

Hierarchical Control of Intelligent Machines Applied to Space Station Telerobots

J.S. Albus, R. Lumia, and H. McCain
National Bureau of Standards
Gaithersburg, MD 20899

188077 57-63
118
NG 315720

ABSTRACT

A hierarchical architecture is described which supports space station telerobots in a variety of modes. The system is divided into three hierarchies: task decomposition, world model, and sensory processing. Goals at each level of the task decomposition hierarchy are divided both spatially and temporally into simpler commands for the next lower level. This decomposition is repeated until, at the lowest level, the drive signals to the robot actuators are generated. To accomplish its goals, task decomposition modules must often use information stored in the world model. The purpose of the sensory system is to update the world model as rapidly as possible to keep the model in registration with the physical world. This paper describes the architecture of the entire control system hierarchy and how it can be applied to space telerobot applications. *Are these correct?*

1. INTRODUCTION

One of the major directions on which the robot research community has concentrated its efforts is concerned with planning and controlling motion. Given a specific task, a motion plan must be calculated which meets the task requirements. Then, the plan must be executed; there must be sufficient control for the robot to adequately effect the desired motion.

Trajectories are often planned as straight lines in Cartesian space (1). Whitney (2,3) developed the resolved motion rate control method for Cartesian straight line motions. Paul (4,5,6) used homogeneous coordinate transformations to describe a trajectory as a function of time, and Taylor (7) used coordinated joint control over small segments to keep the trajectory within a specified deviation of the desired straight line trajectory.

While the research described above employs a "kinematic" approach to robot control, another direction of research takes the manipulator "dynamics" into account in the description of robot motion. The dynamic equations of motion are described either by the Lagrangian formulation (8) or by the Newton-Euler equations (9). Algorithms and computer architectures have been suggested which promise real-time dynamic robot control (10,11).

Another aspect of motion control is concerned with the variables being controlled. The research described to this point was concerned primarily with position control. The robot moved from an initial position to a goal position. While this is perhaps the most common mode, there are many applications for robots which suggest that other variables should be controlled. For example, force control would be desired for assembly operations. Raibert and Craig (12) suggest a method for hybrid position/force control of manipulators.

These examples point to the more general problem of sensory processing. For a great deal of robot motion research, sensory processing has been limited to joint positions, velocities, and accelerations. However, other sensors are often required to accomplish tasks. The control community has concentrated on the control aspects of the robot and as a result, little emphasis has been placed on sophisticated sensory processing.

Machine vision, an offshoot of image processing research, has recently been associated with advanced robot applications. One of the most interesting directions in this research area is concerned with sensor controlled robots. Operating with the constraints imposed by real-time robot control, early methods used structured light and binary images (13,14,15,16). These approaches, though developed at different institutions, shared many concepts. One of the important subsequent research efforts went toward the development of model-based image processing. Bolles and Cain (17) used models of objects to guide the algorithms in a hypothesis/verification scheme known as the local feature focus method. The concept has recently been extended from two dimensional (i.e. nearly flat) objects to three dimensional objects (18). Although the approaches described here have led to a better understanding of real-time vision processing, the systems lacked a sophisticated interconnection with the robot control system.

The Automated Manufacturing Research Facility (AMRF), developed at the National Bureau of Standards, is a hierarchically organized small-batch metal machining shop (19). It separates sensory processing and robot control by a sophisticated world model. The world model has three complementary data representations. Lumia (20) describes the CAD-like section of the model. Shneier, Kent, and Mansbach (21) describe the octree and table representations supported by the model. The model generates hypotheses for the features which are either verified or refuted by empirical evidence. The sensory system's task is to update the appropriate parts of the world model with new or revised data as rapidly as possible. The control system accesses the world model as desired to obtain the current best guess concerning any aspect of the world. Shneier, Lumia, and Kent (22) describe the sensory system and its operation in greater detail. The AMRF was the first deliberate attempt to tie together sensory processing, world modeling, and robot control in a generic fashion. The system developed for the AMRF is applicable to more than manufacturing. This paper describes its use in space telerobotics.

2. A FUNCTIONAL SYSTEM ARCHITECTURE

The fundamental paradigm is shown in Figure 1. The control system architecture is a three legged hierarchy of computing modules, serviced by a communications system and a common memory. The task decomposition modules perform real-time planning and task monitoring functions, and decompose task goals both spatially and temporally. The sensory processing modules filter, correlate, detect, and integrate sensory information over both space and time in order to recognize and measure patterns, features, objects, events, and relationships in the external world. The world modeling modules answer queries, make predictions, and compute evaluation functions on the state space defined by the information stored in common memory. Common memory is a global database which contains the system's best estimate of the state of the external world. The world modeling modules keep the common memory database current and consistent.

2.1. Task Decomposition - H modules (Plan, Execute)

The first leg of the hierarchy consists of task decomposition H modules which plan and execute the decomposition of high level goals into low level actions. Task decomposition involves both a temporal decomposition (into sequential actions along the time line) and a spatial decomposition (into concurrent actions by different subsystems). Each H module at each level consists of a job assignment manager JA, a set of planners PL(i), and a set of executors EX(i). These decompose the input task into both spatially and temporally distinct subtasks as shown in Figure 2. This will be described in greater detail in section 4.

2.2. World Modeling - M modules (Remember, Estimate, Predict, Evaluate)

The second leg of the hierarchy consists of world modeling M modules which model (i.e. remember, estimate, predict) and evaluate the state of the world. The "world model" is the system's best estimate and evaluation of the history, current state, and possible future states of the world, including the states of the system being controlled. The "world model" includes both the M modules and a knowledge base stored in a common memory database where state variables, maps, lists of objects and events, and attributes of objects and events are maintained. By this definition, the world model corresponds to what is widely known throughout the artificial intelligence community as a "blackboard" (23). The world model performs the following functions:

1. Maintain the common memory knowledge base by accepting information from the sensory system.
2. Provide predictions of expected sensory input to the corresponding G modules, based on the state of the task and estimates of the external world.
3. Answer "What is?" questions asked by the executors in the corresponding level H modules. The task executor can request the values of any system variable.
4. Answer "What if?" questions asked by the planners in the corresponding level H modules. The M modules predict the results of hypothesized actions.

2.3. Sensory Processing - G modules (Filter, Integrate, Detect, Measure)

The third leg of the hierarchy consists of sensory processing G modules. These recognize patterns, detect events, and filter and integrate sensory information over space and time. The G modules at each level compare world model predictions with sensory observations and compute correlation and difference functions. These are integrated over time and space so as to fuse sensory information from multiple sources over extended time

intervals. Newly detected or recognized events, objects, and relationships are entered by the M modules into the world model common memory database, and objects or relationships perceived to no longer exist are removed. The G modules also contain functions which can compute confidence factors and probabilities of recognized events, and statistical estimates of stochastic state variable values.

2.4. Operator Interfaces (Control, Observe, Define Goals, Indicate Objects)

The control architecture defined here has an operator interface at each level in the hierarchy. The operator interface provides a means by which human operators, either in the space station or on the ground, can observe and supervise the telerobot. Each level of the task decomposition hierarchy provides an interface where the human operator can assume control. The task commands into any level can be derived either from the higher level M module, or from the operator interface. Using a variety of input devices such as a joystick, mouse, trackball, light pen, keyboard, voice input, etc., a human operator can enter the control hierarchy at any level, at any time of his choosing, to monitor a process, to insert information, to interrupt automatic operation and take control of the task being performed, or to apply human intelligence to sensory processing or world modeling functions.

The sharing of command input between human and autonomous control need not be all or none. It is possible in many cases for the human and the automatic controllers to simultaneously share control of a telerobot system. For example a human might control the orientation of a camera while the robot automatically translates the same camera through space.

2.4.1 Operator Control interface levels

The operator can enter the hierarchy at any level. The operator control interface interprets teleoperation in the fullest sense: a teleoperator is any device which is controlled by a human from a remote location. While the master-slave paradigm is certainly a type of teleoperation, it does not constitute the only form of man-machine interaction. At different levels of the hierarchy, the interface device for the human may change but the fundamental concept of teleoperation is still preserved. Table 1 illustrates the interaction an operator may have at each level.

The operator control interface thus provides mechanisms for entering new instructions or programs into the various control modules. This can be used on-line for real-time supervisory control, or in a background mode for altering autonomous telerobot plans before autonomous execution reaches that part of the plan.

2.4.2 Operator monitoring interfaces

The operator interfaces allow the human the option of simply monitoring any level. Windows into the common memory knowledge base permit viewing of maps of service bay layout, geometric descriptions and mechanical and electrical configurations of satellites, lists of recognized objects and events, object parameters, and state variables such as positions, velocities, forces, confidence levels, tolerances, traces of past history, plans for future actions, and current priorities and utility function values. These may be displayed in graphical form, for example using dials or bar graphs for scalar variables, shaded graphics for object geometry, and a variety of map displays for spatial occupancy.

2.4.3 Sensory processing/world modeling interfaces

The operator interface may also permit interaction with the sensory processing and/or world modeling modules. For example, an operator using a video monitor with a graphics overlay and a light pen or joystick might provide human interpretative assistance to the vision/world modeling system. The operator might interactively assist the model matching algorithms by indicating with a light pen which features in the image (e.g. edges, corners) correspond to those in a stored model. Alternatively, an operator could use a joystick to line up a wireframe model with a TV image, either in 2-D or 3-D. The operator might either move the wireframe model so as to line up with the image, or move the camera position so as to line up the image with the model. Once the alignment was nearly correct, the operator could allow automatic matching algorithms to complete the match, and track future movements of the image.

2.5. Common Memory

2.5.1. Communications

One of the primary functions of common memory is to facilitate communications between modules. Communications within the control hierarchy is supported by a common memory in which state variables are globally defined.

Each module in the sensory processing, world modeling, and task decomposition hierarchies reads inputs from, and writes outputs to, the common memory. Thus each module needs only to know where in common memory its input variables are stored, and where in common memory it should write its output variables. The data structures in the common memory then define the interfaces between the G, M, and H modules.

The operator interfaces also interact with the system through common memory. The operator displays simply read the variables they need from the locations in common memory. If the operator wishes to take control of the system, he simply writes command variables to the appropriate locations in common memory. The control modules that read from those locations need not know whether their input commands derived from a human operator, or from the next higher level in the autonomous control hierarchy.

2.5.2 State Variables

The state variables in common memory are the system's best estimate of the state of the world, including both the external environment and the internal state of the H, M, and G modules. Data in common memory are available to all modules at all levels of the control system.

The knowledge base in the common memory consists of three elements: maps which describe the spatial occupancy of the world, object-attribute linked lists, and state variables.

3. LEVELS IN THE CONTROL HIERARCHY

The control system architecture described here for the Flight Telerobot System is a six level hierarchy as shown in Figure 3. At each level in this hierarchy a fundamental transformation is performed on the task.

- Level 1 transforms coordinates from a convenient coordinate frame into joint coordinates. This level also serves joint positions, velocities, and forces.
- Level 2 computes inertial dynamics, and generates smooth trajectories in a convenient coordinate frame.
- Level 3 decomposes elementary move commands (E-moves) into strings of intermediate poses. E-moves are typically defined in terms of motion of the subsystem being controlled (i.e., transporter, manipulator, camera platform, etc.) through a space defined by a convenient coordinate system. E-move commands may consist of symbolic names of elementary movements, or may be expressed as keyframe descriptions of desired relationships to be achieved between system state variables. E-moves are decomposed into strings of intermediate poses which define motion pathways that have been checked for clearance with potential obstacles, and which avoid kinematic singularities.
- Level 4 decomposes object task commands specified in terms of actions performed on objects into sequences of E-moves defined in terms of manipulator motions. Object tasks typically define actions to be performed by a single multiarmed telerobot system on one object at a time. Tasks defined in terms of actions on objects are decomposed into sequences of E-moves defined in terms of manipulator or vehicle subsystem motions. This decomposition checks to assure that there exist motion freeways clear of obstacles between keyframe poses, and schedules coordinated activity of telerobot subsystems, such as the transporter, dual arm manipulators, multifingered grippers, and camera arms.
- Level 5 decomposes actions to be performed on batches of parts into tasks performed on individual objects. It schedules the actions of one or more telerobot systems to coordinate with other machines and systems operating in the immediate vicinity. For example, Level 5 decomposes service bay action schedules into sequences of object task commands to various telerobot servicers, astronauts, and automatic berthing mechanisms. Service bay actions are typically specified in terms of servicing operations to be performed by all the systems (mechanical and human) in a service bay on a whole satellite. This decomposition typically assigns servicing tasks to various telerobot systems, and schedules servicing tasks so as to maximize the effectiveness of the service bay resources.

Level 6 decomposes the satellite servicing mission plan into service bay action commands. Mission plans are typically specified in terms of satellite servicing priorities, requirements, constraints, and mission time line. The level 6 decomposition typically assigns satellites to service bays, sets priorities for service bay activities, generates requirements for spare parts and tool kits, and schedules the activities of the service bays so as to maximize the effectiveness of the satellite servicing mission. To a large extent the level 6 mission plans will be generated off line on the ground, either by human mission planners, or by automatic or semiautomatic mission planning methods.

4. DETAILED STRUCTURE OF THE N MODULES

The N module at each level consists of three parts as shown in Figure 4: a job assignment manager JA, one or more planners PL(s), and one or more executors EX(s).

The job assignment manager JA is responsible for partitioning the task command TC into s spatially or logically distinct jobs to be performed by a physically distinct planner/executor mechanisms. At the upper levels the job assignment module may also assign physical resources against task elements. The output of the job assignment manager is a set of job commands JC(s), $s=1, 2, \dots, N$ where N is the number of spatially, or logically, distinct jobs.

For each of these job commands JC(s), there exists a planner PL(s) and a executor EX(s). Each planner PL(s) is responsible for decomposing its job command JC(s) into a temporal sequence of planned subtasks PST(s,tt). Planning typically requires evaluation of alternative hypothetical sequences of planned subtasks. The planner hypothesizes some action or series of actions, the world model predicts the results of the action(s) and computes some evaluation function EF(s,tt) on the predicted resulting state of the world. The hypothetical sequence of actions producing the best evaluation function $EF(s,tt)_{max}$ is then selected as the plan PST(s,tt) to be executed by the executor EX(s).

$$PST(s,tt) = PL(s) JC(s), EF(s,tt)_{max}$$

where tt is the time sequence index for steps in the plan. tt may also be defined as a running temporal index in planning space, $tt = 1, 2, \dots, th$ where th is the value of the tt index at the planning horizon. The planning horizon is defined as the period into the future over which a plan is prepared. Each level of the hierarchy has a planning horizon of one or two expected input task time durations.

Each executor EX(s) is responsible for successfully executing the plan PST(s,tt) prepared by its respective planner PL(s). If all the subtasks in the plan PST(s,tt) are successfully executed, then the goal of the original task will be achieved. The executor operates by selecting a subtask from the current queue of planned subtasks and outputting a subcommand STX(s,t) to the appropriate subordinate N module at time t. The EX(s) module monitors its feedback FB(s,t) input in order to servo its output $STX(s,t)$ to the desired subtask activity.

$$STX(s,t+n) = EX(s) PST(s,t), FB(s,t)$$

where n = the number of state clock periods required to compute the function EX(s). n typically equals 1. The feedback FB(s,t) also carries timing and subgoal event information for coordination of output between executors at the same level. When the executor detects a subgoal event, it selects the next planned subtask from the queue.

Executor output STX(s,t) also contains requests for information from the world model M module, and status reports to the next higher (i+1) level in the N module hierarchy. The feedback FB(s,t) contains status reports from the N module at the i-1 th level indicating progress on its current task.

5. CONCLUSION

This paper has described a hierarchically organized control system and has shown how this generic system can be applied to telerobotic applications in space by considering the requirements of a flight telerobotic servicer for the space station.

REFERENCES

- (1) M. Brady, et.al., ed. Robot Motion: Planning and Control, (Cambridge, MIT Press, 1982).
- (2) D.E. Whitney, "Resolved Motion Rate Control of Manipulators and Human Prostheses," IEEE Trans. Man-Machine Systems **MMS-10**, 1969, p. 47.

- (3) D.E. Whitney, "The Mathematics of Coordinated Control of Prostheses and Manipulators," Journal of Dynamic Systems, Measurement, Control, Dec. 1972, p. 303.
- (4) R.P. Paul, "Manipulator Path Control," IEEE Int. Conf. on Cybernetics and Society, New York, p. 147.
- (5) R.P. Paul, "Manipulator Cartesian Path Control," IEEE Trans. Systems, Man, Cybernetics SMC-9, 1979, p. 702.
- (6) R.P. Paul, Robot Manipulators: Mathematics, Programming, and Control, (Cambridge, MIT Press, 1981.)
- (7) R. M. Taylor, "Planning and Execution of Straight-line Manipulator Trajectories," IBM J. Research and Development 23 1979, p. 424.
- (8) J.M. Hollerbach, "A Recursive Formulation of Lagrangian Manipulator Dynamics," IEEE Trans. Systems, Man, Cybernetics SMC-10, 11, 1980, p. 730.
- (9) J.Y.S. Luh, M.W. Walker, and R.P.C. Paul, "On-line Computational Scheme for Mechanical Manipulators," J. Dynamic Systems, Measurement, Control, 102, 1980, p. 69.
- (10) C.S.G. Lee, P.R. Chang, "Efficient Parallel Algorithm for Robot Inverse Dynamics Computation," IEEE Trans. on Systems, Man and Cybernetics, Vol. SMC-16, No. 4, July/August 1986, p. 532.
- (11) E.E. Binder, J.M. Herzog, "Distributed Computer Architecture and Fast Parallel Algorithm in Real-Time Robot Control," IEEE Trans. on Systems, Man and Cybernetics, Vol. SMC-16, No. 4, July/August 1986, p. 543.
- (12) M.H. Raibert and J.J. Craig, "Hybrid position/force control of manipulators," J. Dynamic Systems, Measurement, Control, June, 1981, p. 126.
- (13) W.A. Perkins, "A Model Based Vision System for Industrial Parts," IEEE Trans. on Computers, Vol. C-27, 1978, p. 126.
- (14) G.L. Gleason, G.J. Agin, "A Modular Vision System for Sensor-controlled Manipulation and Inspection," Proc. 9th Int. Symposium on Industrial Robots, 1979, p. 57.
- (15) M.R. Ward, et.al., "CONISIGHT An Adaptive Robot with Vision," Robotics Today, 1979, p. 26.
- (16) J. Albus, E. Kent, M. Washman, P. Mansbach, L. Palombo, M.O. Shneier, "Six Dimensional Vision System," SPIE, Vol. 336, Robot Vision, 1982, p. 142.
- (17) R.C. Bolles, R.A. Cain, "Recognizing and Locating Partially Visible Objects: The Local Feature-Focus Method," Int. Journal of Robotics Research, Vol. 1, 1982, p. 57.
- (18) R.C. Bolles, P. Horaud, M.J. Hannah, "3DFO: Three Dimensional Parts Orientation System," Proc. of The Int. Joint Conf. on Artificial Intelligence, August 1983, p. 1116.
- (19) J.A. Simpson, R.J. Hocken, J.S. Albus, "The Automated Manufacturing Research Facility of the National Bureau of Standards," Journal of Manufacturing System, Vol. 1, No. 1, 1983, p. 17.
- (20) R. Lumia, "Representing Solids for a Real-Time Robot Sensory System," Proc. Prolamat 1985, Paris, June 1985.
- (21) M.O. Shneier, E.W. Kent, P. Mansbach, "Representing Workspace and Model Knowledge for a Robot with Mobile Sensors," Proc. 7th Int. Conf. Pattern Recognition, 1984, p. 199.
- (22) M.O. Shneier, R. Lumia, E.W. Kent, "Model Based Strategies for High-Level Robot Vision," CVGIP, Vol. 33, 1986, p. 293.
- (23) A. Barr, E. Feigenbaum, The Handbook of Artificial Intelligence, (Los Altos, William Kaufman, 1981).

TABLE 1 -- OPERATOR INTERACTION AT EACH LEVEL

LEVEL	TYPE OF INTERACTION
At the servo	replica master, individual joint position, rate, or force controllers.
above level 1	joy stick to perform resolved motion force/rate control
above level 2	indicate safe motion pathways. Robot computes dynamically efficient movements
above level 3	graphically or symbolically define key poses. menus to choose elemental moves.
above level 4	specify tasks to be performed on objects.
above level 5	reassign telerobots to different service bays. insert, modify, and monitor plans describing servicing task sequences.
above level 6	reconfigure servicing mission priorities.

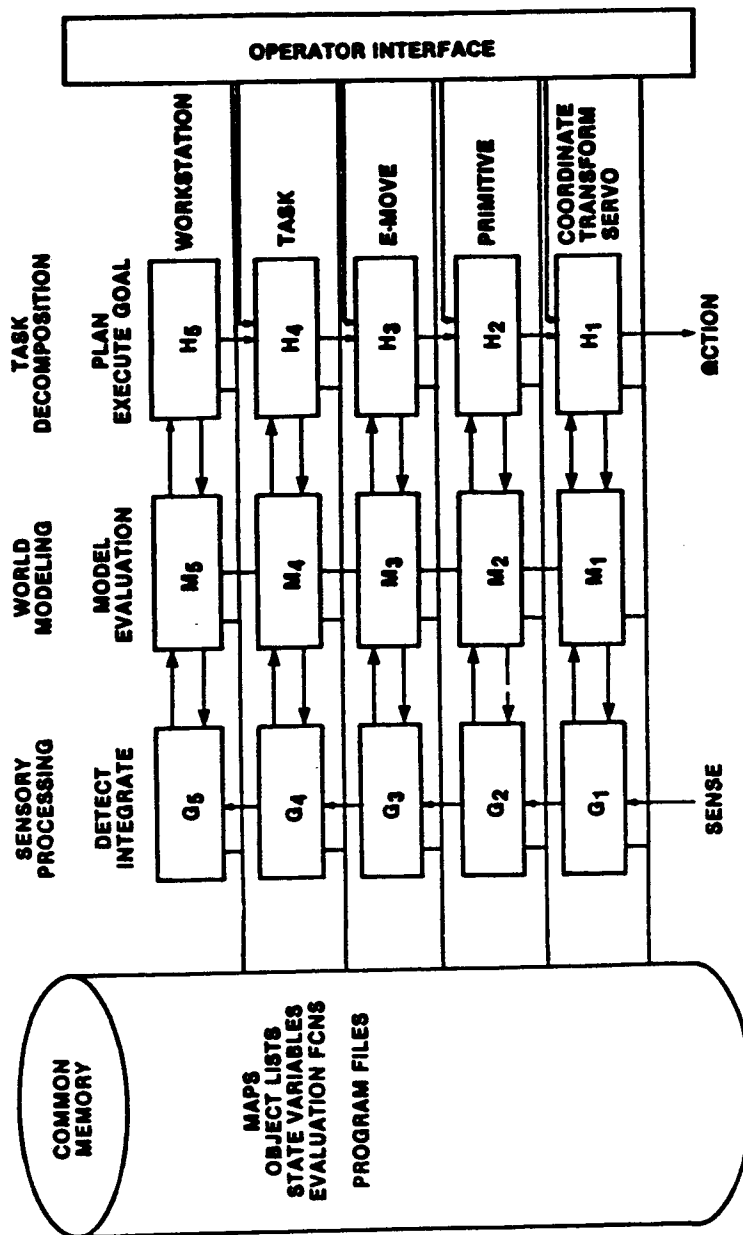


FIGURE 1 : A Hierarchical Control System Architecture for Intelligent Vehicles

Task Decomposition

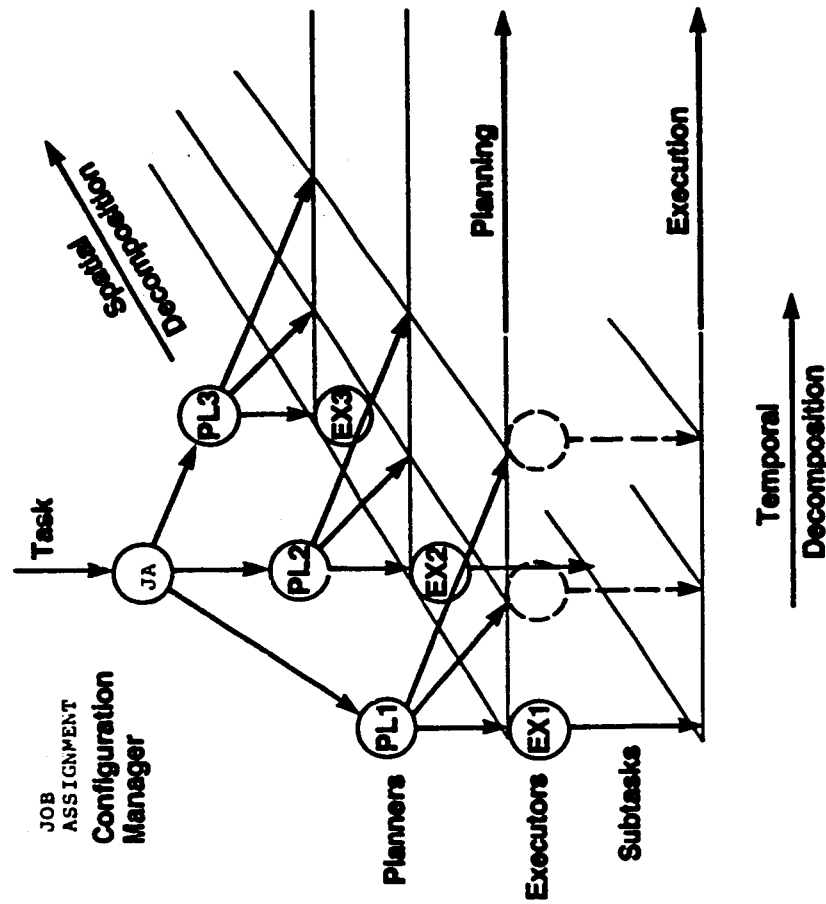


FIGURE 2: The job assignment JA performs a spatial decomposition of the task. The planners PL (j) and executors EX (j) perform a temporal decomposition

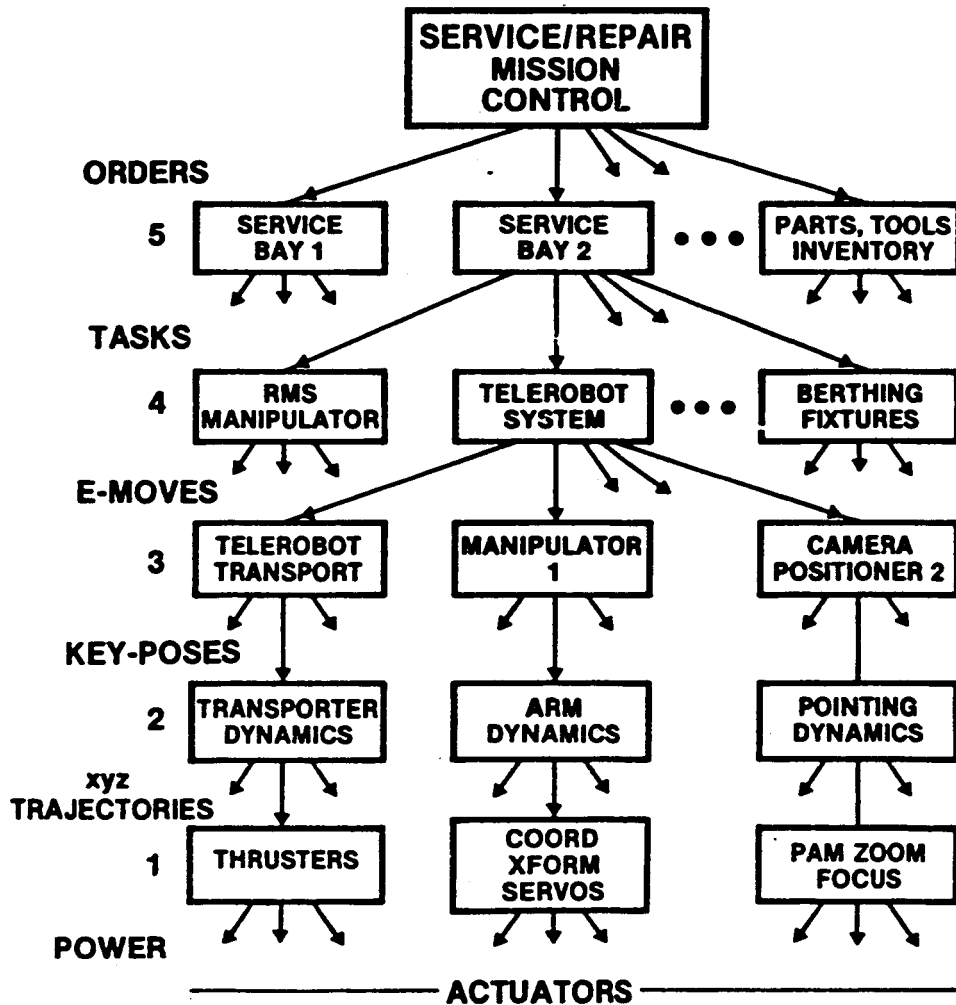


FIGURE 3: A six level Hierarchical Control System. Proposed for Multiple Autonomous Vehicles

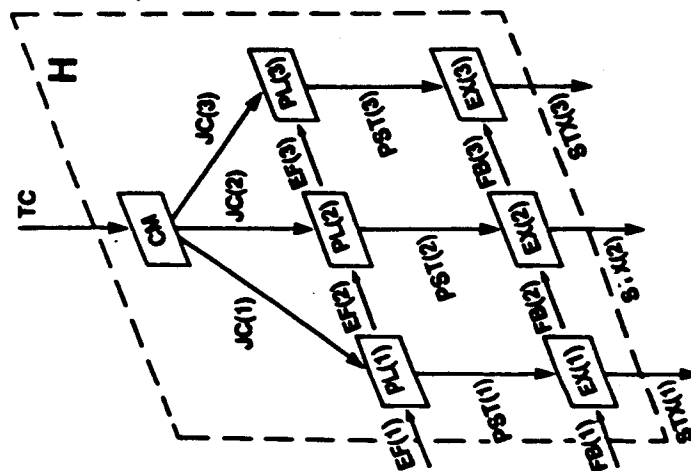


FIGURE 4: The H Module at each level has three parts: A job assignment module JA, Planners PL and a set of executors EX.