HIERARCHICAL CONTROL FOR ROBOTS AND TELEOPERATORS

James S. Albus
National Bureau of Standards
Gaithersburg, MD 20899

Presented at:
IEEE Workshop on Intelligent Control
Albany, NY, August 26-27, 1985

HIERARCHICAL CONTROL FOR ROBOTS AND TELEOPERATORS

J. S. Albus, C. R. McLean, A. J. Barbera, M. L. Fitzgerald

National Bureau of Standards

INTRODUCTION

The basic structure of a hierarchical control system is a tree, wherein each computational module has a single superior, and one or more subordinate modules. The top module is where the highest level decisions are made and the longest planning horizon exists. Goals and plans generated at this highest level are transmitted to the next lower level where they are decomposed into sequences of subgoals. In general, the decomposition at each level takes into account information derived from: (a) processed input data from sensors that measure the state of the environment, (b) reports from lower control levels as to the state of the control hierarchy itself, and (c) predictions (or expectations) generated by models, knowledge bases, or inference engines.

At each level, input commands from the next higher level are decomposed into sequences of output sub-commands to the next lower level in the context of the state of the environment, of the state of the control system, and the internal store of knowledge. At each level predictions and expectations are generated by the internal world model in the context of the state of the task, the goal of the system, and the best current hypothesis about the state of the environment. Also at each level, processed signals from the environment are compared against expectations from the world model. Correlations are computed and differences measured between observation and expectation. A high degree of correlation indicates that the task is proceeding according to plans, and expectations are being met. Differences represent error signals which can be used by the task decomposition module either to modify behavior, or change expectations so that the task goal is successfully accomplished. At the highest level, the input command represents the ultimate goal of the entire organism. At the lowest level, output drive signals are computed and sent to the physical actuators.

Such a hierarchical control system can be used for either robots or teleoperators. In the case of robots, the manipulator system operates automatically without human intervention. In the case of teleoperators, a human operator can enter the control hierarchy at any level to modify or prempt the control commands from the higher levels.

An example of a hierarchical control system for a robot in an automatic factory is shown in Figure 1. A single chain of command from the bottom to the top of such a hierarchy is outlined by the dotted line in Figure 1. This chain of command can be further segmented, as shown in Figure 2, into three separate hierarchies: (1) a goal, or task decomposition, hierarchy (H); (2) a feedback processing hierarchy (G); and (3) a world model hierarchy (M). This has been discussed in a number of previous papers [2,3,4,5].

At all levels, the H, G, and M modules are concurrent processes produced by real-time programs executing simultaneously in each module. Perhaps the simplest way to treat this conceptionally is to model each of the modules in the hierarchy as a finite-state automaton. Each module repeatedly executes a "READ-COMPUTE-WRITE" cycle. At the beginning of each cycle the computing module reads a set of input variables into input buffers. It then performs some computation based on the values of the input variables and state variables within the module. Finally the module writes the results of its computations into output buffers and updates the internal state variables. The activity of each module can thus be described by a state graph, and the activity of the entire hierarchy can be described by a set of state graphs which communicate and syncronize activities through the passage of command, feedback, and status variables.

For each module in this architecture there are three concepts of time: the planning horizon, the response time, and the cycle time. The planning horizon is the interval over which a module plans into the future or analyses the past. The response time is the delay between a change in a module's input and the generation of a new output. The cycle time is the period between sampling the input variables. In general, the response time will be equal to, or slightly longer than, the cycle time. The planning horizon will be many times longer than the response time.

The response time requirements of the finite-state automata at each level depends on the requirements for stability and dynamic response at the respective levels. The response time requirement is shorter at the lower levels, but the complexity of the control computations is less. The response time is longer at the higher levels, and the complexity of the computations is greater. Thus, the total computational power required at any level of the hierarchy is more or less constant.

Communication between the various modules in such a system can be accomplished by writing messages in a database that is common to all modules which either compute or make use of those messages. Each message area (or mailbox) within the database can be restricted so that only one system may write into it, although many can read its contents. If the cycles of the state-clock at all levels are synchronized, information transfer into and out of the common database will occur at predictable time increments and each message can carry a time tag.

TASK DECOMPOSITION

Level 0

At the bottom of the task decomposition hierarchy is the servo level. Input is in terms of desired joint positions, velocities, or forces. Output is voltages to motors or valves.

For a master-slave teleoperator, this is the level of human intervention. Joint angle positions on the master become the desired joint positions input to the servos of the slave manipulator.

Level 1

The next level of the task decomposition hierarchy transforms commanded positions, velocities, and forces expressed in a convenient coordinate system into desired joint positions, velocities, and forces. This level also scales desired joint motions to hardware limits. In the robot control system architecture shown in Figure 2 the bottom (or first) level of the task decomposition hierarchy includes levels 0 and 1.

For a resolved motion rate control teleoperator, this is the level of human intervention. The human moves a joystick, and the Level 1 task decomposition module transforms from the coordinate system represented by the joystick into the desired joint positions and rates of the manipulator.

Level 2

At the second level, robot elemental movements such as <REACH TO (A)>, <GRASP>, <LIFT>, <ORIENT ON (B)>, <MOVE TO (X)>, <RELEASE>, etc. are decomposed into force and velocity trajectories in a convenient coordinate system. That coordinate system may be defined in the robot's work space, in the part, or in a coordinate frame in the robot's gripper.

Human intervention at Level 2 or 3 is usually called

"Supervisory Control". The human imputs commands to the system of the form REACH, GRASP, LIFT, MOVE-TO, etc., and refers to prerecorded points and positions as arguments.

Level 3

At the third level, simple tasks expressed in terms of objects to be manipulated are decomposed into elemental movements which can be interpreted by the second level. Commands to the third level are of the form <FETCH PART (A)>, <MATE PART (B) TO PART (A)>, <LOAD TOOL (C) WITH PART (D)>, etc.

Level 4

At this level, complex tasks to be performed on groups of objects are decomposed into simple tasks performed on one object at a time. In the Automated Manufacturing Research Facility (AMRF) currently under construction at the National Bureau of Standards [9], level 4 is the WORKSTATION CONTROLLER level. The Workstation Controller supervises the activities of a machine tool, a robot, and a number of active clamps and sensing probes. Commands to the fourth level are of the form <MACHINE THE PARTS IN TRAY (X)>.

In the AMRF, trays of parts and tools are delivered to the machining workstations by robot carts which are controlled by a materials transport workstation. It is the task of the machining workstation controller to generate a sequence of simple task commands to the robot, the machine tool, and any other system under its control so that proper set of machining operations are carried out in an efficient sequence. For example, the workstation controller may generate a sequence of simple task commands to the robot to set up the clamping fixtures for the first part; to the machine tool to perform the specified machining operations; to the robot to modify the clamping fixtures for the next job, etc.

The information defining what machining operations need to be performed on each part are stored in a process plan database. Each part to be machined has a part database describing its dimensions, tolerances, and a process plan database describing the sequence of machining processes required to make it. In the AMRF, these databases are available to the Workstation Controller via a communications network. The communication functions are carried out by a Data Administration System. [8]

Level 5

The fifth level of the robot control hierarchy in Figure 2 is the CELL CONTROLLER which is responsible for managing the production of a batch of parts within a particular group technology part family. The task of the Cell Controller is to group parts in trays and route the trays from one workstation to another. The Cell Controller generates dispatching commands to the material transport workstation to deliver the required tools, fixtures, and materials to the proper machining workstations at the appropriate times. The Cell Controller must have planning and scheduling capabilities to analyze the process plans for each part and determine the type of machine required to perform the specified machining operations, the tooling and fixturing requirements, and the machinability time estimates for each operation. The Cell Controller uses these capabilities to optimize the make-up of trays and their routing from workstation to workstation.

Level 6

The sixth level in the robot control hierarchy is the SHOP CONTROLLER which accepts orders, and performs long term production planning and scheduling. It manages inventory, and orders materials and tools to meet production schedules. It determines what workstation resources are required for each cell, and what robot and machine tool resources are required by each cell. The Shop Controller then dynamically allocates workstations to, or reclaims them from the cell controllers as necessary to meet the production schedule [7]. This degree of flexibility becomes important in factories or construction sites where robots are mobile and may move from one physical work site to another.

Level 7

The seventh level is FACILITY CONTROL. It is at this level that engineering design is performed and the process plans for manufacturing each part, and assembling each system, are generated. Here also, management information is analyzed, materials requirements planning is done, and orders are processed for maintaining inventory. Because of the very long planning horizons at this level in the control hierarchy, the activities of the facility control module are not usually considered to be a part of a real-time control system. However, in the context of hierarchical control with exponentially increasing time horizons at each higher level, these facility control activities can be integrated into the real-time control hierarchy of the manufacturing system.

FEEDBACK PROCESSING

Each level of the task decomposition hierarchy is serviced by a feedback processing module which extracts the information needed for control decisions at that level from the sensory data stream and from the lower level control modules. The feedback processing modules at each level detect features, recognize patterns, correlate observations against expectations, and format the results to be used in the decisions and computational procedures of the task decomposition modules at that level.

Levels 0 and 1

At these levels of the hierarchy, the feedback processing modules extract and scale joint positions and force and torque data to be used by the servo and coordinate transformation computations.

In a force reflecting master-slave teleoperator system, these data are used to drive the master so as to give the operator the "feel" of the forces and tactile feedback generated by contact with objects in the environment.

Level 2

At the second level, touch and proximity data, and simple visual measurements of distance and positions of grip points are extracted from the sensory input to be used in computing trajectory end points.

Level 3

At the third level the three dimensional positions of visual features such as edges, corners, and holes are computed and combined to determine the position and orientation of surfaces and volumes of objects. Identities of objects may also need to be computed (or recognized) in order to generate the reaching and grasping commands at this level.

In teleoperator systems, visual feedback to the human operator enables the human to perform the recognition function, and to determine the spatial position and orientation of features such as surfaces, edges, and suitable grip points. In a robot, these recognition and geometric reasoning functions must be performed automatically.

Level 4

At the fourth (WORKSTATION) level, relationships between various objects need to be determined, in order to sequence simple task commands.

Level 5

At the fifth (CELL) level, the location and composition of trays of parts and tools and the length of queues of parts needs to be determined. This may be derived from sensors which read coded tags on trays, or may be inferred from sensory input from lower level sensors on the robot or in the workstation.

Level 6

At the sixth (SHOP) level, the condition of machines, tools, and the amount of inventory on hand must be determined in order to generate schedules, allocate resources, and evaluate and set priorities for production.

Level 7

At the seventh (FACILITY) level, the requirements for changes in part design, or in process plans need to be recognized in order to make engineering changes, or redesign parts or processes.

THE WORLD MODEL

The world model hierarchy, made up of M modules in Figure 2, consists of a knowledge base containing all the information currently known about the task, the parts, or the workplace, together with procedures that allow the M modules to compute a "best estimate" of the state of the external world. The M modules can thus provide the H modules with information about the external world that may not be directly measurable by sensors at the specific instant that it is needed.

The M modules are also able to compute expectations as to what the sensory data to the corresponding G modules "should" be, based on the state of the task and estimated state of the world. This allows the G modules at each level to compare expectations with observations, and to measure both the degree of correlation and the degree of difference. A strong degree of correlation means that the proper model is being matched with the incoming sensory data. It means that the observed object or situation has been correctly recognized, and that information contained in the model can be safely used for decision making even though it may not be directly observable by the sensory system.

A large degree of difference between expectations generated by the model and observations derived from sensors means that either an incorrect choice of models has been made, or the model has not been correctly transformed spatially or temporally so as to generate the proper set of expected feature relationships, or that the incoming sensory data is too noisy, or is being improperly processed and filtered. In this case, the computational problem for the task decomposition module is to decide which type of error is being encountered and what is required to remedy the discrepancy. In general, this type of problem can be solved either by a set of situation/action rules of an expert system, or a set of heuristic search procedures.

The world model counterpart in teleoperators exists entirely in the brain of the human controller.

Levels 0 and 1

At the coordinate transformation and servo level, the world model generates windows or filter functions that are used to screen and track the incoming raw data stream. It also provides kinematic and dynamic models for feedforward control terms.

Level 2

At the elemental move level, the model is able to generate expected positions and orientations of specific features of parts and tools, such as edges, corners, surfaces, holes, and slots [6].

Level 3

At the simple task level, the model contains knowledge of the geometrical size and shapes of three dimensional objects such as parts and tools and the relationships between coordinate systems based in the work space and the robot. These can be used to generate expected positions and orientations of three dimensional objects in a robot or machine tool coordinate system.

Level 4

At the workstation level, the world model contains knowledge of tray layouts including the names of parts and their approximate positions, orientations, and relationships such as on-top-of, underneath, stacked N-deep, leaning-against, etc.

Level 5

At the cell level, the model contains information about workstation task times, and is able to simulate the performance of various hypothetical task sequences.

Level 6

At the shop level, the world model contains information about machine capabilities, machinability of materials, tool life, and inventory levels and is able to simulate the performance of various cell configurations.

Level 7

At the facility control level the model contains information about machining processes, material properties, shop processing capabilities, and expected lead times for procurements which can be used to compute estimated completion times for various production plans.

STATE TABLE TASK EXECUTION

At each level of the control hierarchy, a computing module such as shown in Figure 3 can be used to execute the production rules that encode the control program at that level. The list of production rules that define the actions of each computing level make up a state-transition table. The left-hand side of the table consists of all the command, internal state, and feedback inputs that can be encountered at any tick of the state-clock. The right-hand side contains an output command (and/or a pointer to a procedure which computes an argument which becomes part of the output command) to the next lower level. It also contains a next internal state, and a report to the next higher level, or to other modules at the same level. An alternate form of the state-transition table is a state graph. The state graph is analogous to a flow chart of a procedural program for the task decomposition module [1].

Levels 0 and 1

At the lowest hierarchical levels, the left-hand side of the state-transition table consists of variables which select the type of coordinate transformation required and the type of servo computations needed.

Level 2

At the second level, the left-hand side consists of variables which define the type of trajectories to be generated. The right-hand side contains pointers to procedures that compute forces, positions, accelerations, and velocities in the appropriate coordinate systems.

Level 3

At the third level, the left-hand side consists of variables which specify the state of the environment as reported by sensors, and the right-hand side the names of appropriate elemental movements to be made for each state. Pointers to procedures are used to compute arguments and modifiers.

Levels 4 and above

At the higher levels, the state-tables may be compared to production rules in expert systems. Procedures that are invoked by these state-tables may consist of heuristic search algorithms or linear programming techniques for generating plans, schedules, etc.

The response time and cycle time requirements grow longer for the finite-state automata at the upper levels of the hierarchy. Thus, the amount of computing power needed in the execution mode to execute state-transition tables decreases at higher levels in the hierarchy. On the other hand, there is much more need for planning at the upper levels. For example, the types of control decisions required at the upper levels of the factory control system shown in Figure 2 typicaly involve planning algorithms. The hierarchical control system proposed here thus provides the required planning capability. The upper levels of the control hierarchy can use the excess execution mode computing power to operate in the planning mode.

CONCLUSIONS

The hierarchical control structure described here partitions the robot/teleoperator control problem into simple, well-defined levels with clearly specified inputs, outputs, internal states, and rules for state-transitions. The control problem is also partitioned into separate functions of task decomposition, sensory processing, and world modeling.

The system has a large number of clearly defined interfaces which makes it possible for a human operator to enter the system at a variety of different levels depending on the complexity and novelty of the task. For simple or repetative tasks the manipulator control system can be driven by the upper levels of the automatic hierarchy. For complex or unanticipated conditions, the human operator can readily enter the control hierarchy to modify the actions of the automatic system. The result is a system which has both enormous flexibility and highly autonomous capabilities.

The fully autonomous version of this hierarchical control architecture is under development at the National Bureau of Standards and has proven highly successful in the Automated Manufacturing Research Facility (AMRF). The addition of the multiple interfaces for teleoperation has yet to be fully explored.

BIBLIOGRAPHY

1. Albus, J.S., Barbera, A.J., Fitzgerald, M.L. (1982). Programming a Hierarchical Robot Control System. Proc. 12th International Symposium on Industrial Robots, Paris, France, 9-11 June 1982.

2. Albus, J.S., Barbera, A.J., Fitzgerald, M.L. (1981a). Hierarchical Control for Sensory Interactive Robots. Proc. 11th International Symposium on Industrial Robots, Tokyo, Japan, 7-9

October 1981.

3. Albus, J.S., Barbera, A.J., Fitzgerald, M.L., Nashman, M. (1981b). Sensory Interactive Robots. Proc. of 31st General Assembly, International Institution for Production Engineering Research (CIRP), Toronto, Canada, September 1981.

4. Albus, J.S., Barbera, A.J., Nagel, R.N., (1981c). Theory and Practice of Hierarchical Control. Proc. 23rd IEEE Computer Society International Conference, September 1981, 18-39.

5. Barbera, A.J., Fitzgerald, M.L., Albus, J.S. (1982). Concepts for a Real-Time Sensory-Interactive Control System Architecture. Proc. 14th Southeastern Symposium on System Theory, Blacksburg, Virginia, April 1982, 121-126.

6. Kent, E.W. "A Hierarchical, Model-Driven, Vision System for Sensory-Interactive Robotics." Compsac 82, Chicago, IL, November 11, 1982.

7. McLean, C.R. (1982). The Virtual Manufacturing Cell. Proc. 4th IFAC/IFIP Symposium on Information Control Problems in Manufacturing Technology, Gaithersburg, Maryland, 26-28 October 1982.

8. McLean, C. Mitchell, M., and Barkmeyer, E. "Information Processing and Standardization for Automated Manufacturing" Proceedings of the First International Symposium on Automated Integrated Manufacturing (AIM), April 5-6, 1983

9. Simpson, J.A., Hocken, R.J., Albus, J.S. (1982). The Automated Manufacturing Research Facility of the National Bureau of Standards. Journal of Manufacturing Systems, Society of Manufacturing Engineers 1:(1)17-31, 1982.

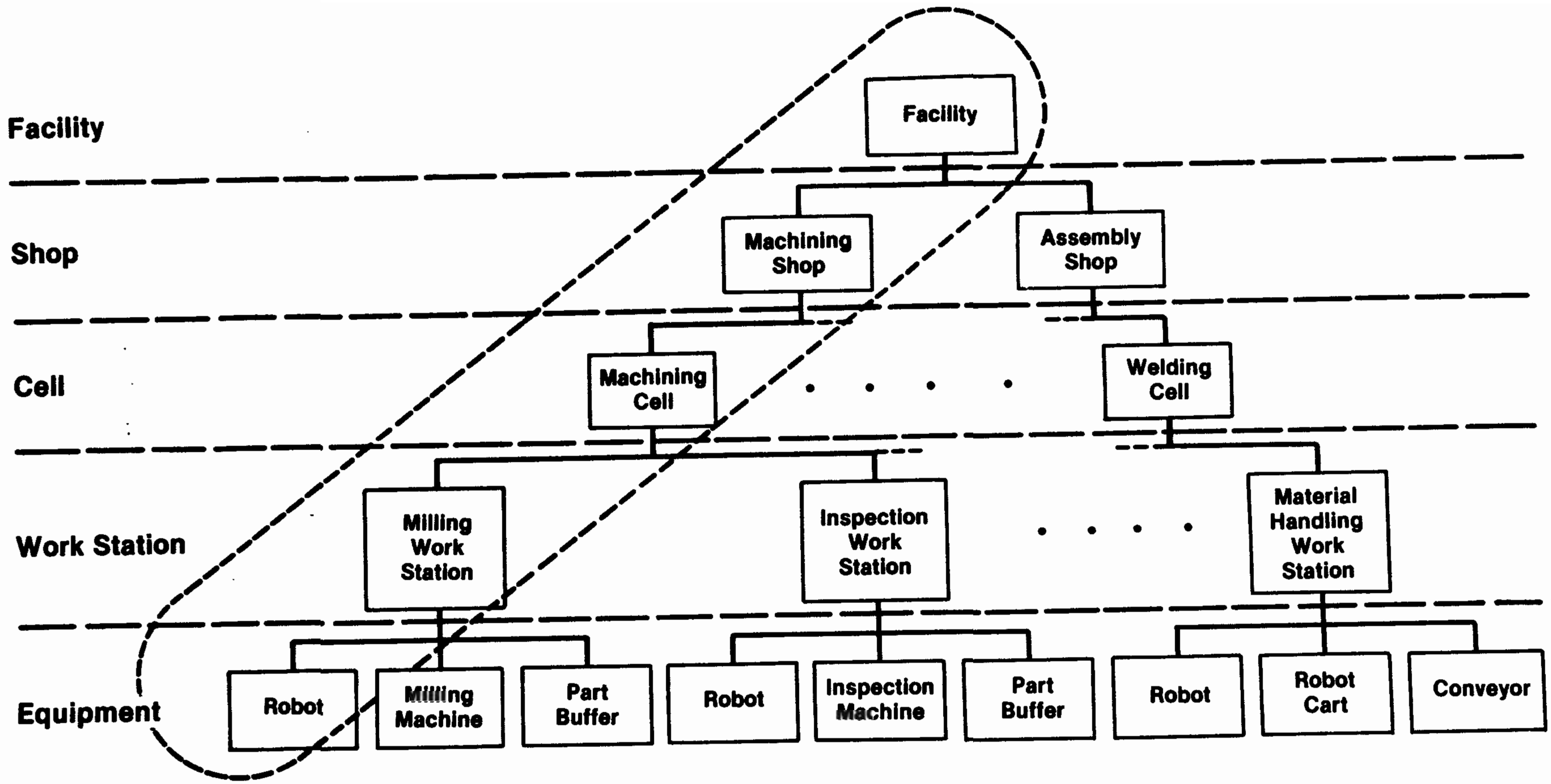

Fig. 1: A hierarchical control system architecture for integration of robots, machine tools and material transport facilities into an automatic factory. The chain of command enclosed in the dotted lines is detailed in Fig. 2.

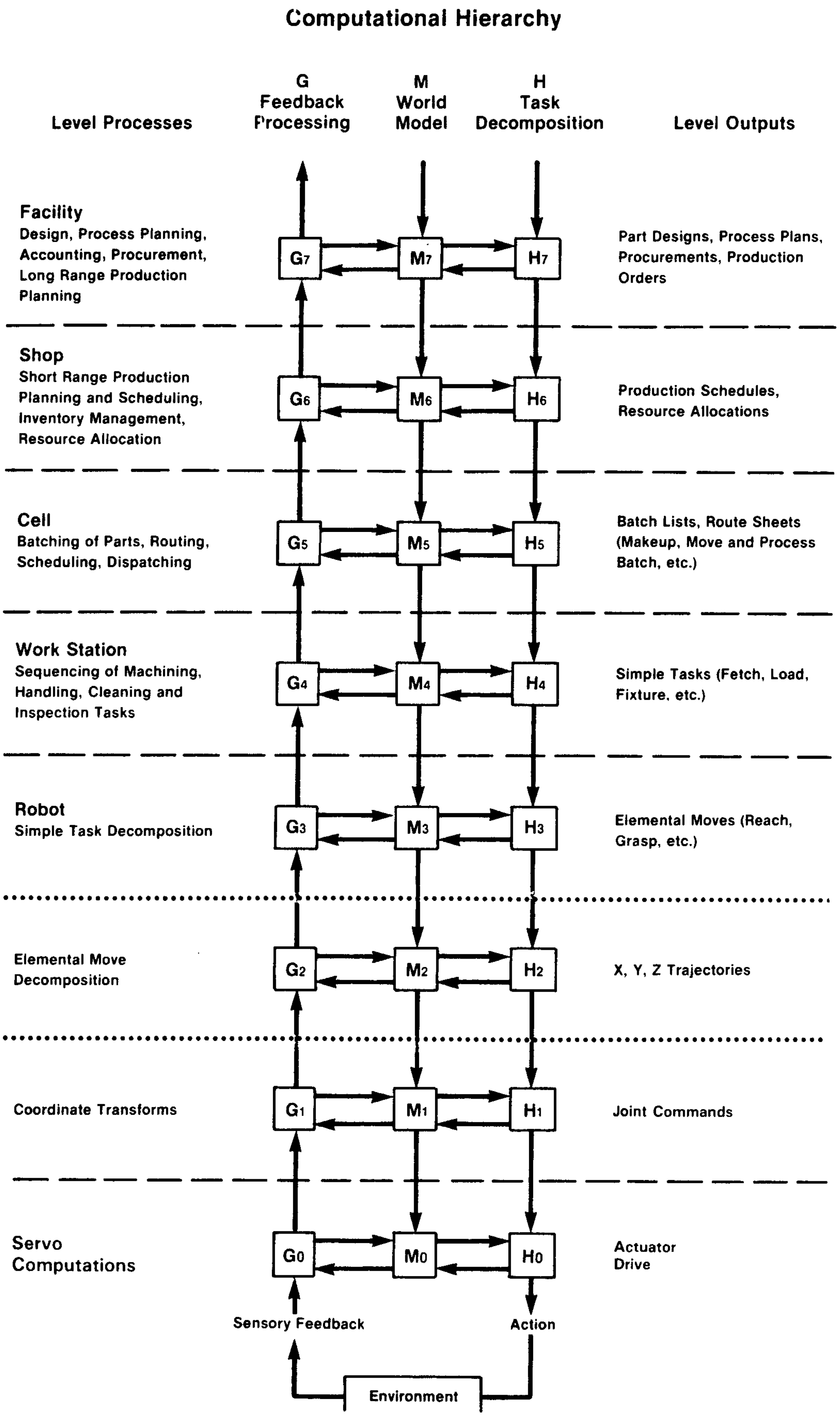

Fig. 2: The computational hierarchy for a robot in a machining work station. This hierarchy corresponds to the chain of command enclosed in dotted lines in Fig. 1.

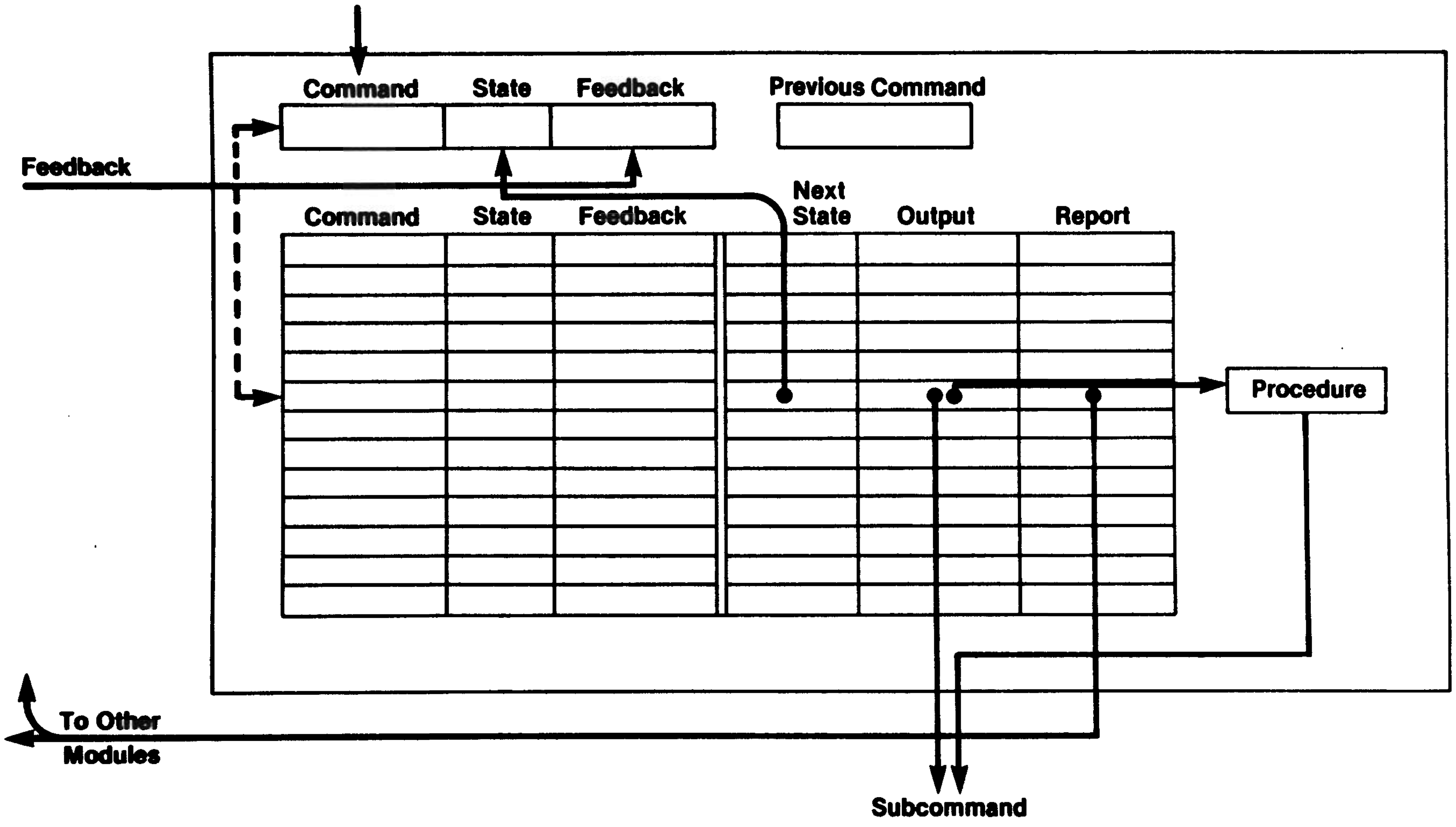

Fig. 3: A computing structure designed to execute state-transition tables.