Robotics -- Past, Present, and Future

James S. Albus
National Bureau of Standards
Washington, DC 20234

Abstract

Robotics is a systems science that attempts to integrate artificial intelligence with feedback control of mechanical devices. It draws on work in pattern recognition, scene analysis, geometrical reasoning, world modeling, language and speech understanding, planning, problem solving, goal seeking, task decomposition, manipulator control, mobility, and navigation.

Mankind's interest in mechanical contraptions that move and act under automatic control dates back at least to the ancient Greeks. The modern history of robotics began with work in the 1950's on mechanical manipulators for handling radioactive materials. In 1959, the first industrial robot was introduced into the commercial marketplace. Academic research into robotics began shortly thereafter at MIT, Case, Stanford, and SRI.

This paper will present a brief history of robotics and examine the following current research topics:

(1) Kinematics, Dynamics, and Mobility
(2) Vision, Kinesthetic, Tactile, and Acoustic Sensing and Sensory Processing
(3) Sensory-interactive Task Decomposition, Planning, and Problem Solving
(4) World Modeling
(5) Programming Techniques and Learning
(6) System Integration

Future applications will cover a broad spectrum. Robot technology for mobility, database access, and sensing will permit robots to leave the relatively structured environment of the factory and enter the dynamic and cluttered environment of the construction site, shipyard, farm, mines, undersea drilling, etc. Eventually the cost will drop and the performance will rise to the point where robots can perform useful tasks in the service industries, and even in the home.

REPRINT FOR: ADVISORY GROUP FOR AEROSPACE RESEARCH & DEVELOPMENT LECTURE SERIES NO 142 ON ARTIFICIAL INTELLIGENCE AND ROBOTICS, SEPTEMBER 1985.

A BRIEF HISTORY

Man's fascination with machines that move under their own power and with internal control is at least as old as recorded history. As early as 3000 B.C., the Egyptians are said to have built water clocks and articulated figures, some of which served as oracles. The Greeks, Ethiopians, and Chinese constructed a great variety of statues and figures that acted out sequences of motions powered by falling water or steam. Hero of Alexandria amused Greek audiences around 100 B.C. with plays in several acts performed entirely by puppets driven by weights hung on twisted cords. In the 15th and 16th centuries, with the invention of clockwork, a number of town clocks and bell towers were built throughout Europe with figures that still today act out scenes on the hour.

During the last half of the 18th century, it became popular in the courts of Europe to commission the construction of lifelike automata of animals, birds, and humans for the amusement of royalty. Most notable of those still in working condition are the automata of Pierre and Henri-Louis Jaquet-Droz. These are on display in the Musee d'Art et d'Histoire in Neuchatel Switzerland where they are operated occasionally. The Scribe, built in 1770 is an elegantly dressed figure of a child that writes with a quill pen that is dipped in ink and moves over the paper with graceful strokes. The device is controlled by an elaborate set of precision cams driven by a spring-powered clock escapement and can be mechanically programmed to produce any text. A similar automaton, the Draughtsman, has a repertoire of four drawings. A third, the Musician actually plays a miniature organ with fingers that strike the keys in the proper sequence to produce the notes. The Musician's breast rises and falls in simulated breathing, the body and head sway in rhythm with the music, and the eyes glance about in a natural way.

The fascination, awe, and sometimes fear that surround the subject of robots center on the notion of creating artificial life. The potentially threatening and uncontrollable consequences of this possibility have provided the dramatic theme of endless science fiction stories, plays, and films. One of the first and most influential works in this area was Frankenstein, published in 1817, only a year after the author Mary Shelley had visited Neuchatel where the Jaquet-Droz automata were then, as now, on display.

The actual word "robot" was not coined until a century later by the Czechoslovakian playwright Capek. Robot derives from the Czech word for "worker". In the play R.U.R (Rossum's Universal Robots), Capek introduces the notion that robots could be used in industry for reducing the human labor required to produce manufactured goods and services. The drama comes when the robots are endowed with emotions, and rebel against their human masters.

Recent science fiction literature, most notably the stories of Issac Asimov and the "Star Wars" series of movies, have portrayed robots more positively as potential friends and loyal companions of human beings.

In the real world of 1985, robots are much less the subject of melodrama and more the subject of capital investment decisions. Industrial robots first entered the marketplace in the 1961, and it was not until 1975 that Unimation, the leading robot manufacturer first made a profit [1]. Contrary to popular opinion, most robot companies are not a good investment. Although the market for robots is growing about 30% per year, the number of robot manufacturers has grown much more rapidly. Today there are literally hundreds of robot companies, many of them among the world's corporate giants. There are less than 150,000 industrial robots in the world and the world market is less than $3 Billion [2]. The bottom line is that there are too many companies in the robot business for more than a few of them to make money.

The most interesting aspect of robotics is their potential, not current, capabilities. Current robots, even in the most advanced research laboratories are surprisingly incapable of more than the most primitive manipulative and locomotive actions. Current industrial robots can manipulate heavy objects, but with the dexterity of a blind, deaf, stupid, one-armed giant wearing a steel boxing glove and with both feet nailed to the floor. Current industrial robot carts can move about on wheels on flat floors, and some research robots can even walk -- but with nothing approaching the ambulatory skill of a beetle, much less that of a human being.

A history of modern robotics research would begin at MIT, with H.A. Ernst. It would include work on robot carts and arms and robot plans at MIT under Marvin Minsky and at Stanford University under John MacCarthy. It would point out important milestones such as Richard "Lou" Paul's work on WAVE [3], Tom Sheridan's work on Supervisory Control, Dan Whitney's development of Resolved Motion Rate Control [4], and Tom Binford's development of AL. It would credit the pioneering theoretical studies of Vucobratavich [5] and Bob McGhee, the early innovative work of Poppelstone and Ambler [6] at Edinborough, and the extensive ARPA investment in the "Shakey" project at Stanford Research Institute [7]. It would come up-to-date with the Robotics Institute at Carnegie Mellon, the Manufacturing Automation programs at the National Bureau of Standards [8], and in a number of aerospace industrial laboratories as a result of funding by the U.S. Air

Force and Navy through their respective Research and Manufacturing Technology Program offices.

CURRENT RESEARCH TOPICS

1. STRUCTURES

a) Kinematics

Although there are a great variety of robots on the market with many different size, shape, and form factors, much remains to be done to improve the mechanical performance of these devices.

Perhaps the most elementary problem is that of accuracy. Most current industrial robots operate without significant sensor feedback. Welds are made, adhesives are applied, parts are picked up, put down, inserted into jigs and fixtures, and assembled by open-loop dead-reckoning. Joint angles are monitored and servoed to commanded positions, but nothing directly measures the position of the endpoint relative to the work-piece. Programs are written in terms of joint angle positions, or cartesian poses which are simple algebraic transformations of joint positions. In order to program such robots off-line, they must be able to go to commanded coordinate points. Although the repeatability of most robots is on the order of one millimeter over the working volume (and in some cases as good as 0.1 mm.), the absolute positioning accuracy is often worse than plus or minus a centimeter. Thus, in many applications it is not possible to program a robot from an external data base, and it is not possible to transfer a program taught on one robot to another.

Some modern robots have absolute accuracy error tables in their software so that systematic errors can be corrected in software, but this is only available on the more sophisticated machines. A more common engineering approach to the accuracy and repeatability problem is to make robot structures very stiff and rigid. Unfortunately, this means that industrial robots tend to be massive and ungainly. Most robots are cumbersome devices that can lift only about one twentieth of their own weight. Compare that to the human arm which can lift about ten times its own weight. The difference in the strength-to-weight ratio is a factor of two hundred.

b) Dynamics

Dynamic performance is also an area where much remains to be done. Presently available robot servo systems do not adapt to the changing inertial configuration of the robot, nor do they adapt to the variety of loads that the robot must carry. The result is that robot servo systems typically are far from optimal, and often it is difficult to find any set of satisfactory servo parameters that will make the robot stable over the full range of possible loads and configurations.

Many technical papers have addressed the dynamic equations of multijointed manipulators. The two most popular approaches are the Newton-Euler method and the Lagrangian formulation [9]. These allow the joint torques to be computed in terms of desired velocities and accelerations. Unfortunately, these equations are so complicated that real-time computation requires a great deal of computing power, and in practice, PID (Proportional, Integral, Differential) controllers are installed on each joint. On almost all industrial robots, the joints are servoed independently and forces resulting from cross-products of inertia are treated as disturbances.

In order to maximize the ratio of load-to-arm-weight, new mechanical designs for robots will use light weight materials such as carbon filament epoxies and hollow foam-filled tubular constructions. Advanced control systems with strain gauges, accelerometers, and end-point sensors will be used to control light-weight structures that flex and twist under gravity and acceleration loads.

Control algorithms for light-weight flexible arms are being investigated in several laboratories, most notably Bob Cannon's at Stanford University, but the work is very preliminary at this time. Nowhere is there a robot device approaching the overall performance of biological arms, legs, and wings. For example, the top slew velocity of a robot arm is typically around 40 inches per second, while the top velocity that can be achieved by the human arm during a task such as throwing a baseball is around 1500 inches per second. The difference in speed is a factor of nearly forty.

c) End Effectors

Much also remains to be done in robot end effectors and gripper design. Typically, robot hands consist of pinch-jaw grippers with only one degree of freedom -- open and shut. Contrast this with the human hand which has five fingers, each with four degrees of freedom. No robot hand comes close to the dexterity of the human hand.

One approach is to design interchangeable grippers and end effector tooling, but this is not without cost. Bringing sensor signals and power for control through an interchangeable interface is expensive.

Another approach is to design sophisticated adaptable grippers. There have been several designs of three fingered grippers. One at the Electrotechnical Laboratory, Tsukuba, Japan, can roll a ball between its fingers or twirl a cardboard baton, but the action is slow and awkward. A similar three-fingered gripper has been developed by Ken Salisbury [10] at M.I.T., and another is under development by Steve Jacobsen [11] at the University of Utah. The development of control algorithms for these types of grippers is in a very primitive state [12].

Other complex hands have been built, such as the one designed at the University of Rhode Island [13], which has little suction cups on the ends of the many extensible rods that conform to the surface of the object. Presumably, a pair of such devices, one in each jaw of a gripper, could grasp, and perhaps even actively reposition an object in its grasp. However, the development of control algorithms for this type of gripper has not yet been seriously addressed.

d) Mobility

Many potential robot applications require mobility. Most robots today are bolted to the floor, or to a tabletop. Small robots can reach only 20 to 50 centimeters,. while larger ones can grasp objects two or three meters away. However, many applications need robots which can maneuver over much larger distances. In construction tasks, such as assembly of large structures, ships, or buildings, it is not practical to bring the work to the robot; the robot must go to the work, sometimes over distances of a hundred meters or more.

Mobility can have considerable economic utility even in machine tool loading. Robots used to load machine tools typically spend most of their time waiting for the machine tool to perform its operations.

Today, this problem is solved by positioning a single robot between two or more machine tools so that it can be more fully utilized. This leads to crowding of the work environment and in many cases is simply not practical. In a few applications robots have been mounted on rails so that they can shuttle between several machines. This type of mobility is often too expensive and cumbersome.

Commercial robot carts of various types typically follow wires buried in the floor, or are pulled by chains like cable cars.

Presumably, robot arms could be mounted on such carts, but no one has yet marketed such a system. Outside the domain of manufacturing there are experimental mobile robots that have been designed for a number of applications.

The Jet Propulsion Lab has tested a variety of wheeled and tracked vehicles for possible use as a planetary roving vehicle [14]. Jean Vertut of the French Atomic Energy Department has built several roving vehicles for performing tasks in a nuclear radiation environment. Marc Raibert at Carnegie Mellon has developed a one-legged, hopping robot, and is now constructing a four-legged multi-gaited robot which can walk, trot, or gallop. The DARPA Strategic Computing program has funded both walking and wheeled autonomous vehicle programs. A ship-building robot should be able to maneuver inside odd-shaped compartments, climb over ribs and bulkheads, scale the side of the ship's hull, and weld seams several hundred feet in length. Similar mobility requirements exist in the construction of large buildings. Construction robots will need to be able to maneuver through the cluttered environment of a building site. In some cases wheeled vehicles will be adequate, but in many applications construction robots will need to climb stairs, work from scaffolding, or even be suspended from cables by cranes.

Future applications for mobile robots will include undersea exploration, and drilling and mining of the seabed. Eventually, mobile robots will explore the moon and planets. Needless to say these will require significant new developments in robot mobility mechanisms.

2. SENSING

A second major robotics research topic is sensors and processing techniques which enable robots to detect information about the state of the environment. This is necessary if robots are to behave in an intelligent way. Sensory guided robots will need to be able to see, feel, hear, and measure the position of objects in a number of different ways. Data from sensors must be processed, and information extracted to direct robot actions so that the robot system can successfully accomplish its task objectives in spite of uncertainties, perturbations, and unexpected conditions and events.

a) Machine Vision

Machine vision is by far the most popular sensory research topic, and also the most difficult. The current state of the art in commercial robot vision systems is almost entirely restricted to the detection and analysis of binary (black and white) silhouette images. Much of the original work in this area was done at the Stanford Research Institute [15]. Typically, a single isolated part is photographed and the image data thresholded to produce a binary connected region. A set of features is then computed on this region. For example, the centroid, the area, the principal axis, the perimeter, and the inclusion relationships of holes can be computed. In many cases these features are sufficient to recognize an object and tell the robot where it is so that it can be picked up.

The connected region analysis method has severe limitations. For example, it cannot deal with parts that are touching or overlapping; and it does not give any information as to the third dimension of depth, or to the orientation of parts relative to the focal plane of the camera.

In recent laboratory research using silhouette images, computation of the position, spacing, and orientation of features such as corners, holes, edges, and curves is performed [16]. The geometrical relationships of these features to each other can be used to characterize the image. Once this is done, these features and relationships can be compared to a model, or an ideal image of the part. If a match is detected between the features of the observed image and those of the model, then the position and orientation of the part can be computed even if it is partially hidden or obscured by touching or overlapping parts.

All binary silhouette image analysis techniques are limited to situations where parts are relatively flat and lying on a known surface. It does not work well for parts that have important three dimensional contours or are stacked in piles of unknown height. In order to deal with three dimensional relationships, some form of stereo, triangulation, or time-of-flight ranging system must be used.

Stereo imaging has been widely researched, but the results have been slow to find industrial applications. The problem is that stereo vision requires the identification of corresponding points (i.e., one must calculate which pixel in the first image is illuminated by the same point in the world as the corresponding pixel in the second image). This is not easy to determine since it typically requires some form of cross correlation, which is computationally very expensive.

Structured light is perhaps the most commonly used technique for simplifying the corresponding points problem. A simple ray, or plane of light is often projected on an object from one point, and viewed from another point some distance from the projector. In Figure 1, two vertical planes, one on either side of the camera, cast two streaks of illumination across the landscape. The apparent position of the streaks in the thresholded image gives a measure of distance to the reflecting object. The distance from the edge of the frame to each illuminated pixel is a measure of the range along the ray generated by that pixel. The shape of the observed streak gives a measure of the shape of the object. The plane of light reveals the depth profile of the environment along the intersection of the plane of light with the object.

If a camera and light projector are mounted on a robot wrist, a single horizontal plane of light can be used to compute the distance to an object, as well as the yaw angle between the surface of the object and the robot grippers. The yaw angle is proportional to the slope of the illuminated streak [17].

More stripes, or even matrices of points and lines can be used to analyze more complex curved surfaces. The problem is that the more complex the projected light pattern, the more difficult it is to identify which reflected point in the image corresponds to which projected ray or plane -- that is, the problem of corresponding points reasserts itself. In some cases this can be solved by time sequencing, and thus encoding the various components of the projected light pattern. If a two plane structured light system is combined with a binary image analysis program, it becomes possible to compute all six degrees of freedom of the object relative to the gripper. As shown in Figure 2, a pair of planes of light can measure the range, yaw, and pitch angles of a surface of an object. Binary image analysis can measure the elevation and azimuth angles of the centroid of the surface. The direction of the principal axis (or of one of the edges) can be used to compute the roll angle of the robot gripper. These measurements (range, elevation, azimuth, roll, pitch, and yaw) are the six degrees of freedom needed to control the motion of the hand of the robot relative to a surface on the object. [18].

b) Other Sensors

To be truly dexterous, robots need sensors other than vision. Typically, the scanning rate for TV cameras and the processing algorithms required to extract information from vision systems are too slow for high performance servo loops. Just to scan a single image requires about 30 milliseconds. Vison processing algorithms may take several hundred milliseconds. Thus, TV camera images can be used to acquire stationary objects, or to track moving objects at a distance, but for high performance approach and gripping operations, faster acting sensors are required. For example, force servoing may require loop bandwidths greater than 100 Hertz. This corresponds to loop time delays of less than 10 milliseconds. Typically, proximity, force, and touch sensors can easily meet these requirements.

Force sensors can be mounted either in the fingertips, or in the wrist. A number of commercial wrist force sensors are now available. These can resolve and measure the three forces and three torques at the robot wrist. The principal disadvantage of a wrist force sensor is that the weight of the hand itself is a significant factor. It is thus difficult to measure small forces and torques because they are masked by the weight of the hand.

Work is being done at a number of different laboratories on arrays of touch sensors which enable the robot to detect the shape of the object being grasped, as well as the position of the object in the hand. However, at present there seems to be limited utility in using large finely spaced arrays of touch sensors to recognize shape, particularly in a factory environment. Seldom does one program a robot to grasp an object by the edge such that the outline of the edge of a surface can be sensed by touch. The overall shape of an object is usually easier to measure by visual or other non-contact sensors before touch occurs, and surface orientation can be measured by as few as three tactile sensors. Of course, there are applications where sophisticated tactile shape discrimination is crucial to task performance, such as underwater where vision is obstructed by murky water. In a factory environment such difficulties are seldom a problem.

Proximity sensors often use infra-red light-emitting diodes in a variety of configurations. Sensors may measure distance as inversely proportional to reflected intensity. This requires some method of compensating for variations in reflectance of the object.

Once the object is within the grippers, beam breaking sensors can be used to detect the exact position of edges of the object. Other techniques for measuring proximity over small distances are eddy current detectors, and air pressure detectors, which sense the back pressure from an air jet projected onto the surface of an object.

Acoustic sensors that measure the time of flight of an ultrasonic pulse can be used for detecting the distance to objects up to 15 feet away. The most popular commercially available acoustic ranging sensors saturate inside a few inches, so they are not useful for the terminal phase of gripping operations. However, such sensors are ideal for measuring the height of objects in a stack, or for detecting the presence of obstacles or intruders in the robot work area. Thus, they can be used for safety sensors.

(3) CONTROL

The fundamental technical problem in robotics is goal-seeking, i.e., the generation and control of behavior that is successful in accomplishing a task or goal. The purpose of a robot control system is to accomplish commanded tasks. The purpose of sensors and sensory processing is to detect the state of the environment (i.e., the position, orientation, and spatial-temporal relationships of objects in the world) so that control signals appropriate to the task goal can be generated. Among other things, this implies that the processing of sensory data must be done in the context of the control problem. Because of this tight interaction between sensing and control, we will constantly intermix sensory processing in our discussion of the control system.

Most industrial robots today have no sensors, and in many cases their control system is nothing more than a memory which can store a series of points and a sequencer which can step the robot through the series of recorded points.

The situation is more complicated if a robot has sensors. Robots with sensors require as a minimum, the ability to modify the sequence of programmed points in response to sensor data. To achieve full real-time sensory-interactive behavior, a robot must have the ability to change the actual positions of the recorded points in real time. Precomputed trajectories will not work. Trajectories must be recomputed on the fly.

Really sophisticated robot control systems need to be able to accept feedback data at a variety of levels of abstraction and have control loops with a variety of loop delays and predictive intervals. Force and velocity data used in servo loops for high speed or high precision motions can be processed and introduced into the control system with delays of no more than a few milliseconds. Vision data for detecting the position and orientation of objects to be approached typically requires hundreds of milliseconds. Processing sensory data to recognize complete objects or interpret complicated relationships between groups of objects can take seconds. Control systems that are properly organized in a hierarchical fashion so that they can accommodate a variety of sensory delays of this type are not available on any commercial robot.

Figure 3 illustrates the basic concepts of a hierarchical control system. On the left is an organizational hierarchy wherein computing modules are arranged in layers. The basic structure of the organizational hierarchy is a tree.

At the top of the hierarchy is a single high-level computing module. Here at the highest level, the most global goals are decided upon and long-range strategy is formulated. Feedback to this level is integrated over an extensive time period and is evaluated against long-range objectives. Decisions made at this highest level commit the entire hierarchical structure to a unified and coordinated course of action designed to achieve the selected goal. At each lower level, computing modules decompose their input commands in the context of feedback information generated from other modules at the same or lower levels, or from the external environment. Sequences of subcommands are then issued to sets of subordinates at the next lower level. This decomposition process is repeated at each successively lower hierarchical level, until at the bottom of the hierarchy there is generated a set of coordinated sequences of primitive actions which drive individual actuators such as motors, or hydraulic pistons, in generating motions and forces in mechanical members.

Each chain-of-command in the organizational hierarchy consists of a computational hierarchy of the form shown in the center of Figure 3. This computational hierarchy contains three parallel hierarchies: (1) a task decomposition hierarchy which decomposes high-level tasks into low level actions, (2) a sensory processing hierarchy which processes sensory data and extracts the information needed by the task decomposition modules at each level and (3) a world model hierarchy which generates expectations of sensor data at each level based on the subtask currently being executed at that level. Each level of the task decomposition hierarchy consists of a processing unit which contains a set of procedures, functions, or rules for decomposing higher level input commands into a string of lower level output commands in the context of feedback information from the sensory processing hierarchy. At every time increment, each H module in the task decomposition hierarchy samples its inputs (command and planning inputs from the next higher level and feedback from the world model module at the same level) and computes an appropriate output.

a) Task Decomposition

In a robot control system, servo computations are made at the bottom (or zeroth) level of the task decomposition hierarchy.

At level one, coordinate transformations are done, and motion commands are scaled to hardware limits on velocity and force.

At level two, elemental moves (such as <REACH TO (A)>, <LIFT>, <ORIENT ON (B)>, <MOVE TO (X)>, <RELEASE>, etc.) are decomposed into force and velocity trajectories in a convenient coordinate system. Ideally, the control system will allow a coordinate frame to be defined either in the robot's work space, in the part, or in the robot's gripper.

At level three, simple tasks (such as <FETCH (A)>, <MATE (B) TO (A)>, <LOAD TOOL (C) WITH PART (D)>, etc.) are decomposed into the set of elemental moves which can be interpreted by the second level. Input commands to the third level are in terms of objects: object positions, orientations, and velocities, and forces and torques between objects.

b) Sensory 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.

At the zeroth level of the hierarchy, sensory processing modules filter and scale joint position, force, and torque data to be used by the joint servos.

At the first level, sensory processing modules transform sensor data into the proper coordinate frame for servoing the robot hand in position, velocity, and force.

At the second level, data variables representing robot position, velocity, and force relative to goal points and trajectories are extracted from the sensory data stream.

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) at this level in order to access information from a world model knowledge base.

In general, sensory information at the higher levels is more abstract and requires the integration of data over longer time intervals. However, behavioral decisions at the higher levels need to be made less frequently, and therefore the greater amount of sensory processing required can be tolerated.

A number of other organizational structures have been proposed for robot control systems. The hierarchical approach has an advantage over other methods of robot control in that it allows the control system to be partitioned in a way that maps directly onto the task decomposition hierarchy.

There is, of course, nothing new about the concept of hierarchical control. It was the basic command and control structure used in the Roman Empire. It is still used today by military organizations, governments, and business corporations.

It should be noted that in the control hierarchy described here, as well as those which have proven effective in military, government, and corporate applications, many types of information such as sensory, modelling, and status variables, (but not commands) flow back and forth across the hierarchy at the same level, even between control modules at the same level. Only control commands flow strictly according to a hierarchical tree. All other types of information are typically available to all members of a given level.

(4) WORLD MODEL

The representation of knowledge about the world in an internal model is absolutely crucial to both the processing of sensory data and the decomposition of tasks and goals. The world model hierarchy shown in the middle of Figure 3 contains prior knowledge about the robot's work environment. The data in the world model may be learned (i.e., entered by storing feature parameters during a training session using a sample part), or it may be generated from a Computer Aided Design (CAD) data base which contains a geometrical representation of expected parts. In either case, the world model hierarchy contains algorithms which can compute information as to the expected shape, dimensions, and surface features of parts and tools, and may even compute their expected position and orientation at various moments in the task history. This information assists the sensory processing modules in selecting processing algorithms appropriate to the expected incoming sensory data, and in correlating observations against expectations. The sensory processing system can thereby detect the absence of expected events and measure deviations between what is observed and what is expected.

a) A Hierarchy of Models

At the coordinate transformation and servo level, the model generates windows or filter functions that are used to screen and track the incoming raw data stream.

At the elemental move level, the model generates expected positions and orientations of specific features of parts and tools, such as edges, corners, surfaces, holes, and slots. The vision processing modules attempt to fit these models to incoming visual data. Differences between the predictions and the observations are reported back to the model, and the fitted ideal features are passed on to the next higher level as the best guess of the actual position of the features in the environment. An example of this is the two dimensional model matching work of Bolles and Cain [19].

At the simple task level, the model contains knowledge of the geometrical shapes of surfaces and volumes of three dimensional objects such as parts and tools. The vision system attempts to fit the set of detected features to these surfaces and volumes. Differences between the observations and the predictions are reported back to the model, and the shifted prediction is passed on to the next higher level as the best guess as to the position and orientation of solid objects in the environment.

b) Observations and Predictions

Differences between predictions and observations are measured by the sensory processing module at each level. These differences are fed back to revise the world model. New predictions generated by the revised model are then sent to the sensory processing module. The resulting interaction between sensory processing and world modeling is a looping, or relaxation process, which tends to pull the expectations into correspondence with observations. In the case of time dependent data, such as speech or music, this matching process takes on the character of a phase-lock loop, or synchronous detection process.

Errors between observations and predictions at each level may also be used by the task decomposition hierarchy to modify actions and bring sensory observations into correspondence with world model expectations.

In either case, once a match is achieved between observation and expectation, recognition can be said to have been achieved. The model can then be used as the best guess of the state of the external world, and the task decomposition hierarchy can act on information contained in the model which cannot be obtained from direct observation. For example, a robot control system may use model data to reach behind an object and grasp a surface which the model predicts is there, but which is currently hidden from view. In many cases, the model can provide much more precise and more noise free data about an object than can be obtained from direct measurements, because many different sensory measurements can be fit to the model by statistical regression techniques. Individual sensory measurements are often are made under less than optimal conditions with relatively low resolution and sometimes noisy instruments. Once it has been determined that a particular model fits the object being observed, the model can therefore provide more complete and reliable control data than direct measurement of the object itself.

A large degree of difference between expectations generated by the model and observations derived from sensors means that a recognition has not yet been made, or that there is no prior knowledge or experience which applies to the current state of the environment, or that the appropriate model has not yet been correctly transformed spatially or temporally 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 any of these cases, the computational problem is to decide which type of error is being encountered and what is required to remedy the discrepancy. In many cases, 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.

It is possible to use the topology of an object to define a parcellation of space. In other words, there are regions in space around the object in which a particular aspect of the object is visible. The boundaries to these regions are defined by the points along which features just come into view, or just sink below the horizon. Within these regions the relationship between features changes smoothly with motion of the observer and can be described parametrically. The topographical relationships between these regions can be described by a graph structure which defines the entire parcellation of space around the object [20]. Since this graph is an invariant property of the object itself, it may be computed off-line and stored in the data base of the world model.

(5) PROGRAMMING METHODS

Techniques for developing robot software must be vastly improved. Programming-by-teaching is impractical for small lot production, especially for complex tasks where sensory interaction is involved.

Shop floor personnel unskilled in computers must be able to instruct robots in what to do and what to look for in making sensory decisions. The development of compilers and interpreters and other software development tools, as well as techniques for making use of knowledge of the environment derived from a number of different sensors and CAD data-bases are research topics that will occupy the attention of robot systems software designers for at least the next two decades.

It is not clear just yet what the characteristics of good robot programming methods will be. However, top-down structured programming techniques will surely be necessary. The real-time demands of sensory-interactive goal-directed behavior imply that timing and synchronization will be a primary concern. If the control system is hierarchically structured as suggested in Section (3), there will need to be a separate programming language, or at least a separate subset of the programming language, for each level of the hierarchy. The command verbs are different at the various hierarchical levels, the type of decisions that need to be made are level dependent, and the procedures executed by the computing modules are unique to each level. It may be useful to have a variety of programming and debugging tools at each level of the hierarchy.

Yet, the various levels have much in common. Each level performs a task decomposition function, hence much of the control system and the software which runs in it will tend to have the same logical structure.

If the symbolic commands generated at each level of the task decomposition hierarchy are represented as vectors, or points, in a multidimensional "state-space", and if these points are plotted against time, the behavioral trajectories shown on the right of Figure 3 result. The lowest level trajectories of the behavioral hierarchy correspond to observable output behavior. All the higher level trajectories represent the deep structure of the control programs. This implies that state-trajectories generated by a hierarchical robot control systems define a deep structure of behavior analogous to Chomsky's notion of the deep structure of language [20]. The study of state-space trajectories which form the deep structure of robot behavior may some day provide the mathematical and computational tools for simulating and modeling the neuronal state trajectories in the brain which generate human behavior, including natural language [21].

The programming languages at each level may be procedural. There exist a large number of procedural robot programming languages such as VAL, AL, RAIL, RAPT, MCL, AML and others [22].

Alternatively, robot programs at each level can be represented as state graphs, or as state transition tables [23] (Barbera and Fitzgerald, 1982). State transition tables are a particular form of production rules such as are used in expert systems. Each line in the table corresponds to an IF/THEN rule. IF (the command is such, and the state is so, and the feedback conditions are thus) / THEN (the output is whatever is stored on the right hand side of the table, and the system steps to the next state). The addition of each node or edge to the state-graph, and the corresponding lines added to the state transition table is the equivalent of the addition of a new chunk of knowledge about how to deal with a specific control situation at a particular point in a problem domain at a unique time in the task execution. This approach thus bridges the gap between servomechanisms and finite state automata at the lower levels, and expert system technologies at the upper levels.

Research is being done on methods of generating robot programs by simply drawing state graphs on a CRT screen, and using interactive graphics to label states and to describe commands and task decompositions. A state graph has all the properties of a flow chart, which makes it easy to construct given the task requirements, and to read once it is constructed. The formal properties of state graphs make it feasible to automatically translate them into state-transition tables once the state graphs have been constructed at each level. It is possible to write compilers which translate the state graph flow charts directly into executable code. The RCS robot control system developed at the National Bureau of Standards has the convenience and debugging advantages of an interpreted language, but the execution efficiency of compiled code.

(6) SYSTEM INTEGRATION

The sixth major problem area is the integration of robots into factory control systems so that many robots, machine tools, inspection devices, and materials storage, retrieval, and transportation systems can all be interconnected to function as a unified system.

The computing architecture shown in Figure 4 is being implemented in an Automated Manufacturing Research Facility at the National Bureau of Standards. It is intended as a generic system that can be applied to a wide variety of automatic manufacturing facilities. At the lowest (equipment) level in this hierarchy are the individual robots, N/C machining centers, smart sensors, robot carts, conveyors, and automatic storage systems, each of which may have its own internal hierarchical control system. Input to the equipment level in Figure 4 corresponds to input to the object (third) level of the robot hierarchy in Figure 3. These equipment level machines are organized into work stations under the control of a work station control unit. Several work station control units are organized under, and receive input commands from a cell control unit. Several cell control units may be organized under and receive input commands from a shop control unit. At the top there is a facility control level which generates the product design, produces the manufacturing process plans, and makes the high level management decisions.

a) Data Bases

The right side of Figure 4 shows a data base which contains the part programs for the machine tools, the part handling programs for the robots, the materials requirements, dimensions, and tolerances derived from the part design data base, and the algorithms and process plans required for routing, scheduling, tooling, and fixturing. This data is generated by a Computer-Aided-Design (CAD) system and a Computer-Aided-Process- Planning (CAPP) system. This data base is hierarchically structured so that the information required at the different hierarchical levels is readily available when needed.

On the left, a second data base contains the current status of the factory. Each part in process in the factory has a file in this data base which contains information as to the position and orientation of that part, its stage of completion, the batch of parts it is with, and what quality control information is known. This data base is also hierarchically structured. At the equipment level, the position of each part is referenced to a particular tray or table top. At the work station level, the position of each part refers to which tray it is in. At the cell level, position refers to which work station the part is in. The feedback processors on the left scan each level of the data base and extract the information of interest to the next higher level. A management information system makes it possible for a human to query this data base at any level and determine the status of any part or job in the shop. It can also set or alter priorities on various jobs.

b) Interfaces

Interfaces between the many various computing modules and data bases are defined in a standardized way, so that a large number of robot, machine tools, sensors, and control computers can be connected together in integrated systems. For example, a typical workstation in the AMRF consists of a robot, a machine tool, a work tray buffer, and several tools and sensors that the robot can manipulate. Trays of parts and tools are delivered to the workstation by a robot cart.

The workstation controller is given commands consisting of lists of operations to be performed on the parts in the trays. It is the task of the workstation controller to generate a sequence of simple task commands to the robot, the machine tool, and any other systems under its control so that the set of operations specified by its input command list are carried out in an efficient sequence. For example, the workstation controller may generate a sequence of simple task commands to the robot to setup 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 planning horizon for the workstation may vary from several hours up to about a day, depending on the complexity and number of parts that are being processed.

Feedback to the workstation consists of positions of parts and relationships between various objects in order to sequence the simple task commands.

The workstation world model contains knowledge of expected tray layouts including the names of parts and their expected positions, orientations, and relationships.

The Cell control level of the hierarchy is responsible for managing the production of a batch of parts within a particular group technology part family. The task of the cell is to group parts in trays and route the trays from one workstation to another. The cell 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 has planning and scheduling capabilities to analyze the process plans for each part, to compute the tooling and fixturing requirements, and to produce the machining time estimates for each operation. It uses these capabilities to optimize the makeup of trays and their routing from workstation to workstation. The planning horizon for the cell depends on the size and complexity of the batch of parts in process, but may be on the order of a week.

Feedback to the cell indicates the location and composition of trays of parts and tools and the status of activity in the workstation. This information may be derived from sensors which read coded tags on trays, or may be inferred from processed sensory input from sensors on the robot or in the workstation.

The cell world model contains information about workstation task times, and is able to predict the expected performance of various hypothetical task sequences.

The shop control level in the AMRF hierarchy is not yet implemented. In the future it will perform long-term production planning and scheduling. It also manages inventory and places orders for parts, materials, and tools. The shop control planning and scheduling functions will be used to determine the material resources requirements for each cell. The shop then can dynamically allocate machines and workstations to the cells as necessary to meet the production schedule.

Feedback to the shop level of control will indicate the condition of machines, tools, the completion of orders, the consumption of goods, and the amount of inventory on hand.

The shop world model will contain information about machine capabilities, expected tool life, and inventory levels. It will be able to predict the performance of various cell configurations, and predict shortages of parts or materials in time to initiate reordering procedures.

The topmost level is facility control. It is at this level that engineering design is performed and the process plans are generated for maufacturing each part, and assembling each system. 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 part of a real-time control system. However, in a hierarchical control system, time horizons increase exponentially at each higher level. Using this concept, then, facility control activities can be integrated into the real-time control hierarchy of the total manufacturing system.

Feedback to the facility level consists of requirements for engineering changes in part design, or modifications of process plans.

The facility world model contains information about machining processes, material properties, shop processing capabilities, and expected lead times for procurements.

c) Interface Data Formats

One approach to the interface problem is to simply define the data elements (commands, feedback variables, status variables, sensory data parameters, etc.) which need to flow between computing modules.

These data elements can then be stored under agreed-upon names and in agreed-upon formats in the status data base. The status data base then becomes the interface between all the computing modules. At each increment of the state clock, each computing module reads its input variables from the status data base. It then performs its required computations and, before the end of the state clock period, writes its output back into the status data base. The status data base thus becomes the interface. An agreed upon format and protocol for the status data base then can become an interface standard.

This is analogous to the Initial Graphics Exchange Specification (IGES). IGES is a standard data format used as the exchange medium between diverse graphics systems [24].

The hierarchical levels described in this section correspond to well defined levels of task decomposition in the real world of manufacturing, particularly in machine shop environment. The data variables that flow between computing modules at each level correspond to physical parameters that are intrinsic to the operations being performed at those levels. There is therefore good reason to believe that it will be possible for manufacturers and users of automated manufacturing systems to agree upon a particular set of variables to be exchanged, and a particular format for exchanging this information between computing modules. If so, then such a structure as is described here may form the basis for interface standards in the factory of the future.

(7) CONCLUSION

For the most part, the six technical problem areas described above encompass profound scientific issues and engineering problems which will require much more research and development. It may be possible to improve robot mechanical accuracy and servo performance with little more than careful engineering. However, much more research and development will be required before robot mobility and dexterity can be substantially improved; the sensor, control, internal modeling, software generation, and systems interface issues represent fundamental reseach problems. Much remains to be done in sensor technology to improve the performance, reliability, and cost effectiveness of all types of sensory transducers. Even more remains to be done in improving the speed and sophistication of sensory processing algorithms and special purpose hardware for recognizing features and analyzing patterns both in space and time. The computing power that is required for high speed processing of visual and acoustic patterns may require new types of computer architectures. Sensory interactive control systems that can respond to various kinds of sensory data at many different levels of abstraction are still very much in the research phase. Current commercial robot control systems do not even allow real-time servoing of six-axis coordinated motions in response to sensory data. None have convenient interfaces by which sensory data of many different kinds can be introduced into the servo loops on a millisecond time scale for true real-time sensory interaction. None of the commercial robot control systems can interface directly with CAD data bases or computer graphics models of the environment and workpieces. Finally, current programming techniques are time consuming and not capable of dealing with internal knowledge or sophisticated sensory interactions.

These very complex problems will require many years of research effort. Until they are solved, robot capabilities will be limited and robot applications will continue to be relatively simple.

The problems listed herein are amenable to solution. It is only a matter of time and expenditure of resources before sensors and control systems are developed that can produce dexterous, graceful, skilled behavior in robots. Eventually, robots will be able to store and recall knowledge about the world that will enable them to behave intelligently and even to show a measure of insight regarding the spatial and temporal relationships inherent in the workplace. High order languages, computer-aided instruction, and sophisticated control systems will eventually make it possible to instruct robots using graphics generated pictures together with natural language vocabulary and syntax much as one might use in talking to a skilled worker.

As these problems are solved, robots will make ever increasing contributions to productivity improvement in manufacturing, construction, and service industries. By the end of the century, mobile robots are likely to be routinely used for work on the seabed, in outerspace, and personal robots will perform useful duties in the home.

REFERENCES

1. Engelberger, Joseph, Robotics in Practice, Kogan Page Limited in association with Avebury Publishing Co. 1980.

2. Yonemoto, K., Robotization and its future outlook in Japan, Proc. of 14th Int'l Symposium on Industrial Robots, Oct. 1984.

3. Paul, R.P., "WAVE: A model-based language for manipulator control," Industrial Robot, March, 1977.

4. Whitney, D.E., "The Mathematics of Coordinated Control of Prostheses and Manipulators," J.Dynamic Systems, Measurement, Control, Dec., 1972.

5. Vucobratavich, M., and D. Stokic, "Scientific Fundamentals of Robotics, VOl. 1, Dynamics of Manipulation Robots and Vol. 2, Control of Manipulation Robots," Springer-Verlag, New York, 1982.

6. Popplestone, R.J., A.P. Ambler, I. Bellos, "RAPT, A Language for Describing Assemblies," Industrial Robot" 5,3, 1978, pp.131-137.

7. Nilsson, N.J., and B. Raphael, "Preliminary Design of an Intelligent Robot," Computer and Information Sciences, Academic Press, New York, 1967.

8. Simpson, J.A., R. Hocken, J. Albus,"The Automated Manufacturing Research Facility of the National Bureau of Standards," J. Of Manufacturing Systems, Vol. 1, (1) 17-31, 1982.

9. Brady. M., J. Hollerbach, T. Johnson, T. Losano-Perez, M. Mason (Eds.), "Robot Motion: Planning and Control," MIT Press, Cambridge, MA, 1982.

10. Salisbury, J.K.,, and J.J. Craig, "Articulated Hands: Force Control and Kinematic Issues", Int'l J. of Robotics Research, Vol. 1, No. 1, 1982.

11. Jacobson, S.C., J.E. Wood, D.F. Knutti, K.B. Biggers, "The Utah/MIT Dextrous Hand: Work in progress," Robotics Research, MIT Press, 601-653, Cambridge, MA, 1982.

12. Kriegman, D.J., D.M. Siegel, S. Narasimhan, J.M. Hollerbach, G.E. Gerpheide, "Computational Architecture for the Utah/MIT Hand," Proc. 1985 IEEE Int'l Conf. on Robotics and Automation, 918-924, March 25-28, 1985.

13. Birk, J., R. Kelly, T. Barron, J. Crouch, D. Duncan, J. Hall,

F. Kolanko, A. Mak, H. Martins, R. Mehta, A. Spirito, R. Tella, A. Vinci, "General Methods to Enable Robots with Vision to Acquire, Orient and Transport Workpieces," 6th Report, Univ. of Rhode, Island, August, 1980.

14. Miller, J.A., "Autonomous Guidance and Control of a Roving Robot," Proc. of the 5th Int'l Joint Conference on Artificial Intelligence, Cambridge, MA, August, 1977.

15. Rosen, C., D. Nitzan, G. Agin, G. Andeen, J. Berger, J. Eckerle, G. Gleason, J. Hill, J. Kremers, B. Meyer, W. Park, A. Sword, "Exploratory Research in Advanced Automation," Second Report, Stanford Res. Institute, Menlo Park, CA, 1974.

16. Bolles, R.C. and R. A. Cain, "Recognizing and Locating Partially Visible Objects: The Local-Feature-Focus Method," The Int'l J. of Robotics Research, Vol. 1, No. 3, 1982.

17. VanderBrug, G. J., J.S. Albus, E. Barkmeyer, "A Vision System for Real Time Control of Robots," Proc. 9th Int'l Symposium on Industrial Robots, pp. 213-233, March 13-15, 1979, Washington, DC.

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

19. Koenderink, J.J. and A.J. van Doorn, "The Internal Representation of Solid Shape with Respect to Vision," Biol. Cybernetics, 32, 211-216, 1979.

20. Chompsky, N., "Three Models for the Description of Language," IREE Trans. Information Theory, Vol. 1T2: 113-124, 1956.

21. Albus, J.S., Brains, Behavior and Robotics, BYTE/McGraw-Hill, 1981.

22. Taylor, R.H., P.D. Summers and J.M. Meyer, "AML: A Manufacturing Language", The Int'l J. of Robotics Research, Vol. 1, (3), 1982. (Bibliogrpahy references papers on other robot programming languages).

23. Albus, J.S., A. J. Barbera, and M.L. Fitzgerald, "Programming a Hierarchical Control System," Proceedings, 12th Int'l Symposium on Industrial Robots, Paris, France, 1982.

24. Smith, B.M., K.M. Brauner, P.R. Kennicott, M. Liewald, J. Wellington, Initial Graphics Exchange Specification (IGES), Version 2.0, NBSIR 82-2631 (AF), Nat'l Bur. of Standards, 1983.

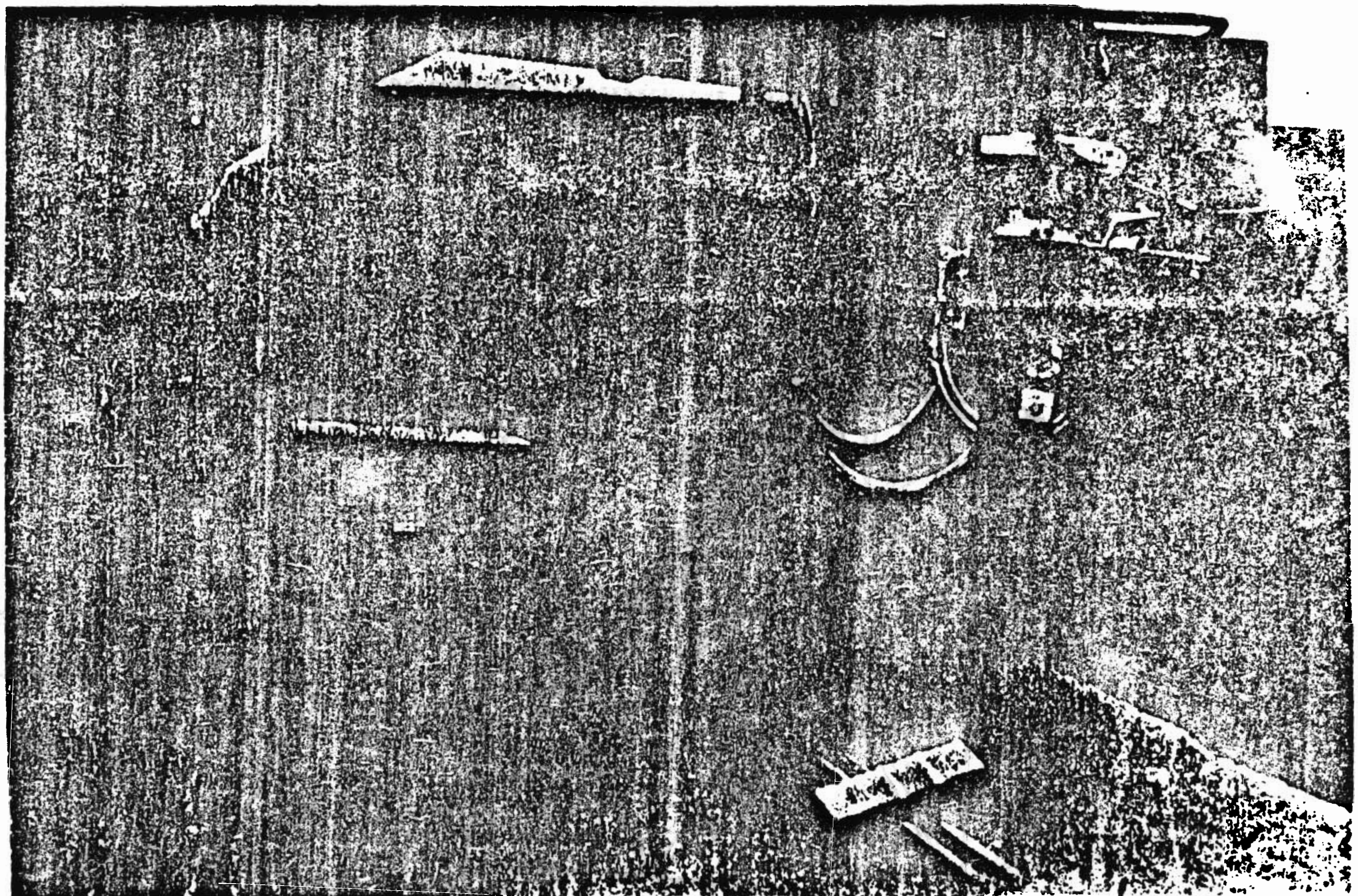

Figure 2. A two-plane structured light system combined with binary image analysis.

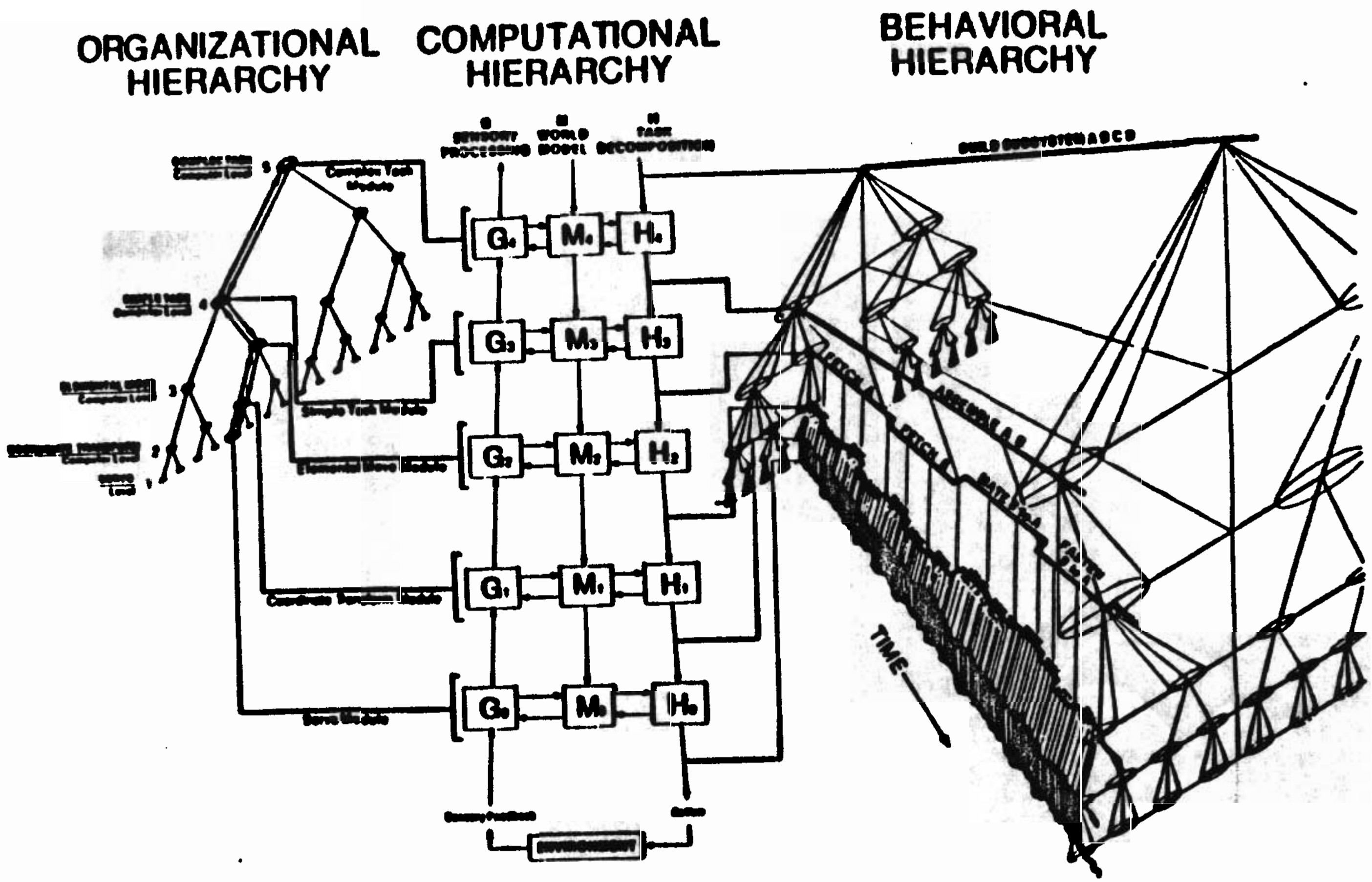

Figure 3. Relationships in hierarchical structures.

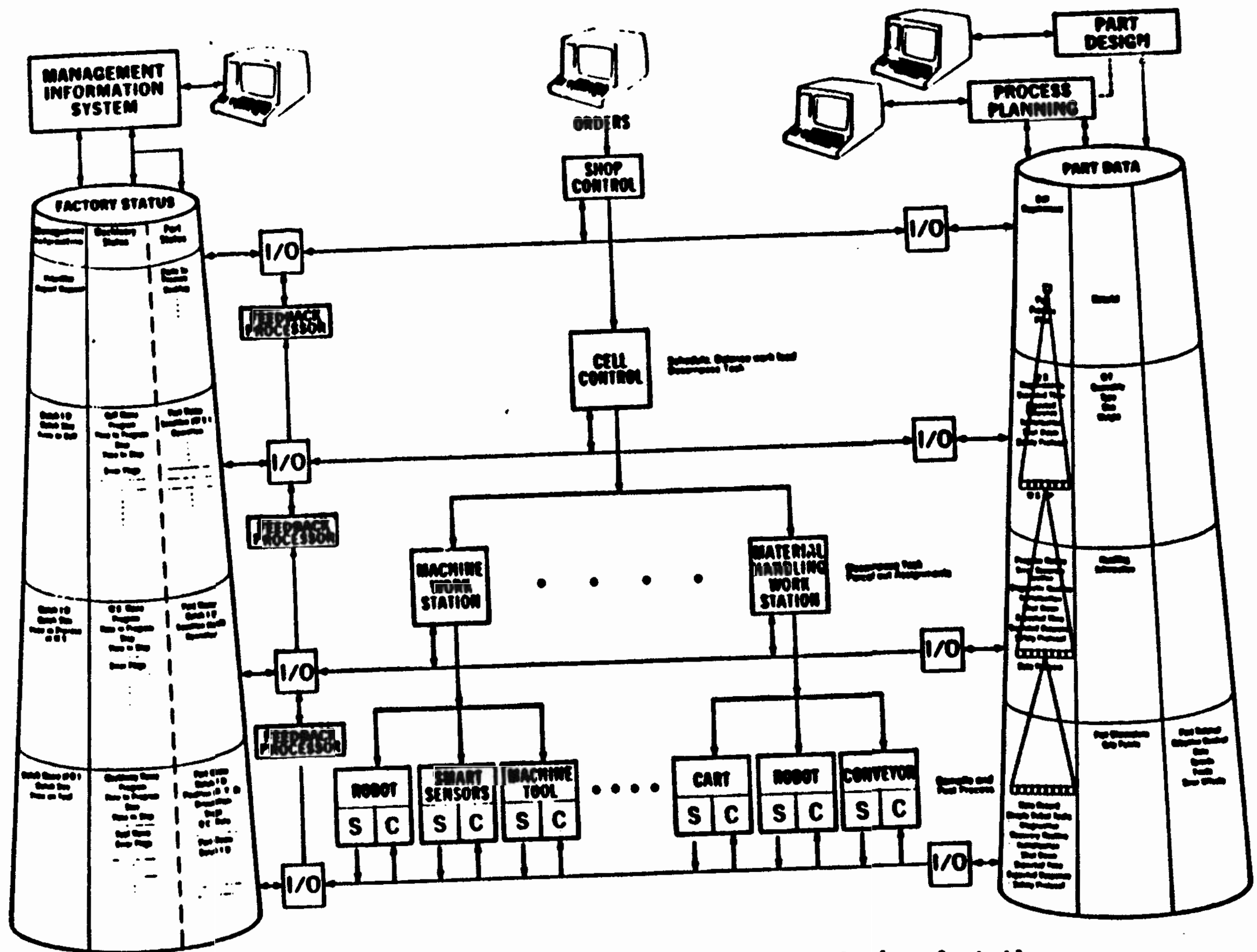

Figure 4. A hierarchical control structure being designed at the National Bureau of Standards for controlling an automatic machine shop.