A Control System Architecture for Multiple Autonomous Undersea Vehicles (MAUV)

James S. Albus
National Bureau of Standards

D. Richard Blidberg
University of New Hampshire Marine Systems Lab

1. Introduction

The National Bureau of Standards and the University of New Hampshire are engaged in creating a pair of intelligent autonomous undersea vehicles based on the EAVE-EAST vehicle design [1] and the NASA/NBS standard reference model control system architecture (NASREM) [2]. The project is funded by the DARPA Naval Technology Office. Martin Marietta Baltimore is providing the environmental and sensor simulator for the project, and Decision Science Applications the value-driven logic. Robotic Technology Incorporated is providing performance analysis.

The issues being addressed are: hierarchical distributed control, knowledge based systems, real-time planning, world modeling, value-driven reasoning, intelligent sensing and communication, and cooperative problem solving by two intelligent vehicles in a natural and potentially hostile environment. In short, the project represents basic research on the nature of intelligent behavior.

The environment chosen for this study is the underwater environment of Lake Winnepausaki. The vehicles are of University of New Hampshire EAVE-East design. In October of 1987, two scenarios will be demonstrated: 1) A cooperative search and approach scenario, and 2) a cooperative search and map scenario.

The search and approach scenario mimics a deep ocean mission, and no use is made of bottom features. Search techniques will include various cooperative maneuvers such as fly-formation, split-circle-and-rendezvous, leader-follower, and high-low.

The search and map scenario mimics a harbor or coastal shallows survey mission, in either friendly or unfriendly waters. In this scenario bottom features such as ridges and gullies will be used for navigation and for cover from detection by unfriendly agents. The two vehicles will demonstrate the ability to search for and map the positions of objects on the lake bottom, to sense and plot the position of acoustic beacons, and to perform a number of maneuvers relative to such beacons such as using the bottom topography to "shadow" the vehicles from the beacons so as to minimize the probability of being detected by active sonar

systems located at the site of the beacons. Obstacle avoidance sonar and bottom distance sonar will give the vehicles the ability to follow bottom topographic features such as gullies and ridges.

The scenarios chosen have are designed to study, and attempt to mimic, both aggressive predation and exploratory curiosity. Such behavioral activities are common in all intelligent creatures in nature, and are particularly refined in humans. It is well known that aggression, predation, exploratory behavior, and curiosity lie at the very root of intelligence. Intelligent animals not only exhibit these behaviors, but if we subtract resting, sleeping, feeding, and grooming, they spend most of their time engaged in such behaviors. Thus, the demonstration scenarios may be said to illustrate intelligent cooperative behavior (including communication) between intelligent beings in an uncertain and potentially dangerous world.

There are many advantages to two vehicles. One vehicle can search an area while the other relays messages about what has been found. One vehicle can illuminate a target while another takes pictures. Two vehicles can hunt in pairs such that while one moves, the other lies still and listens for prey. One can attract the prey's attention, while the other closes in for the kill. When in danger, one vehicle can draw attention to itself, while the other gets away with information. Value driven logic allows the value of each vehicle's survival to be weighed in the balance against the success of the mission.

Two vehicles also permits research on communication as a goal directed activity. For example, What information should be transmitted? For what purpose? and When? When is the value of a piece of information of sufficient value to incur the risk to survival of revealing one's presence by transmitting a message? What are communication strategies which balance the risk against the benefits?

How should control systems be structured so that both vehicles can behave equally intelligently when they are apart, but one vehicle is recognized as the leader of the pack when they are together? How do they share knowledge acquired by only one? What if they cannot agree on a strategy? These are deep issues for software and control system engineers, particularly those interested in the reliability of autonomous systems.

The National Bureau of Standards (NBS) is pursuing this project because of its broad interest in advanced automation, including intelligent machine systems. NBS is conducting research in performance measures and interface standards for intelligent machines in several application areas: including manufacturing, construction, undersea research, and space telerobotics. This particular project derives from the NBS interest in autonomous undersea vehicles as members of the class of intelligent machines. NBS is supplying a hierarchical control system

architecture which incorporates real-time planning, world modeling, and value driven reasoning. The NBS control system is a prototype for a proposed Standard Reference Architecture Model for intelligent machine systems.

The University of New Hampshire (UNH) is involved because of its interest in autonomous undersea vehicles, and Knowledge Based Systems for controlling them. UNH is supplying the vehicles, the operational expertise in autonomous undersea vehicles, and a Knowledge Based Control System for performing its own set of experiments with intelligent control.

The MAUV project has been in existance for about one year, and represents about 20 person-years of effort per year.

2. The MAUV Vehicles

Figure 1 is a picture, and Figure 2 a diagram, of a MAUV vehicle. The MAUV vehicles are a derivitive of the EAVE-EAST vehicle, developed at the Marine Systems Engineering Lab of the University of New Hampshire by Richard Blidberg and his associates. MAUV is gravity stablized in pitch and roll, with thrusters that allow it to be controlled in x, y, z, and yaw. It is a battery powered with the batteries stored in cylindrical tanks at the bottom of the vehicle, and flotation tank on the upper part of the vehicle. Each vehicle carries three acoustic navigation transponders which allow it to measure the range and bearing to navigation bouys placed in the water. The vehicle has a compass, pressure and temperature sensors, and bottom and surface sounders. In front, it has an obstacle avoidance sonar consisting of five narrow beam acoustic transmitter-receivers. These are arranged such that the center sonar beam points straight ahead, two point ten degrees to the right and left, and two point ten degrees up and down from the center beam.

The Marine Systems Engineering Laboratory (MSEL) at the University of New Hampshire has, over the past 11 years, focussed its research efforts soley on intelligent underwater systems. This effort is manifested in the EAVE robot system; an autonomous underwater vehicle (AUV) which is evolutionary in nature.

The first Eave vehicle system was completed in 1978 and was tasked to autonomously follow an underwater pipeline using acoustic sensors to recognize the pipe. In 1983 the vehicle demonstrated its ability to acurately follow a pre defined path using acoustic transponder navigation. This demonstration was followed, in 1984, by the addition of a cavitation cleaning device on the vehicle. The vehicle was then programmed to go to a specified location on an underwater structure, attach itself to that structure using docking arms (controlled by the onboard computer system) and clean the underwater structure using the Lithium gas driven cleaning system.

In 1986, two new EAVE vehicle systems were designed which were similar to the original EAVE vehicle but with greatly increased computer capability. The new vehicles resulted from the interest within MSEL, over the previous few years, in addressing research issues dealing with the ability of an AUV to have an on-board decision making capability allowing the system to accomplish much more complex tasks where path and task planning and replanning is required without human intervention. In order to accomplish this goal, it was necessary to apply techniques from artificial intelligence research to develop a knowledge based guidance and control system; hence the need for a more powerful computational environment. This interest in developing guidance and cotrol architectures matched the interest of the Robotics Group at the National Bureau of Standards and a cooperative program was begun under DARPA sponsorship. The goal of that program was to consider the problems associated with two autonomous systems cooperating and communicating with each other to accomplish a specified task.

The NBS group will interface with the low level control systems of two EAVE vehicles and implement the higher level functions required to control the vehicles systems using their RCS (Robot Control System) software architecture. They will then program this architecture to control the overall vehicle systems to perform a set of demonstration mission scenarios which require the two vehicles to cooperate in accomplishing a task. The role of MSEL in this program is twofold. First, to build two new EAVE vehicles and to complete all of the programming required for the low level control functions. Secondly, to continue the developments within MSEL to define a software and hardware architecture for the high level functions (similar to the NBS effort) of the vehicle system. The Architecture being devloped at MSEL [IEEE Paper for a reference] differs from the NBS architecture in some ways but is also very similar in basic concept. By working on both of these systems in parallel it will be possible to compare the two systems and gleen from the two development efforts the "best of both". The result of this dual effort will provide insight into the real problems associated with the guidance and control of an autonomous underwater vehicle in a real world system.

3. THE MAUV CONTROL SYSTEM ARCHITECTURE

The MAUV control system architecture incorporates a number of concepts developed in previous and on-going robotics research programs, including the NASA telerobotics program [2], the DARPA Autonomous Land Vehicle [3], the Air Force/DARPA Intelligent Task Automation program [4], the supervisory control concepts pioneered by Sheridan at MIT [5], and the hierarchical control system developed for the Automated Manufacturing Research Facility at the National Bureau of Standards [6-8]. The MAUV architecture integrates many artificial intelligence concepts such as goal decomposition, hierarchical real-time planning, model driven image analysis [9-13], blackboards [14], and expert

systems into a systems framework with modern control concepts such as multivarient state space control, reference model adaptive control, dynamic optimization, and learning systems [14-17] The MAUV architecture framework also readily accommodates concepts from operations research, differential games, utility theory, and value driven reasoning [18-20].

A block diagram of the NBS MAUV control system architecture is shown in Figure 3. In the MAUV control system architecture the task decomposition modules perform real-time planning and task monitoring functions, and decompose task goals both spatially and temporally, as shown in Figure 4. The sensory processing modules filter, correlate, detect, and integrate sensory information over both space and time so as 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 global memory, as shown in Figure 5. Global memory is a database which contains the system's best estimate of the state of the external world. The world modeling modules keep the global memory database current and consistent.

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

The task decomposition hierarchy consists of H modules which plan and execute the decomposition of high level goals into low level actions. Task decomposition involves both a spatial decomposition (into concurrent actions by different subsystems), and a temporal decomposition (into sequential actions along the time line).

Each H module at each level consists of three sublevels as shown in Figure 4:

1) a planner manager PM

2) a set of planners PL(i) and

3) a set of executors EX(i).

These three sublevels decompose the input task into both spatially and temporally distinct subtasks as shown in Figure 4.

For each level:

2.1.1 Planner Manager

The planner manager PM is responsible for partitioning the task command into i spatially or logically distinct jobs to be performed by i physically subsystems. Each subsystem has

a planner/executor mechanisms. At the upper levels the job assignment module may also assign physical resources to the subsystems for each job.

2.1.2 Planners

For each job assigned to a subsystem, there exists a planner PL(i) and a executor EX(i). Each planner is responsible for decomposing its job assignment into a temporal sequence of planned subtasks to be executed by the respective executor.

Planning typically requires evaluation of alternative hypothetical sequences of planned subtasks. Each planner PL(i) functions by hypothesizing some action or series of actions. The world model then predicts the results of the action(s) and computes the value of predicted resulting state of the world. This value is computed by an evaluation function which performs a priority weighted cost-benefit analysis on the predicted results. The hypothetical sequence of actions producing the best evaluation is then selected as the plan to be executed by the executor EX(i).

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 approximately two input task time durations. Replanning is done at cyclic intervals, as well as whenever emergency conditions arise. The cyclic replanning interval is about an order of magnitude less than the planning horizon (or about equal to the expected output subtask time duration). Emergency replanning begins immediately upon the detection of emergency conditions.

2.1.3 Executor

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

The feedback also carries subgoal event information and a time of day clock for coordination of output between executors at the same level. When the executor detects a subgoal event, it steps to the next subtask in the plan.

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

Def: 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 global memory 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 in the artificial intelligence community as a "blackboard" [14].

The world modeling leg of the hierarchy consists of M modules which model (i.e. remember, estimate, predict) and evaluate the state of the world.

As shown in Figure 7, the M modules at various levels:

2.2.1. Maintain the global memory knowledge base, keeping it current. The M modules update the knowledge base based on correlations and differences between model predictions and sensory observations.

2.2.2. Provide predictions of expected sensory input to the corresponding G modules, based on the state of the task and estimates of the external world.

2.2.3. Answer "What is?" questions asked by the planners and executors in the corresponding level H modules. The task executor requests information about the state of the world, and uses the answers to monitor and servo the task, and/or to branch on conditions to subtasks that accomplish the task goal.

2.2.4. Answer "What if?" questions asked by the planners in the corresponding level H modules. M modules predict the results of hypothesized actions.

2.2.5. Evaluate the current situation and potential future consequences of hypothesized actions by applying evaluation functions to current states and to future states expected to result from hypothesized actions. The evaluation functions include a set of values assigned to events such as vehicle survival, subtask completion, and information gathered by the vehicles. They also include a set of priorities assigned to each of these values.

The mission level priorities are defined at the beginning of the mission and typically are not changed during the mission. Task priorities for lower levels

are derived from the mission level priorities in the context of specific state variables contained in the world model.

The evaluation functions thus provide value driven decision logic [20] at several different hierarchical levels. Working together, the world model predictors, evaluation functions, and the planners are able to search the space of possible futures, and choose the sequence of planned actions that produce the best evaluation. The executors are also able to apply value driven logic to moment by moment behavioral decisions.

2.3 Global Memory

Def: Global memory is the database wherein is stored knowledge about the state of the world including the internal state of the control system.

2.3.1 Contents of Global Memory

The knowledge in the global memory consists of:

a) Maps

Maps describe the spatial occupancy of the world. A map is a spatially indexed database showing the relative position of objects and regions. At different levels the maps have different resolution. Resolution increases at each successively lower level, while area covered by the map increases at each successively higher level. The maps at different levels thus represent a pyramid structure. Maps may also contain a number of overlays. These overlays may indicate values such as utility, cost, risk, etc. to be used for planning.

b) Lists

All known objects, features, regions, relationships, and events are listed in the global memory database indexed by name, along with frames containing their attributes. Object and feature frames contain information such as position, velocity, orientation, shape, dimensions, reflectance, color, mass, and other information of interest. Event frames contain information such as start and end time, duration, type, cost, payoff, etc. Recognized objects and events may also have associated with them confidence levels, and degrees of believability and dimensional certainity.

At different levels, object frames have different levels of detail and spatial resolution, and event

frames have different levels of temporal resolution.

c) State Variables

The state variables in global 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 global memory is available to all modules at all levels of the control system.

2.3.2 Implemention of Global Memory

Global memory in the MAUV architecture is not located in a single physical database, but is distributed over several computers, memory boards, and mass storage devices on a VME bus. Global memory is, in fact, distributed over more than one vehicle. Variables in global memory are globally defined, i.e., they may be accessed (read or written) by name from local processes running at any level. Of course, the time required to access a global variable is not the same for all processes. For example, in order for a global variable in vehicle-A to be read or updated by a process in vehicle-B, the two vehicles may have to rendezvous and communicate world model updates. This may take many minutes or hours.

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

The sensory processing leg of the MAUV control hierarchy consists of G modules which recognize patterns, detect events, and filter and integrate sensory information over space and time. The G modules are similar to the H modules in that they also consist of three sublevels which:

1) compare sensor observations with world model predictions
2) integrate correlation and difference over time
3) integrate correlation and difference over space

These spatial and temporal integrations 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 global 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.

4. Functional Levels in the MAUV Task Decomposition Hierarchy

The flow of commands and status feedback in the MAUV task decomposition hierarchy is structured into six layers, as shown in Figure 3 and 6. High level commands, or goals, are decomposed both spatially and temporally through a hierarchy of control levels, each with planners and executors, into strings and patterns of subcommands.

Each task decomposition module thus represents a node in a command and control tree, which receives input commands from one and only one supervisor, and outputs subcommands to a set of subordinate modules at the next level down in the tree. Outputs from the bottom level consist of drive signals to motors and actuators.

At each layer of the MAUV architecture a different functional transformation is performed.

Level 1 -- Coordinate Transform/Servo

transforms coordinates from a vehicle coordinate frame into actuator coordinates. This level also servos actuator power.

Level 2 -- Dynamic (Primitive)

works in vehicle or world coordinates. It computes inertial dynamics, and generates smooth trajectory positions, velocities, accelerations for efficient vehicle manuevers.

Level 3 -- Elementary Move (E-Move)

works in both symbolic and geometric space. It decomposes elementary move commands (E-moves) into strings of intermediate poses, or dynamic (primitive) level commands.

As shown in Figure 6, each MAUV vehicle consists of three subsystems: pilot, communications, and sonar. E-moves are defined for each vehicle subsystem.

A pilot E-Move can be defined as a smooth coordinated motion of the vehicle designed to achieve some position, orientation, or "key-frame pose" in state-space, or space-time. The level 3 pilot planner computes clearance with obstacles sensed by on-board sensors and generates strings of intermediate poses that define motion pathways between key-frame poses.

A communications E-move is a message. The level 3 communications planner encodes messages into strings of symbols, adds redundancy for error detection and

correction, and formats the symbols for transmission.

A sonar E-Move may be defined as a temporal pattern of sonar pings or a scanning pattern for a passive listening beam designed to obtain a specific type of information about a specific target. The level 3 sonar planner decomposes sonar E-Moves into patterns of sonar pings and scanning beam dwell times.

Level 4 -- Vehicle Task

Level 4 works in object/task space. It decomposes vehicle commands, defined in terms of tasks to be performed by a single AUV on a target object, into sequences of E-moves, defined in terms of vehicle subsystem actions on aspects, or features, of an object.

The level 4 planner manager decomposes vehicle tasks into work elements to be performed by the various vehicle subsystems. It also coordinates, syncronizes, and resolves conflicts between vehicle subsystem plans.

The level 4 planners schedule sequences of E-Moves for the pilot, the communications, and the sonar subsystems.

The level 4 pilot planner checks the world model map to assure that there exists at least one pathway between keyframe poses. From the map, it estimates the cost, risk, and benefit of various routes and chooses a path that maximizes some cost-benefit evaluation function.

The level 4 communications planner schedules the messages to be sent. It computes the value of each message, its urgency, the risk of breaking communications silence, the power needed to make the message heard, and decides if and when to send the message.

The level 4 sonar planner analyses the nature of the target, plans scanning patterns for passive or active beams, estimates the value of taking an active sonar sounding, and compares that against the risk of breaking silence.

Level 5 -- Group

Level 5 decomposes group tasks into vehicle tasks. Group tasks define actions to be performed on groups of objects by groups of autonomous vehicles. Level 5 decomposes these into sequences of tasks for individual vehicles to perform on individual objects. The level 5

route planners use a Group level world model map to compute vehicle transit times. Level 5 also estimates costs, risks, and benefits of various vehicle task sequences. It schedules the actions of each AUV to coordinate with the other AUV in the group so as to maximize the effectiveness of the MAUV group.

Level 6 -- Mission

Level 6 decomposes a commanded MAUV mission into a sequence of group tasks and assigns priorities and values to various group tasks and mission substasks. Missions are typically specified by a list of mission objectives, priorities, requirements, and time line constraints. The level 6 planning manager assigns mission objectives to MAUV groups. The level 6 planner generates requirements for resources such as fuel, and time, develops a schedule, and sets priorities for each respective group assignment. It schedules the activities of the group so as to maximize the effectiveness of the total mission.

5. Real Time Planning in the MAUV Hierarchy

One of the unique features of the MAUV hierarchical computing architecture is its multiple levels of planners. These provide a unique ability to deal with the real-time aspects of planning and task execution. Beacause planning is done at each level, plans at any one level typically consist of less than ten steps, and hence can be derived relatively quickly.

Planning is done top-down. The highest level plan covers the entire backlog of tasks to be accomplished by the end of the mission. At each lower level, plans are formulated (or selected) in real-time to accomplish the next step (or two) in the plan of the level immediately above.

Figure 7 shows an example of three levels of hierarchical planning activity. The activity represented at the highest level is input to the top level H module as a task command. This task is decomposed by the job assignment manager and three planners of the top H module into three concurrent plans consisting of four activity-event pairs each. The first executor of the top level H module outputs the current subtask command in its plan to a second level H module.

At the second level, the input task command is further decomposed by the H module into three concurrent plans, each consisting of four subtasks each. At the third level, the H module further decomposes its input task into three plans of four subtasks.

At each level, the final subgoal events in the plans correspond

to the goal of the input task. At each successively lower level, the planning horizon becomes shorter, and the subtask activities become more detailed and fine structured.

The MAUV control system always has a hierarchy of plans in place. The MAUV vehicles begin each mission with a mission plan, with a planning horizon to the end of the mission. The group level always has a plan for how to accomplish the next two steps in the mission plan, and so one down the hierarchy. If the mission goes as planned, each level of the control system will always be able to anticipate the next subtask, and there is no need to pause to replan. However, if unexpected events cause a plan at some level of the hierarchy to become obsolete, a new plan must be generated. If a new plan cannot be developed in time, the real-time executor will be without a plan. A condition in which one or more levels has no plan available for execution can be described as a state of confusion. The executor must then execute some preplanned emergency action until the emergency planner can generate a new plan.

6. Implementation

The MAUV control architecture is being implemented on the computing systems shown in Figure 8. In each vehicle, a VME bus supports high bandwidth communication between sensory processing, world modeling, task planning, and task execution modules at each level of the hierarchy. The set of computing modules is partitioned between three separate single board computers so as to maximize the use of parallel computation. A two megabyte common memory board is used for communication between processes, and a 800 megabyte optical disk will be used for mass storage. pSOS pRISM from Ironics is being used as the real-time multiprocessor operating system.

Also shown in Figure 8 is the MAUV software development and simulation environment. A variety of software development tools, such as Sun workstations, a VAX 11/785, a micro-VAX, and Iris graphics systems, PC's and Duals are being be tied into the development environment for code development and simulation. Translators and cross compilers are being provided so that software developed in this environment can be downloaded into the 68020 target hardware for real-time execution.

The MAUV computing architecture can accomodate many additional computers as the complexity of the tasks assigned to the vehicles is increased. In the future, as vision and sonar arrays are added to the vehicles, the MAUV computer systems will incorporate special purpose computing elements, such as pipeline image processors and vector accelerators. These will physically interface with the control system through the VME bus.

8. REFERENCES

[1] D. R. Blidberg, "Guidance Control Architecture for the Eave Vehicle", IEEE Journal Oceanic Engineering, October 1986.

[2] J. S. Albus, H. G. McCain, and R. Lumia, "NASA/NBS Standard Reference Model for Telerobot Control System Architecture (NASREM)", NBS Technical Report (to be published)

[3] J. Lowrie, et.al. "Autonomous Land Vehicle" Annual Report, ETL-0413, Martin Marietta Denver Aerospace, July 1986.

[4] John Graham, "Intelligent Task Automation (Phase II) Sixth Quarter Quarterly Report", Martin Marietta Denver Aerospace, March, 1987.

[5] T.B. Sheridan "Supervisory Control of Remote Manipulators, Vehicles and Dynamic Processes". In Rouse, W.B. (Ed.) Advances in Man-Machine Systems Research, Vol. 1, NY JAI Press, 49-137, 1984.

[6] 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.

[7] C. McLean, M. Mitchell, E. Barkmeyer, "A Computer Architecture for Small Batch Manufacturing," IEEE Spectrum, May, 1983, p. 59.

[8] J. Albus, C. McLean, A. Barbera, M. Fitzgerald, " An Architecture for Real-Time Sensory-Interactive Control of Robots in a Manufacturing Environment," 4th IFAC/IFIP Symposium on Information Control Problems in Manufacturing Technology, Gaithersburg, Oct., 1982.

[9] 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.

[10] W.A. Perkins, "A Model Based Vision System for Industrial Parts," IEEE Trans. on Computers, Vol. C-27, 1978, p. 126.

[11] 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.

[12] R.C. Bolles, P. Horaud, M.J. Hannah, "3DPO: Three Dimensional Parts Orientation System," Proc. of The Int. Joint Conf. on Artificial Intelligence, August 1983, p. 1116.

[13] T.F. Knoll and R.C. Jain, "Recognizing Partially Visible Objects Using Feature Indexed Hypotheses," Proc. IEEE Conf. on Robotics and Automation, San Francisco, 1986, p. 925.

[14] A. Barr, E. Feigenbaum, The Handbook of Artificial Intelligence, (Los Altos, William Kaufman, 1981).

[15] M. Brady, et.al., ed. Robot Motion: Planning and Control, (Cambridge, MIT Press, 1982).

[16] D.E. Whitney, "Resolved Motion Rate Control of Manipulators and Human Prostheses," IEEE Trans. Man-Machine Systems MMS-10, 1969, p. 47.

[17] R.P. Paul, Robot Manipulators: Mathematics, Programming, and Control, (Cambridge, MIT Press, 1981.)

[18] J.C. Crowley, "Navigation for an Intelligent Mobile Robot," IEEE Journal of Robotics and Automation, Vol. RA-1, No. 1, 1985, p. 31.

[19] George E. Pugh, D.F. Noble "An Information Fusion System for Wargaming and Information Warfare Applications", Decision-Science Applications, Inc., Report No. 314, May 1981

[20] George Pugh, G.L. Lucas, "Application of Value-Driven Decision Theory to the Control and Coordination of Advanced Tactical Air Control Systems", Decision-Science Applications,Inc., Report No. 218, April 1980.

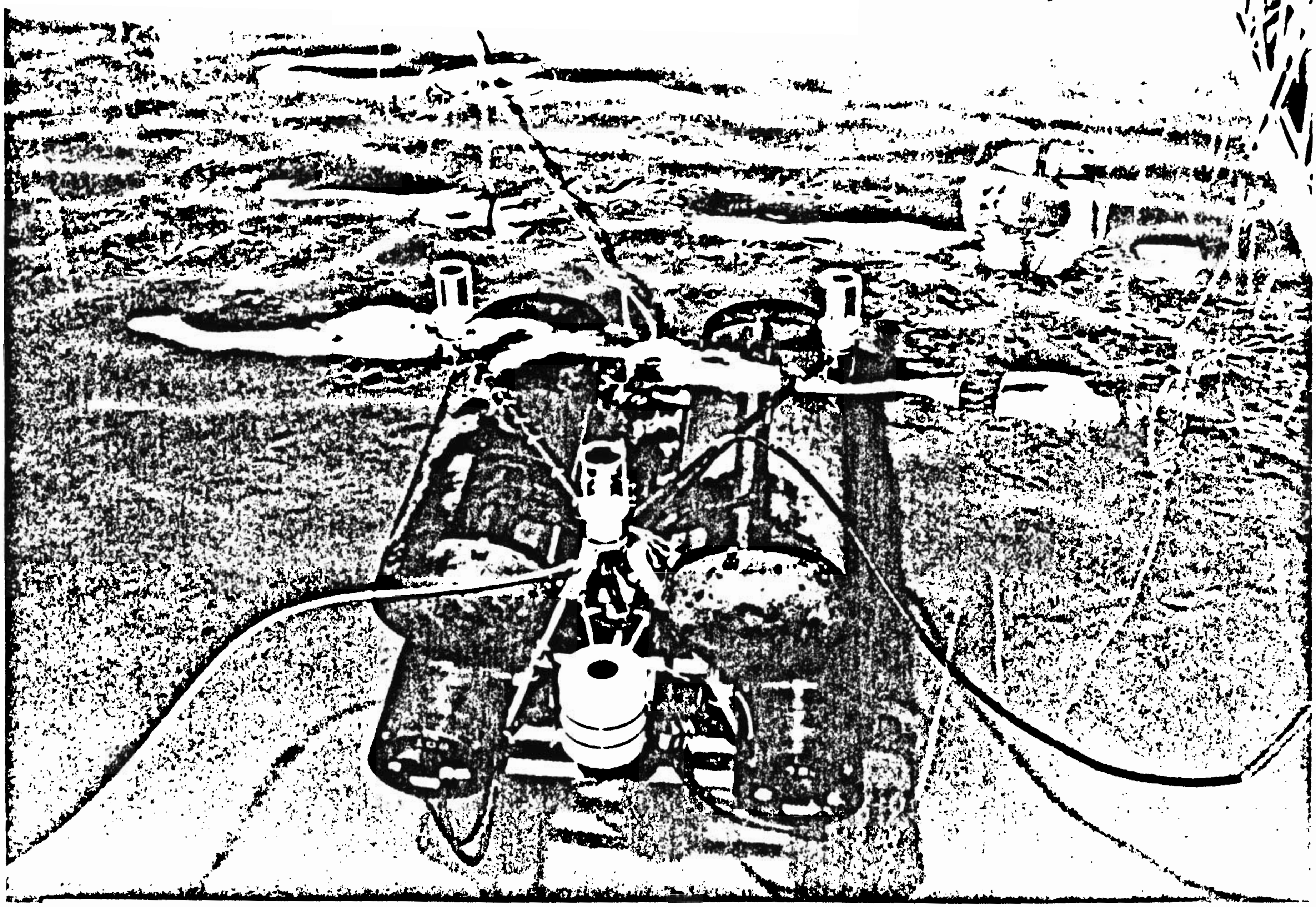

FIGURE 1: One of the MAUV Vehicles undergoining tests in a tank. Ropes and tethers will be removed during operational manuevers.

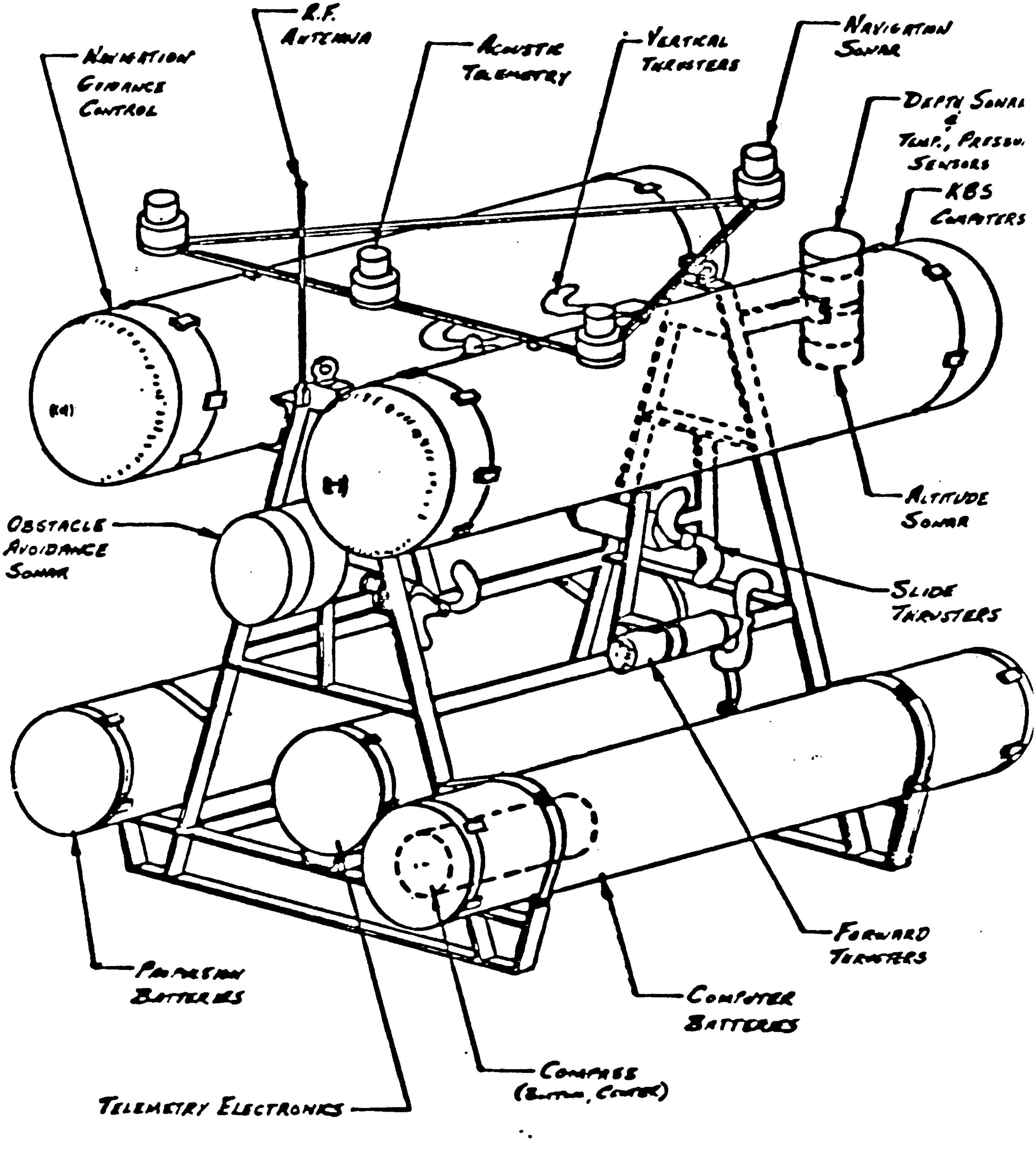

FIGURE 2: A diagram of a MAUV Vehicle.

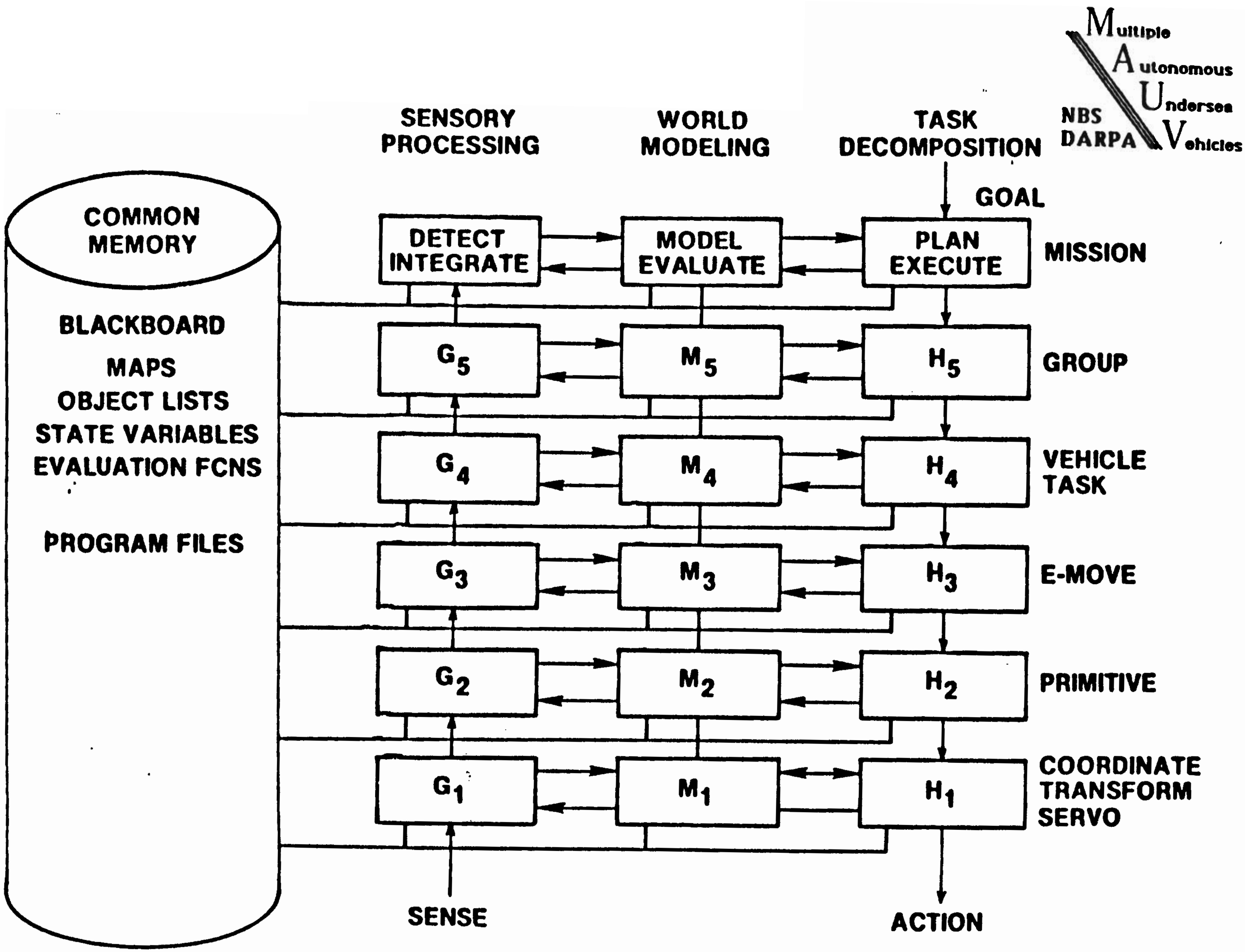

FIGURE 3: A block diagram of the NBS MAUV Control System

Task Decomposition

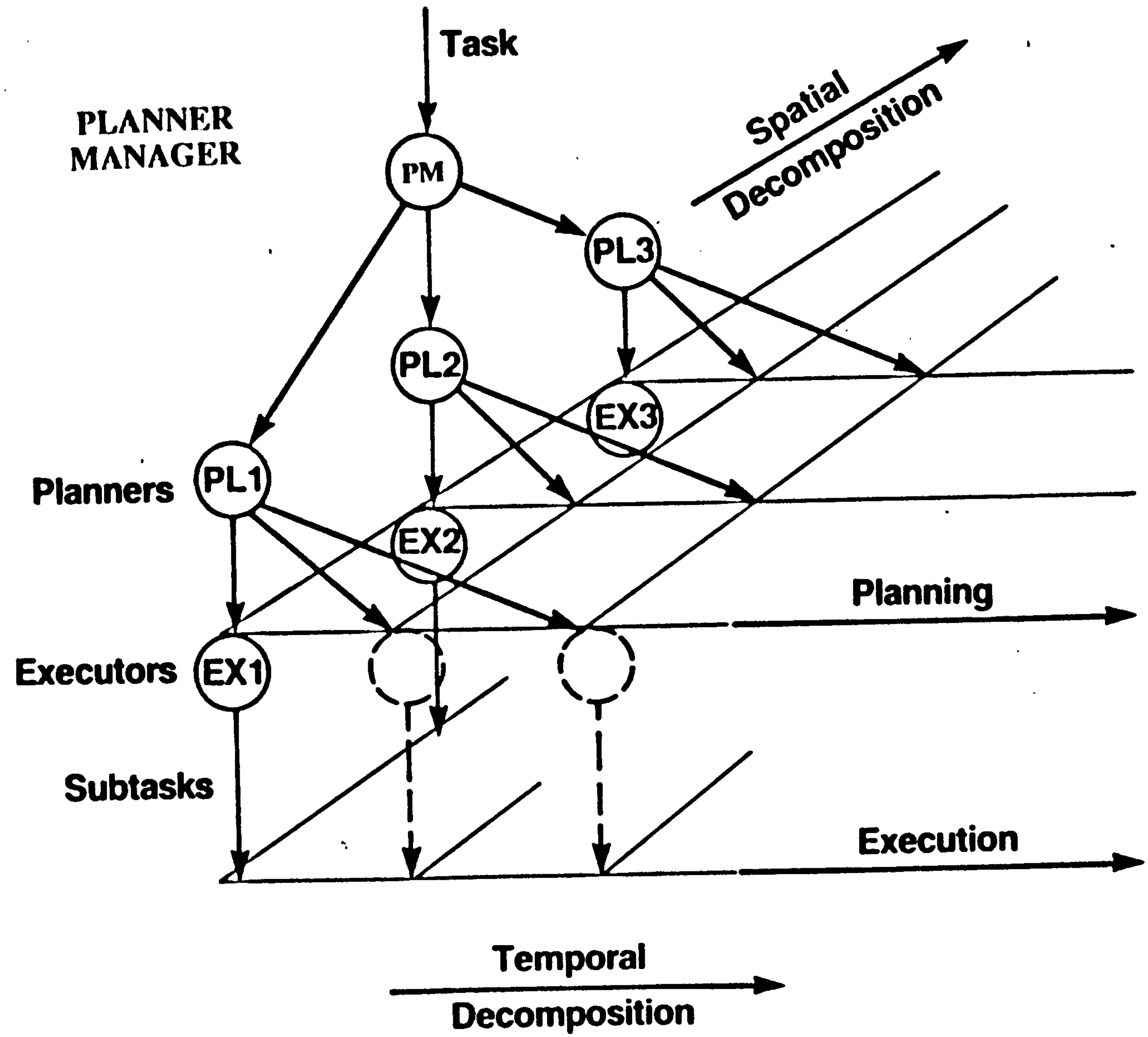

FIGURE 4: Internal structure of the Task Decomposition Modules in the MAUV Control System Architecture at every level of the hierarchy

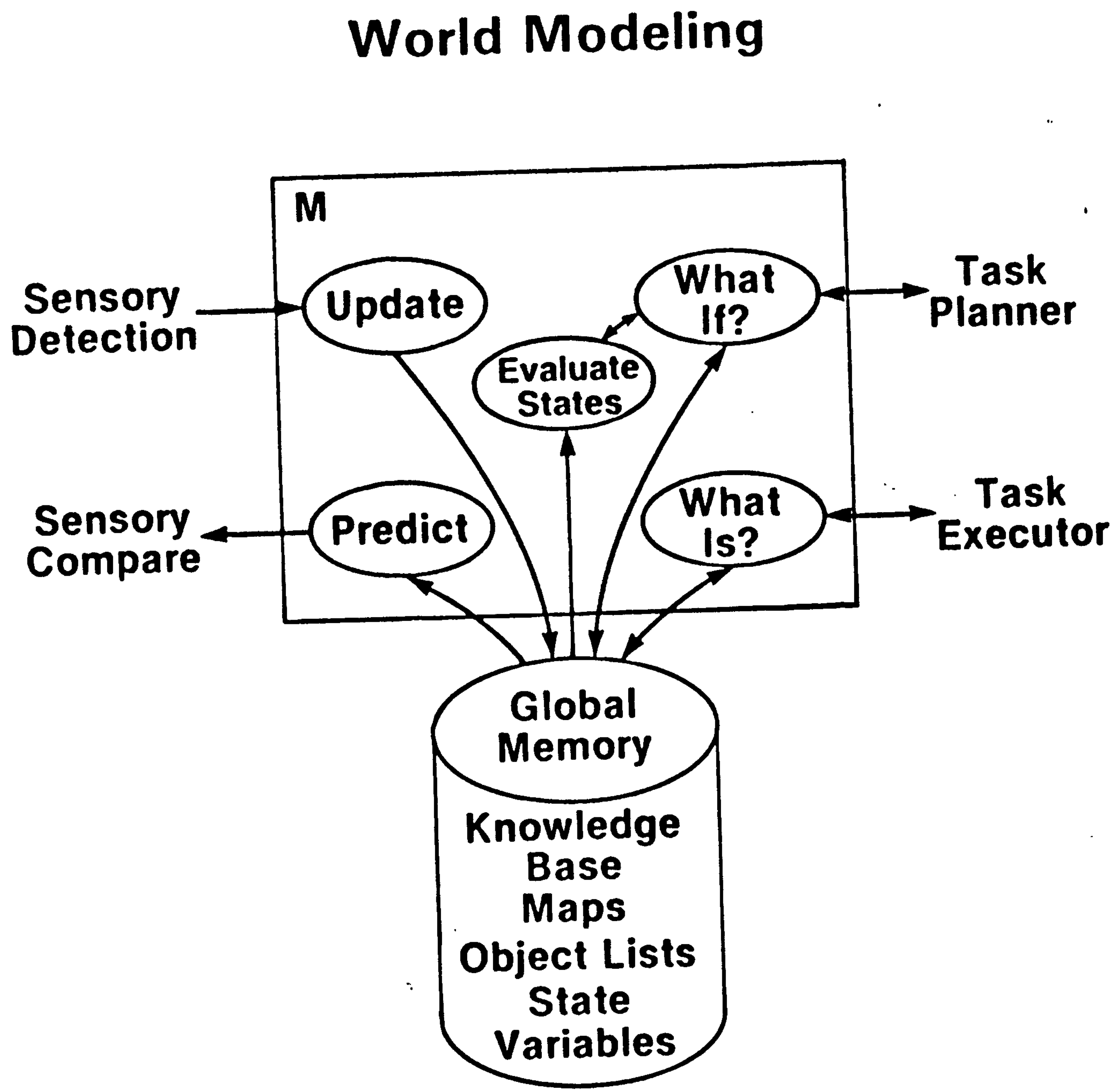

FIGURE 5: Internal structure of the World Modeling Modules at each level.

NBS DARPA
Multiple Autonomous Undersea Vehicles

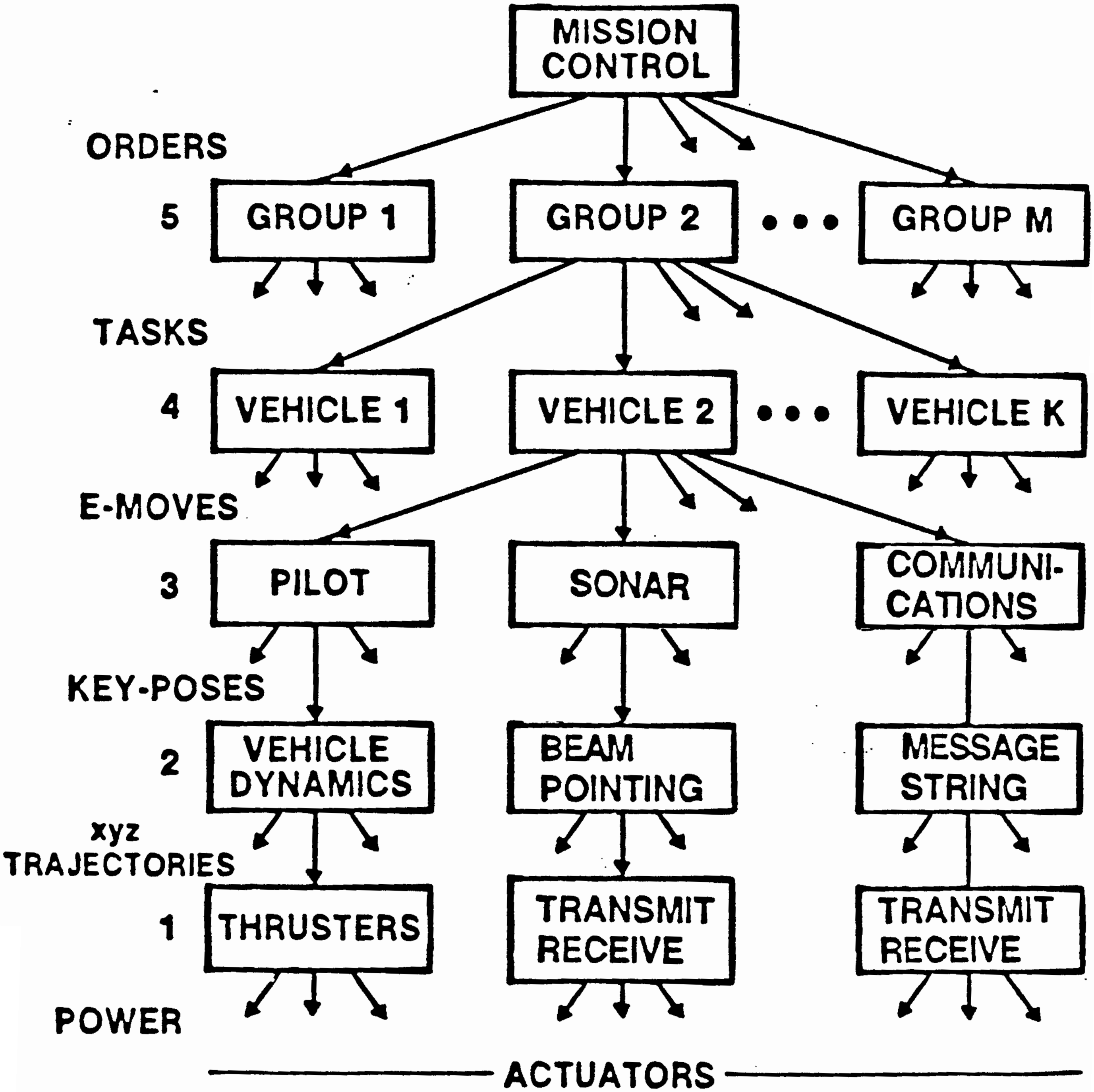

FIGURE 6: Tree structure of the MAUV task decomposition hierarchy (spatial decomposition). The current MAUV project has only one group of two vehicles.

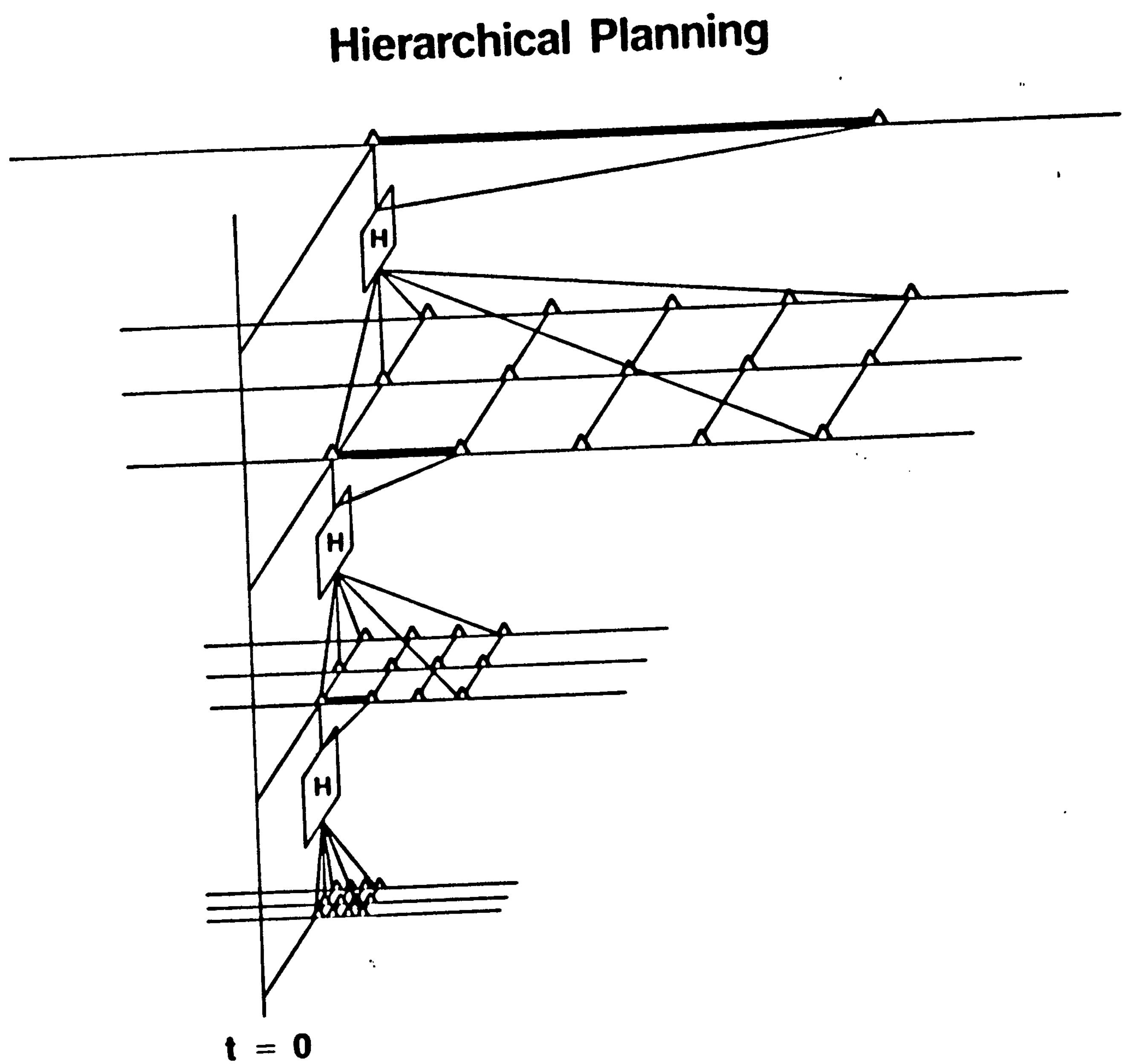

FIGURE 7: Three levels of real-time planning activity in the MAUV hierarchy.

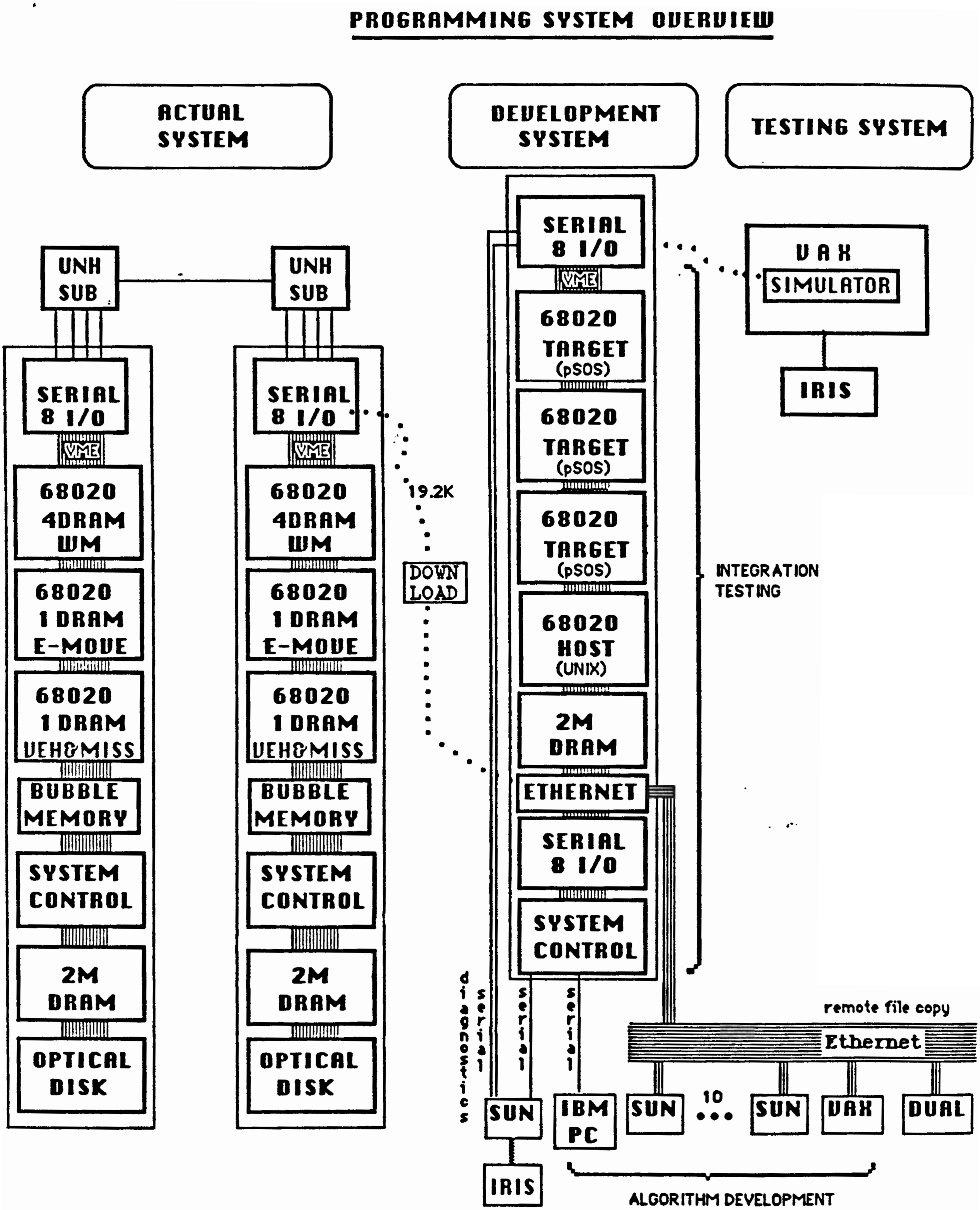

FIGURE 8: On the left is the target hardware for the two MAUV vehicles. On the right is the MAUV software development and simulation environment.