafe condition is detected.

In the future, the safety computer will have a number of additional sensors, such as safety mats and ultrasonic sonar systems, to detect obstacles and intruders. These will enable the watchdog to inform the RCS controller of the position and velocity of obstacles or intruders so that it can move the robot to a safe position until the obstacle or intruder is removed. The watchdog will retain the ability to do an emergency stop on the robot in the case of any detected system failures.

The High Level Machining Controller allows full functionality and total control over the Horizontal Machining Center from a remote site. It utilizes both commercially available machine vendor communications options and special wire links added to exert control over all machining and controller functions. In addition, it allows all status messages at the machine controller to be reported to higher levels of control. The High Level Machining Controller accepts commands from Workstation Control as to which machining sequence to execute, requests the necessary NC programs from the AMRF database, downloads them into the NC machine, and initiates execution via a CYCLE START command.

The fixture controller can operate several clamps to hold a variety of parts for machining. The materials transfer system can receive commands to move trays and report status on when trays are properly positioned.

THE CELL CONTROLLER

The function of the cell controller is to manage and coordinate the performance of all part production and support tasks in the three workstations currently comprising the AMRF [6]. In performing this function, the cell is required to accept orders input from an operator command terminal, to enter these orders in a queue, to group the parts in the queue into batches that can be loaded in trays, to select the tooling needed by the various machining workstations, to route the tools and parts from one workstation to another, and to monitor the progress of the batches through the various workstations that comprise the cell.

The cell controller is composed of three levels as shown in Figure 3:

1) Queue configuration management -- This level accepts input from an operator command module (which eventually will become the Shop Controller). It analyses the cell configuration requested,

issues the appropriate commands to the workstations, and reports actions back to the operator command module. It also analyses operator requests for parts, enters those requests in the queue, retrieves from the database the operations that need to be done to each part, and assigns the tasks required to the appropriate schedulers. It receives feedback from the machining workstations schedulers, and uses that feedback to update the request queue and compute material requests to the materials handling workstation. Future versions of the queue configuration manager will decide what parts should be batched together, what tooling will be required for each batch, and will compute routings for those batches that make most efficient utilization of workstation resources.

2) Scheduling -- There is a scheduling module for each workstation. These scheduling modules generate the sequence of tasks to be performed at the respective workstations, and issue to the corresponding dispatcher modules the next task in that sequence. They also monitor the dispatching module, clear the scheduler queue of all cancelled and completed tasks, and update the status of new and in-process tasks.

3) Dispatching -- This level communicates with each of the workstations within the cell's sphere of control issuing task commands, receiving status reports, and monitoring how well the schedule is being followed.

The cell issues commands to the materials handling workstation to SETUP-PART-TRAYS, SETUP-TOOL-TRAYS, ROUTE-TRAYS; and to the machining workstations to MAKE-BATCHES-OF-PARTS. It receives status reports of DONE, EXECUTING, or FAIL.

COMMUNICATIONS

Data communication between computing processes in the AMRF takes place via a communications network that is transparent to the computing processes [7]. Each process merely writes to and reads from what appears to it to be common memory locations. Each process that has information intended for another process "writes" that information into a "mailbox" in a conceptual AMRF-wide database, from which the intended receiver "reads."

In reality, much of the AMRF-wide database consists of memory areas in the computer systems which contain the control and sensory processes. The sending process stores the information it wishes to send in a block of memory designated as its "mailbox." In the case where both the sending and receiving process are resident on the same computer, the receiver simply reads from the "mailbox" area of memory into which sender writes. In the case where the sending and receiving processes are resident on different computers, a communications process copies the "mailgram" from the sender's "mailbox" into the receiver's. A Network Interface Process (NIP), diagramed in Figure 4, provides the communications protocols needed to make the message passing transparent to the sending and receiving processes [7].

The advantage of this type of communication system is that it encourages the development of standard functional information formats for interprocess communication. One of the major goals of the AMRF is to develop interface standards between computing machines and processes in an automated manufacturing environment. The definition of mailboxes which appear to the sender and reciever to be simply areas in common memory is a major step in this direction.

There are, of course, some difficulties with this form of communication. First, there must be some mechanism to prevent the receiver from reading a mailgram while a sender is writing it, or vice versa. This can be solved either by synchronous read/write time slices, or by a system of semaphores. Both methods have been used in the AMRF. Second, the receiver process may run slower than the sender, and hence may miss some mailgrams; or run faster than the sender, and hence may read some mailgrams twice. These problems can be solved by having the sender encode each message with a unique number, and having the receiver echo the received message number back to the sender.

The first implementation of the network shown in Figure 5 uses 9600 baud serial RS-232C links using HDLC link control protocol. Future versions of the AMRF will use a highspeed local area network. The mailboxes and NIP protocol will appear the same to the user processes. Only the speed and response times will improve.

DATABASES

The AMRF is serviced by a distributed Data Administration System (DAS) [7] which provides a uniform method of access to data for all the major control modules. Data residing in the DAS include: 1) Materials to be kitted and delivered to a workstation for fabrication of a batch of parts. 2) Inventory control information and intermediate views of parts in process. 3) Tray names and current status. 4) Tray layouts, current contents, and part position and orientation in the trays. 5) Data resources required by workstation controllers before batch production can begin. 6) Numerical control programs (and version numbers) for machine tools. 7) State table decision logic (and version numbers) for control modules. 8) Customer orders for part fabrication.

The type of data that is required by the various processes at the equipment, workstation, and cell levels is shown in Figure 6. In the future, the database will also contain CAD information which fully describes parts to be manufactured in the AMRF. Extentions to CAD data are being developed so that the database will include information about how parts should be fixtured for machining, how they should be gripped and manipulated by the robots, what tooling will be needed to manufacture the parts, and other information such as surface finish, heat treating, and inspection requirements and tolerances. Work is also progressing on standardization of these extensions so that they can be included in a national standard such as the IGES graphics exchange specification [18].

The structure of the interface between the database and the communications network is shown in Figure 7. The database interfaces with the network through mailboxes just like all other processes in the AMRF. Each user process of the DAS establishes three mailboxes: 1) A command mailbox where requests for service are written. 2) A status mailbox where the DAS responds with the status of the request. 3) A data mailbox where the DAS writes the data satisfying the request.

Commands include: SELECT, INSERT, UPDATE, and DELETE. The system provides a full set of logical and boolean operators for specifying the sets of data to be manipulated. The DAS presents data in the format most directly usable to the control process users. Data requirements are normalized into relational tables which provide a uniform interface so that any one of a number of commercial database management systems can be used in the AMRF.

SIMULATION/EMULATION

A Hierarchical Control System Emulator (HCSE) has been developed in order to facilitate the integration and testing of the AMRF before all the hardware processors are ready for installation [9,10,11,12,14]. This emulator was used in the first test integration run of the AMRF to implement the cell controller, the material handling workstation, the database manager, and the inventory control system. Figure 5 shows which parts of the first integration test run were implemented in the emulation mode on the VAX.

The HCSE has the capability to define modular processes, to pass commands and status through common memory mailboxes, and to run in real-time. This enables systems designers to test software and system interfaces before hardware control modules are implemented. When hardware modules eventually become ready for integration, they can then be plugged into the system in place of their emulated counterparts.

The current version of the HCSE is compiled and optimized to run in real-time so that it can support this system integration and interface test role. A second version of the HCSE is now being developed that will be interpretive and optimized as a program development tool. This version will not attempt to run in real-time, but will be highly interactive with a wide variety of editing, debugging, and diagnostic tools. Code developed on this version-2 interactive HCSE can then be compiled to run on the version-1 real-time HCSE.

CONCLUSION

The NBS AMRF project is now two years into a six year effort, and considerable progress has been made. A first integration test run occurred during an AMRF open house in November of 1983, and numerous additions to that implementation have been made in the succeeding months. The hierarchical architecture has proven itself extremely powerful in defining interfaces between modular processes so that components can be developed independently and then integrated into a system. Very few changes have been made in the original system concept as published previously. The interfaces developed have proven able to accomodate a wide variety of different manufacturer's products, and a number of interfaces have been identified as candidates for interface standards. Much additional work is yet to be done, but the progress to date suggests that a system of interface standards may be possible for interconnecting computer-aided design, process and production planning, scheduling, materials transport, and control systems, with machine tools, robots, sensors and sensory processing, databases, modeling, and communications systems. If so, it may become practical to implement computer integrated manufacturing systems incrementally using components from a variety of vendors.

BIBILOGRAPHY

1. Albus, J.S., McLean, C.R., Barbera, A.J., Fitzgerald, M.L., Hierarchical Control for Robots in an Automated Factory, 13th ISIR/Robots 7 Symposium on System Theory, Chicago, IL, April 1982.

2. Barbera, A.J., Fitzgerald, M.L., Albus, J.S., Concepts For Real-Time Sensory-Interactive Control System Architecture, Proceedings of the 14th Southeastern Symposium On System Theory, April 1982.

3. Albus, J.S., Barbera, A.J., Fitzgerald, M.L., Programming A Hierarchical Robot Control System, 12th International Symposium on Industrial Robots, Paris, France, June 1982.

4. Albus, J.S., Barbera, A.J., Nagel, R.N., Theory and Practice of Hierarchical Control, 23rd IEEE Computer Society International Conference, Sept. 1981.

5. McLean, C.R., Mitchell, M.J., Barkmeyer, E., A Computing Architecture for Small Batch Manufacturing, IEEE Spectrum Magazine, 1983, p. 59-64.

6. McLean, C.R., Bloom, H.M., Hopp, T.H., The Virtual Manufacturing Cell, 14th IFAC Symposium of Information Problems in Manufacturing Technology, October, 1982, p. 105-111.

7. Barkmeyer, E. and Mitchell, M., Data Distribution in the NBS AMRF, Proceedings of the 3rd IPAD Conference, Denver, CO, April 1984. (In press).

8. 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(1):17-32, 1982.

9. Johnson, T.L., Milligan, S.D, Fortmann, T.E., HCSC User's Manual, National Bureau of Standards, NBS-GCR-82-414, October, 1982.

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

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

12. Johnson, T.L., Milligan, S.D., Fortmann, T.E., of BBN, Inc., Cambridge, MA 02238, Bloom, H.M., McLean, C.R., 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.

13. Furlani, C.M., Kent, E. W., The Automated Manufacturing Research Facility of the National Bureau of Standards,Summer Simulation Conference, Vancouver, B.C. July 11-13, 1983.

14. Bloom, H.M., Furlani, C. M., Emulation as a Design Tool in Real-Time Control Systems, Submitted to ACM IEEE 21st Design Automation Conference, Vancouver, B.C., July 11-13, 1983.

15. Albus, J., Kent, E., Nashman, M, Mansbach, P., Palombo, L., Six-dimensional Vision System, SPIE Vol. 336 Robot Vision, 1982.

16. Kent, Ernest W., Hierarchical Control of Robot Vision by Internal Models, Proceedings, 2nd Annual Control Engineering Congress, Chicago, IL, 1983.

17. Kilmer, Roger, McCain, Harry, Juberts, Maris, Legowik, Steve, Watchdog Safety Computer Design and Implementation, Proceedings, Robots 8, Detroit, MI, 1984 (In press)

18. Smith, Bradford M., Brauner, Kalman M., Kennicott, Philip R. Liewald, Michael, Wellington, Joan, Initial Graphics Exchange Specification (IGES) Version 2.0., U.S. Dept. of Commerce (NBS), NBSIR 82-2631 (AF), February 1983.

The AMRF project is funded by NBS and the Navy Manufacturing Technology Program and significantly supported by industry through donations or loans of major components and through cooperative research programs. The Hierarchical Control System Emulator is supported by the Air Force/DARPA ITA project.

Certain commercial equipment and software is identified in this paper in order to adequately specify the experimental facility. Such identification does not imply recommendation or endorsement by the National Bureau of Standards, nor does it imply that the equipment or software is necessarily the best available for the purpose.

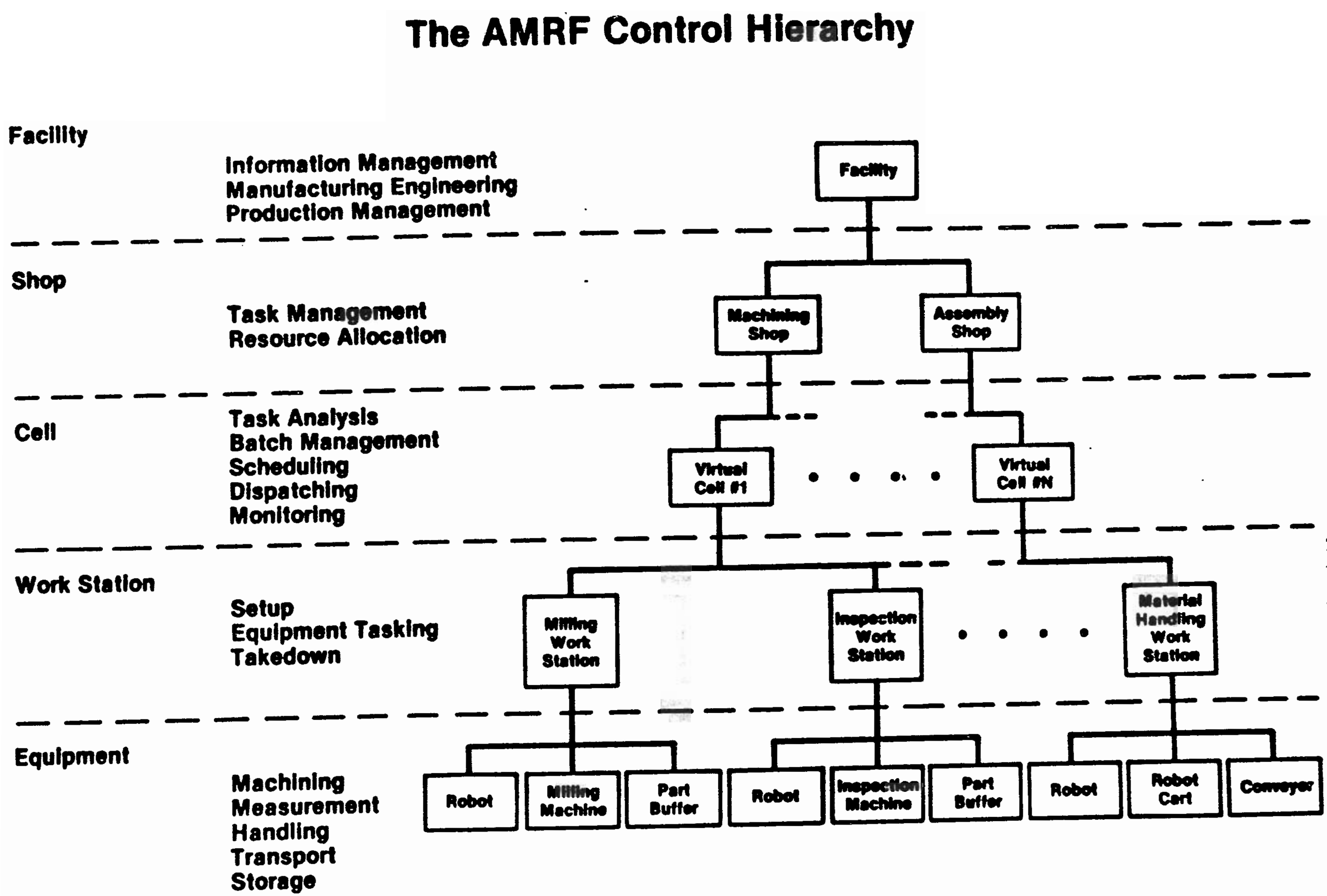


FIGURE 1. A control hierarchy for an Automated Manufacturing Research Facility (AMRF) being constructed at the National Bureau of Standards.

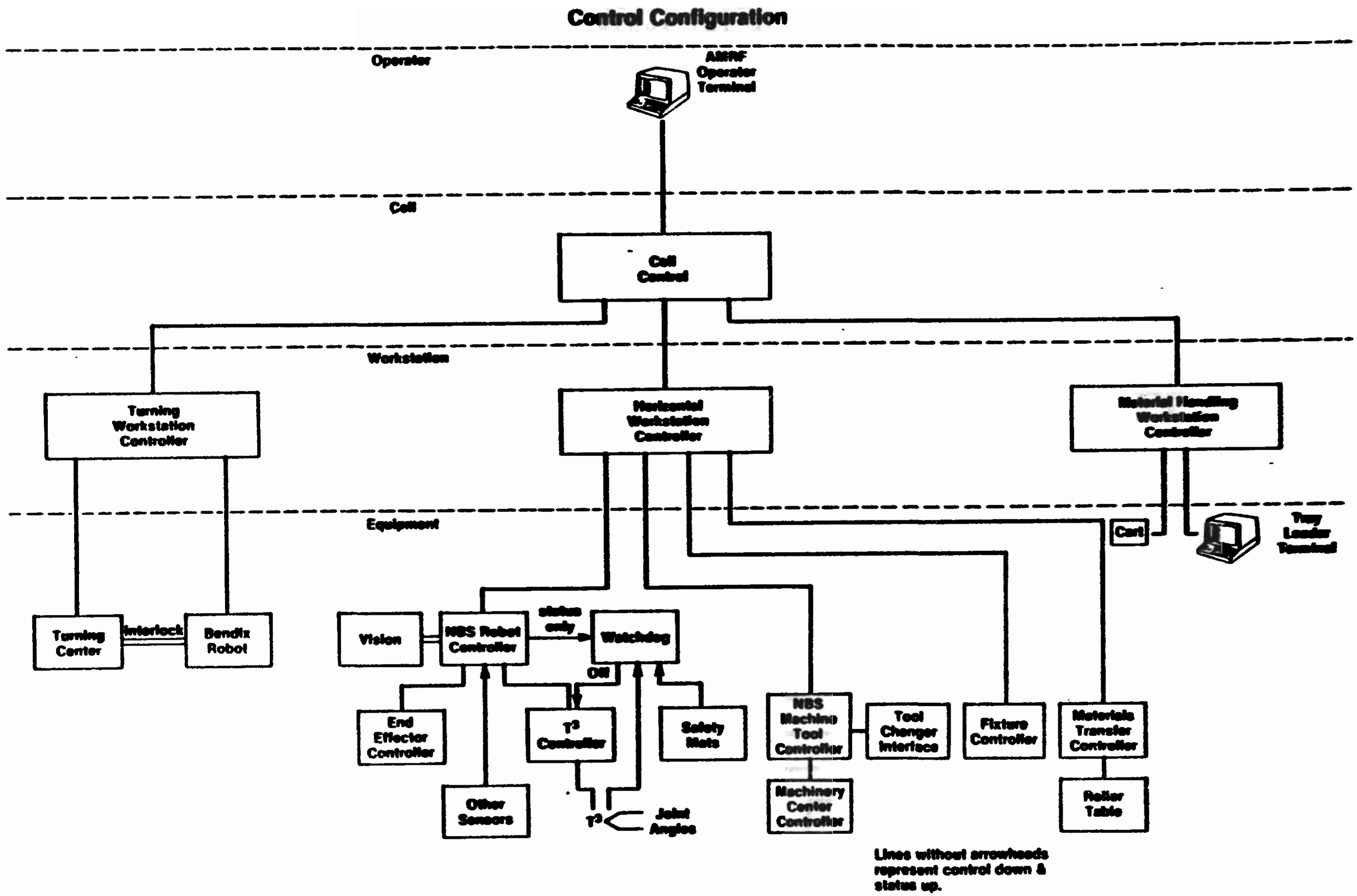
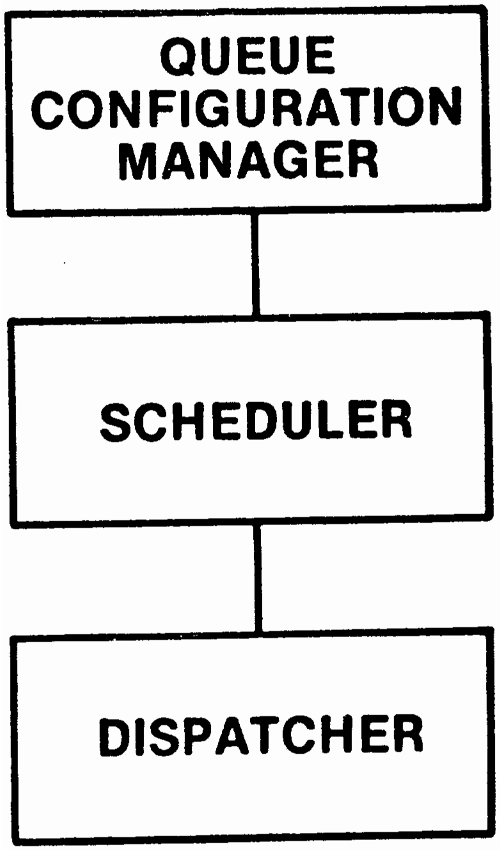


FIGURE 2. A block diagram of the portion of the NBS Automated Manufacturing Research Facility (AMRF) that is currently implemented.

CELL MODULE FUNCTIONS



- Manage Cell Configuration
- Route Production Jobs
- Assign Tasks to Schedulers
- Monitor Scheduler Status
- Create Required Support Jobs

- Check Status of Assigned Tasks
- Schedule Ready Tasks
- Create Required Support Tasks
- Assign Next Task to Dispatcher
- Monitor Dispatcher Status

- Manage Startup and Shutdown of Workstations
- Process Communications Messages Between Cell and Workstations
- Issue Tasking Commands and Monitor Task Progress

FIGURE 3. The Cell Control module shown in Figure 2 is composed of three levels: 1) the queue configuration manager, 2) the scheduler, 3) the dispatcher.

ANATOMY OF A NETWORK INTERFACE PROCESS (NIP)

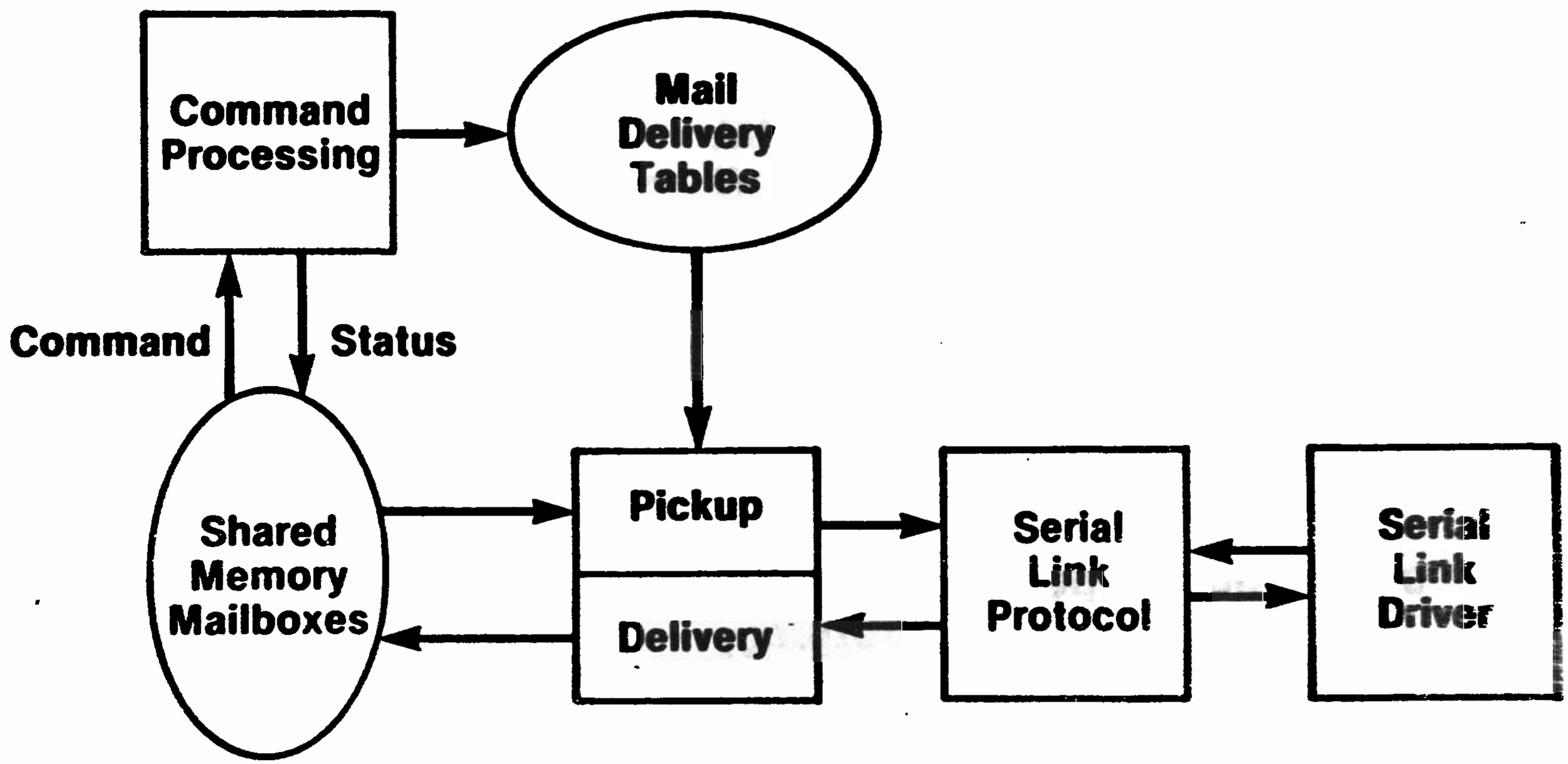


FIGURE 4. A block diagram of the Network Interface Processor (NIP) which handles the pickup, delivery, and communication of "mailgrams" between common memory "mailboxes" in various parts of the AMRF.

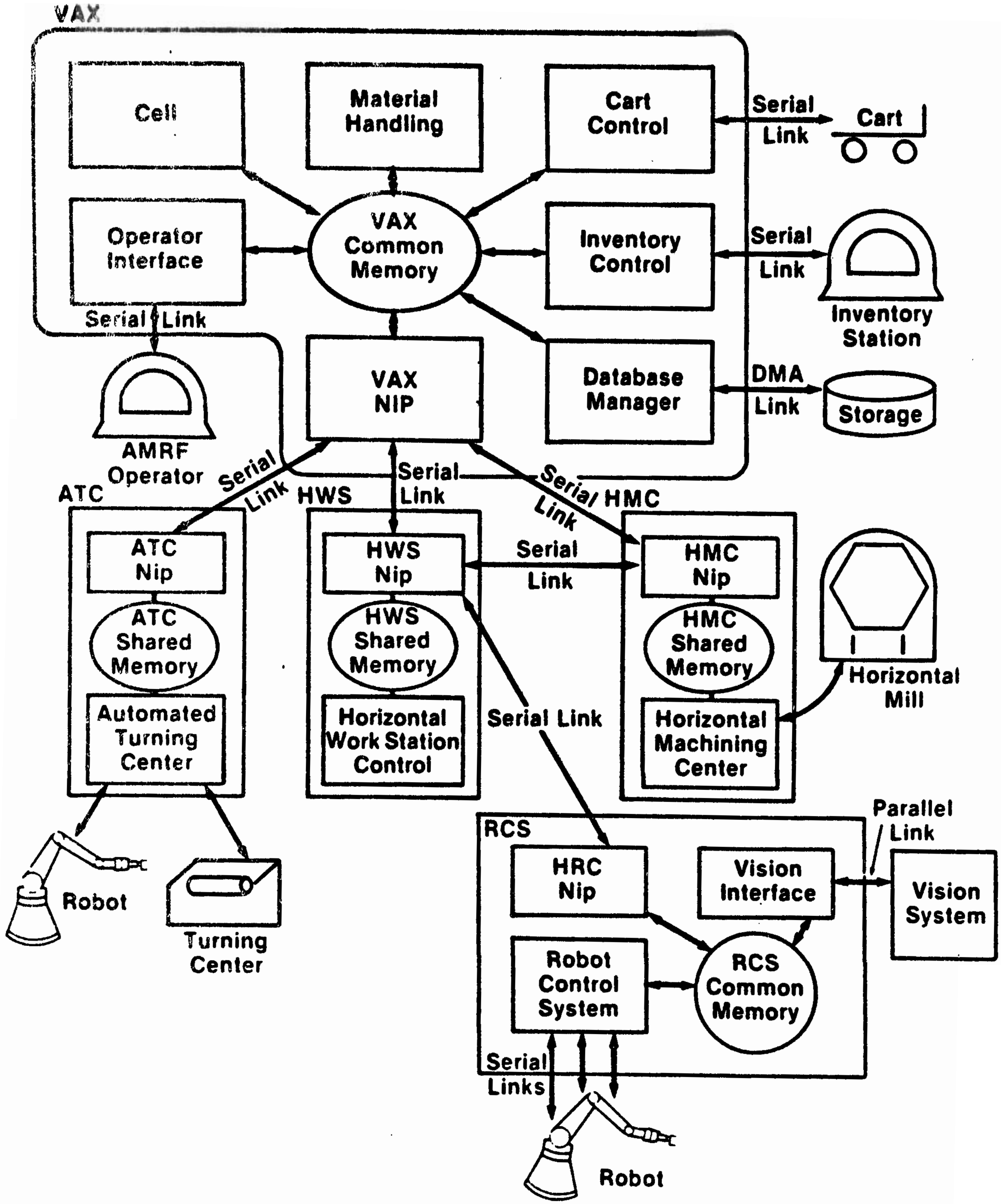


FIGURE 5. A communications network diagram of the currently implemented portion of the AMRF. This network topology corresponds to the control configuration shown in Figure 2.

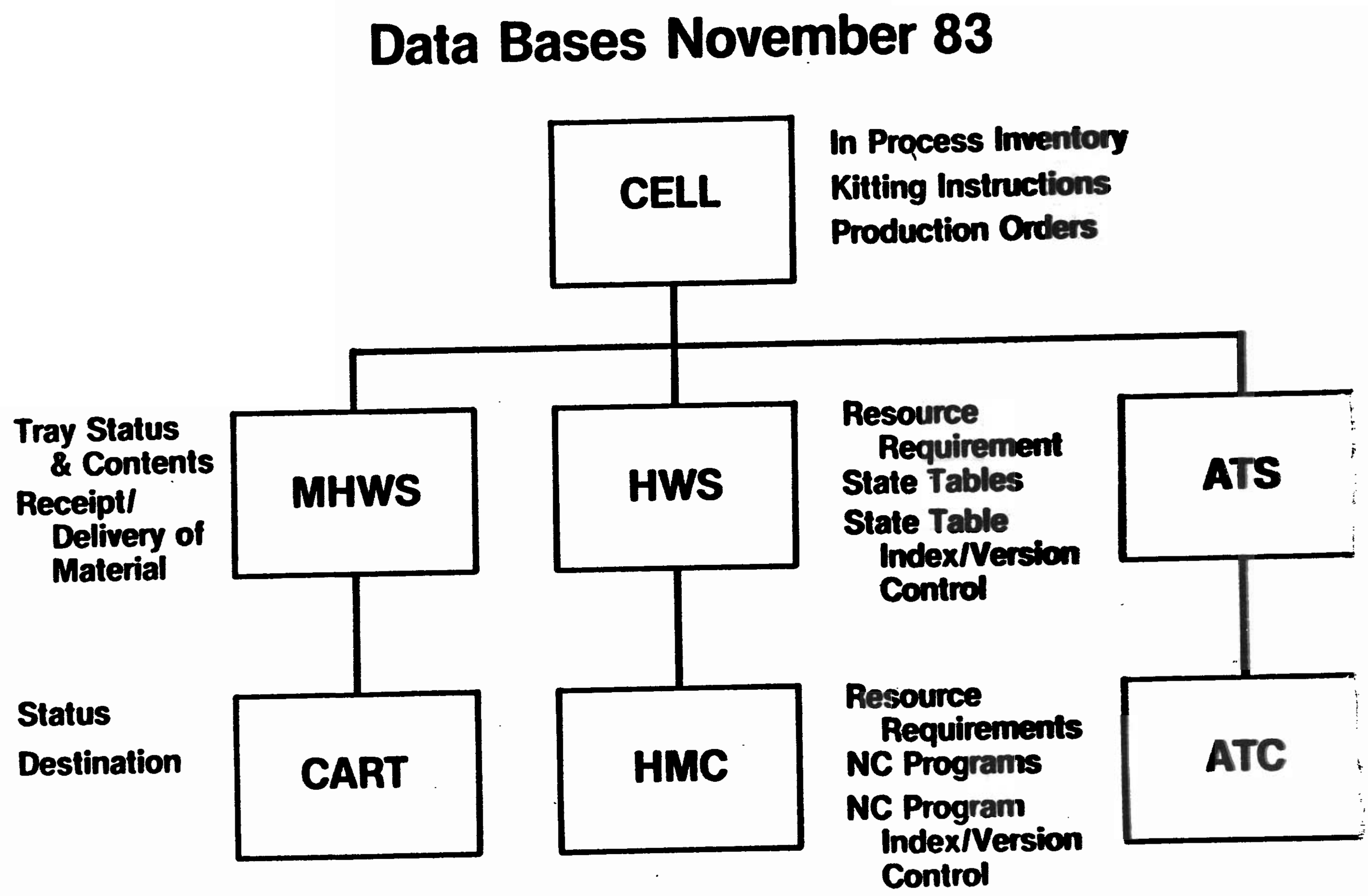


FIGURE 6. The data bases which service the Cell, the Workstation, and the Equipment levels of the AMRF.

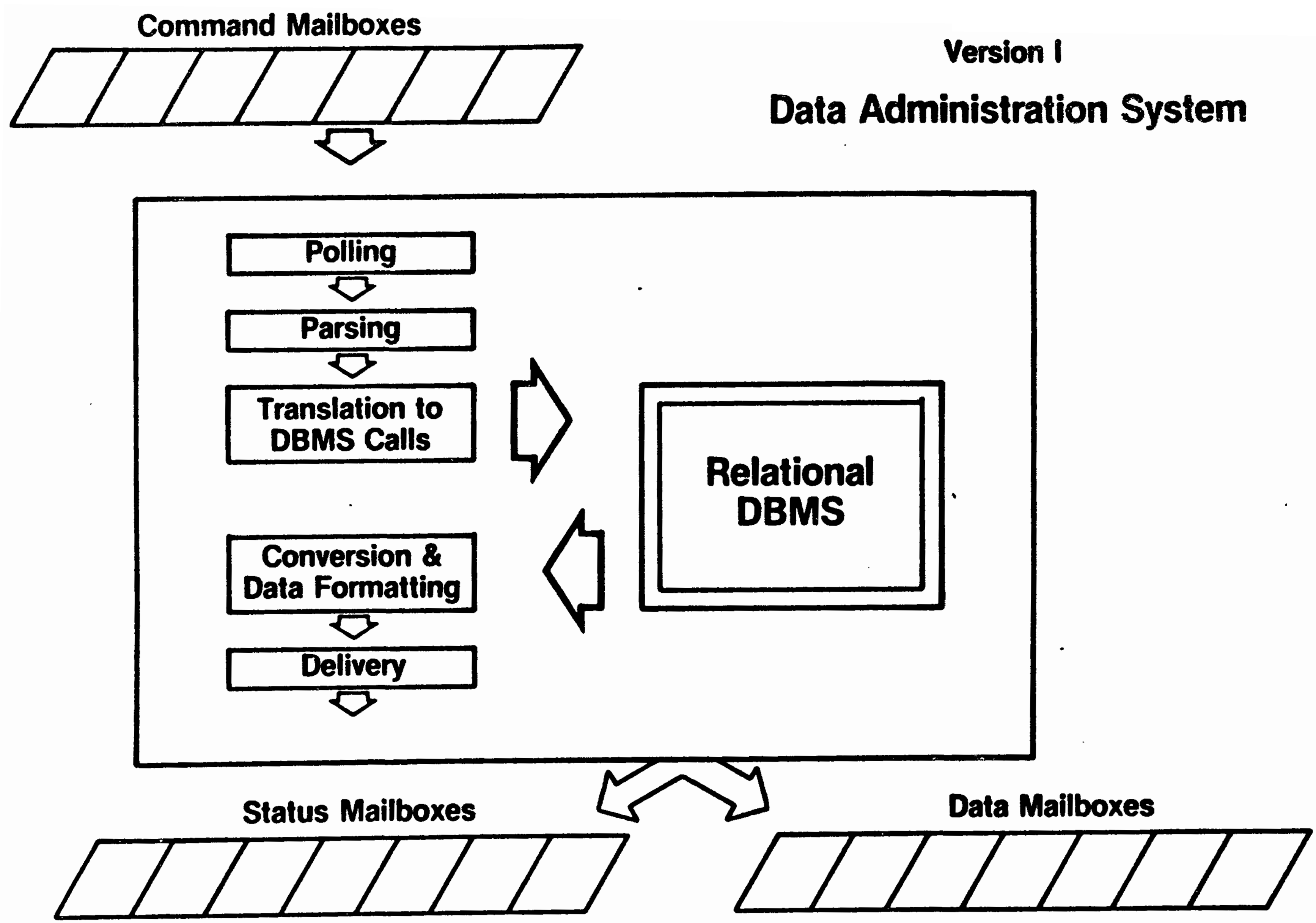


FIGURE 7. The interface between the data base management system and the Data Administration System NIP mailboxes.