Proceedings of the SPIE 16th Annual International Symposium on Aerospace/Defense Sensing, Simulation and Controls, Orlando, FL, April 1 - 5, 2002.

4D/RCS
A Reference Model Architecture for Intelligent Unmanned Ground Vehicles

James S. Albus
Intelligent Systems Division
National Institute of Standards and Technology

ABSTRACT

4D/RCS consists of a multi-layered multi-resolutional hierarchy of computational nodes each containing elements of sensory processing (SP), world modeling (WM), value judgment (VJ), and behavior generation (BG). At the lower levels, these elements generate goal-seeking reactive behavior. At higher levels, they enable goal-defining deliberative behavior. Throughout the hierarchy, interaction between SP, WM, VJ, and BG give rise to perception, cognition, imagination, and reasoning. At low levels, range in space and time is short and resolution is high. At high levels, distance and time are long and resolution is low. This enables high-precision fast-action response over short intervals of time and space at low levels, while long-range plans and abstract concepts are being formulated over broad regions of time and space at high levels.

4D/RCS closes feedback loops at every level. SP processes focus attention (i.e., window regions of space or time), group (i.e., segment regions into entities), compute entity attributes, estimate entity state, and assign entities to classes at every level. WM processes maintain a rich and dynamic database of knowledge about the world in the form of images, maps, entities, events, and relationships at every level. Other WM processes use that knowledge to generate estimates and predictions that support perception, reasoning, and planning at every level.

4D/RCS was developed for the Army Research Laboratory Demo III program. To date, only the lower levels of the 4D/RCS architecture have been fully implemented, but the results have been extremely positive. It seems clear that the theoretical basis of 4D/RCS is sound and the architecture is capable of being extended to support much higher levels of performance.

Keywords: Intelligent control, reference model architecture, hierarchical architecture, unmanned ground vehicles, Demo III

1. INTRODUCTION

4D/RCS is a reference model architecture that provides a theoretical foundation for designing, engineering, integrating, and testing intelligent systems software for unmanned vehicle systems. 4D/RCS was developed for the Army Research Laboratory Demo III program.[1, 2]

Df: architecture
the structure of components, their relationships, and principles of design; the assignment of functions to subsystems and the specification of the interfaces between subsystems.

Df: reference model architecture
an architecture in which the entire collection of entities, relationships, and information units involved in interactions between and within subsystems are defined and modeled.

4D/RCS integrates the NIST RCS [3] with the German (Universitat der Bundeswehr Munchen) VaMoRs 4-D approach to dynamic machine vision.[4]

4D/RCS has the following properties:

1. It defines the functional elements, subsystems, interfaces, entities, relationships, and information units involved in intelligent vehicle systems.
2. It supports the selection of goals, the establishment of priorities and rules of engagement, the generation of plans, the decomposition of tasks, and the scheduling of activities; and it provides for feedback to be incorporated into control processes so that both deliberative and reactive behaviors can be combined into a single integrated system.
3. It supports the processing of signals from sensors into knowledge of situations and relationships; and it provides for the storage of knowledge in representational forms that can support reasoning, decision-making, and intelligent control.
4. It provides both static (long-term) and dynamic (short-term) means for representing the richness and abundance of knowledge necessary to describe the environment and state of a battlefield and the intelligent vehicle systems operating within it.
5. It supports the transformation of information from sensor signals into symbolic and iconic representations of objects, events, and situations, including semantic, pragmatic, and causal relationships; and it supports transformations from iconic (pictorial) to descriptive (symbolic) forms, and vice versa.
6. It supports the acquisition (or learning) of new information and the integration and consolidation of newly acquired knowledge into long-term memory.
7. It provides for the representation of values, the computation of costs and benefits, the assessment of uncertainty and risk, the evaluation of plans and behavioral results, and the optimization of control laws.

2. THE 4D/RCS ARCHITECTURE

Figure 1 is a diagram of the 4D/RCS multiresolutional hierarchy. It contains many layers of computational nodes each containing elements of sensory processing, world modeling, value judgment, and behavior generation.[5, 6]

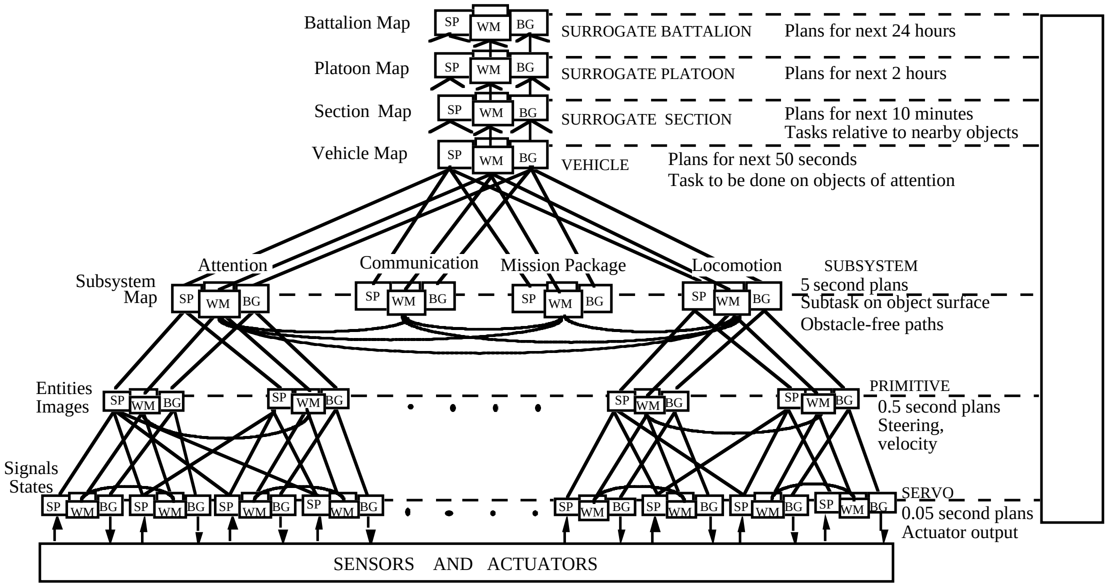

Figure 1. A 4D/RCS reference model architecture for an individual vehicle. Processing nodes are organized such that the BG modules form a command tree. Information in the KD is shared between WM modules in nodes above, below, and at the same level within the same subtree. KD modules are not shown in this figure. On the right, are examples of the functional characteristics of the BG modules at each level. On the left, are examples of the data types and maps maintained by the WM in the KD knowledge database at each level. Sensory data paths flowing up the hierarchy typically form a graph, not a tree. VJ modules are hidden behind WM modules. An operator interface provides input to, and output from, modules in every node. A control loop may be closed at every node.

At the lower levels, these elements generate goal-seeking reactive behavior. Range in space and time is short and resolution is high. This enable high precision and quick response to be achieved at low levels over short intervals of time and space. For example, at the Servo level, the reactive loop bandwidth is 200 Hz and the output to actuators is updated every 5 ms. The planning horizon at the Servo level is only 50 ms, and plans are very simple (e.g., a third order spline for a single variable.) Replanning occurs frequently (possibly every 5 ms) and the replanning delay can be as short as the reaction loop latency.

At each successively higher level, range in space and time increase by about an order of magnitude, accompanied by an order of magnitude decrease in resolution. For example, plans at the Primitive level have a time horizon of 500 ms. Output from the Primitive level is updated on 50 ms intervals, and plans may be recomputed as often as every 50 ms. At the Subsystem level, plans extend to a 5 s time horizon (or a 50 m distance, assuming vehicle speed of 10 m/s.) Output is updated and plans recomputed every 500 ms. At the Vehicle level, plans extend to a 1 min time horizon (or 500 m at 10 m/s.) Output is updated and plans recomputed every 5 sec. At the Section level, plans extend to a 10 min time horizon (or 5km at 10 m/s.) At each successively higher level, plans extend further into the future, and are recomputed less frequently. Output is also updated less frequently.

3. COMPUTATIONAL NODES

The 4D/RCS nodes have internal structure such as shown in Figure 2. Within each node there typically are four functional elements or processes: 1) behavior generation, 2) world modeling, 3) sensory processing, and 4) value judgment. There is also a knowledge database that represents the node's best estimate of the state of the world at the range and resolution that are appropriate for the behavioral decisions that are the responsibility of that node.

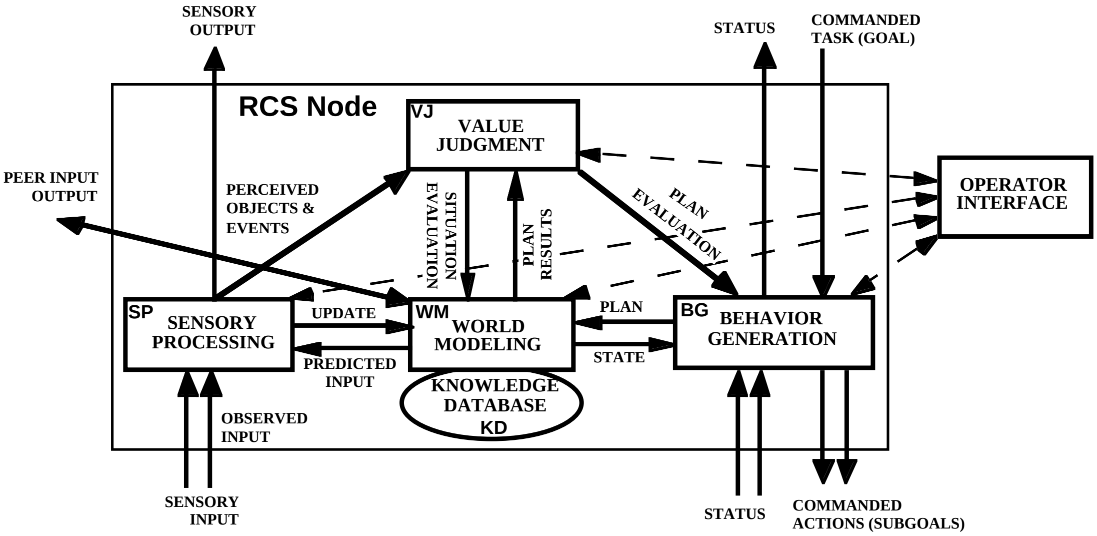

Figure 2. Internal structure of a typical 4D/RCS node. The functional elements within a RCS_NODE are behavior generation, sensory processing, world modeling, and value judgment. These are supported by a knowledge database. Each functional element in the node may have an operator interface. The connections to the Operator Interface enable a human operator to input commands, to override or modify system behavior, to perform various types of teleoperation, to switch control modes (e.g., automatic, teleoperation, single step, pause,) and to observe the values of state variables, images, maps, and entity attributes. The Operator Interface can also be used for programming, debugging, and maintenance.

As shown in Figure 3, each node of the 4D/RCS architecture is an augmented finite-state automata (fsa) that executes one cycle whenever triggered by a clock or other event. When triggered, each node reads from its input buffers, executes a state transition, calls a computational procedure (such as a real-time planner or map update function), and without waiting for the procedure to return, writes into its output buffers, stores its new state, and waits until triggered again. The rate at which the fsa is triggered defines the bandpass of the node. The fsa does not wait for called procedures to return because procedures such as planning or map updating may take longer than the interval between fsa triggers. In this case, the called procedure continues to run until it is done, at which time it writes its results into a buffer and waits to be called again by the fsa.

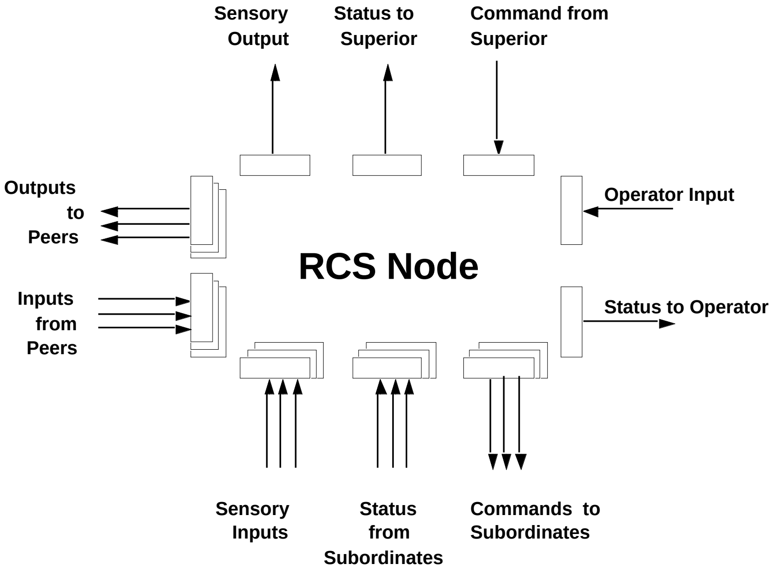

Figure 3. An RCS computational node showing input and output buffers.

All communication between nodes is in the form of messages carried by a communication process called the Neutral Messaging Language (NML.) NML establishes and maintains a set of mailboxes that allow 4D/RCS nodes to communicate with each other. NML supports a variety of communications protocols including common memory, point-to-point messaging, queuing or overwrite message delivery, and blocking or non-blocking read mechanisms. A typical NML message consists of a unique identifier, the size, and the message body. The message body can contain a data structure, such as a C struct or a C++ class or object. NML provides a mechanism for handling many different kinds of messages. It can support communicating between local modules that reside on the same computer or remote modules that reside on different computers. NML can be configured to provide communication services between modules on a single computer or multiple computers, on a single board or multiple boards. NML is very portable and has been implemented on a wide variety of operating systems including SunOS, Unix, Linux, VxWorks, and Windows NT. It can handle messages to be transmitted between different types of computers across a backplane, via a local area network, or across the Internet. NML can be configured to interface with standard military radio communications protocols such as FBCB2 (Future Battlefield Communications, Brigade and Below). A detailed discussion of how to write NML messages and configuration mailboxes is contained at *http://www.isd.mel.nist.gov/projects/rcs_lib/* along with tools for building NML channels and NML messages. Examples of how to use NML can also be found in *The RCS Handbook.*[7]

NML communications occur during the interval between when a sending fsa finishes its compute cycle and writes its output, and when the receiving fsa begins its compute cycle by reading its input. Internal communication within a node can move data at any time (i.e., on any machine instruction), not just between fsa execution cycles. Thus, high-speed frequent communication between functional elements within a node often does not use NML.

4. FUNCTIONAL ELEMENTS

4D/RCS nodes contain functional elements of behavior generation, world modeling, sensory processing, and value judgment.

Df: behavior generation
is the functional element that plans and controls actions designed to achieve behavioral goals.

Df: behavioral goal
a desired state that a behavior is designed to achieve or maintain

Behavior generation accepts task commands (with goals and priorities,) formulates and/or selects plans, and controls action. Behavior generation uses a priori task knowledge and value judgment functions combined with real-time information to find the best assignment of tools and resources to agents, and to find the best schedule of actions (i.e., the most efficient plan to get from an anticipated starting state to a goal state). Behavior generation controls action by both feedforward actions and by feedback error compensation.

Df: world modeling
a functional element that builds, maintains, and uses a world model to support behavior generation and sensory processing

Df: world model
an internal representation of the world

The world model may include models of objects, events, classes, tasks, and agents; models of portions of the environment, and a model of the intelligent system itself. The world model includes all of the knowledge that is distributed throughout all the nodes in the 4D/RCS architecture.

Df: knowledge database
the data structures and the information content that collectively form the intelligent system's world model.

The knowledge database has three parts:

(1) A long term memory containing symbolic and iconic representations of all the generic and specific objects, events, and rules that are known to the intelligent system.

(2) A short term memory containing iconic and symbolic representations of geometric entities and events that are the subject of current attention.

(3) An instantaneous dynamic representation of immediate experience consisting of current sensor signals and values of observed, estimated, and predicted attributes and state variables.

Df: sensory processing
a set of processes by which sensory data interacts with a priori knowledge to detect or recognize useful information about the world.

Sensory processing accepts signals from sensors that measure properties of the external world or conditions internal to the system itself. In general, sensors do not directly measure the state of the world. Sensors only measure phenomena that depend on the state of the world. Signals generated by sensors may be affected by control actions that cause the sensors to move through the world. Sensor output signals are also corrupted by noise.

Sensory processing scales, windows, groups, segments, and filters data, computes observed features and attributes, and compares them with predictions from internal models. Correlations between sensed observations and internally generated expectations are used to detect and classify entities, events, and situations. Differences between sensed observations and internally generated predictions are used to update the knowledge database. Sensory processing also computes attributes of entities and events, and clusters or groups recognized entities and detected events into higher order entities and events.

Df: value judgment
a process that:
    computes cost, risk, and benefit of actions and plans,
    estimates the importance and value of objects, events, and situations,
    assesses the reliability of information,
    calculates the rewarding or punishing effects of perceived states and events.

Value judgment evaluates perceived and planned situations, thereby enabling behavior generation to select goals and set priorities. It computes what is important (for attention), and what is rewarding or punishing (for learning). It assigns values to recognized objects and events, and computes confidence factors for observed, estimated, and predicted attributes and states.

5. DELIBERATIVE AND REACTIVE

The 4D/RCS is a hybrid architecture that behaves both deliberatively (with the hierarchical ability to plan) and reactively (with the autonomic ability to respond rapidly to exigencies). At every level within the BG hierarchy, planning processes receive goals and priorities from superiors and deliberatively decomposes those goals into subgoals with priorities and timing requirements for subordinates at levels below. Also at every level, executor processes close a feedback loop that produces reactive behavior.

At every level within the SP hierarchy, perception processes focus attention, segment, filter, and classify information derived from subordinate levels. Events are detected, objects recognized, situations analyzed, and status reported to superiors at the next higher level.

At every level, sensory processing and behavior generation processes have access to a model of the world that is resident in a knowledge database. This world model enables the intelligent system to analyze the past, plan for the future, and perceive sensory information in the context of expectations.

At every level, a set of cost functions enable value judgments and determine priorities that support intelligent decision making, planning, and situation analysis. This provides a robust form of value driven behavior that can function effectively in an environment filled with uncertainties and unwanted signals.

6. A MILITARY COMMAND AND CONTROL EXAMPLE

At the top level of any military hierarchy, there is a human commander supported by a staff that provides intelligence and decision support functions. This is where high-level strategy is defined and strategic goals are established. The highest level commander formulates strategy that defines what kind of operations will be conducted, what rules of engagement will be followed, and what values will determine priorities and shape tactical decisions. Strategic goals are decomposed through a chain of command that consists of operational units made up of intelligent agents (humans or machines), each of which possesses a particular combination of knowledge, skills, and abilities, and each of which has a well-defined set of duties and responsibilities. Each operational unit accepts tasks from a higher level unit and issues sub-tasks to subordinate units. Within each operational unit, intelligent agents are given job assignments and allocated resources with which to carry out their assignments. The intelligent agents then schedule their activities so as to achieve the goals of the jobs assigned to them. Each agent is expected to make local executive decisions to achieve goals on schedule by solving local problems and compensating for local unexpected events. Within a unit, each agent acts as a member of a team in planning and coordinating with peers at the same level, while simultaneously acting as the commander of a subordinate unit at the next lower level.

Each agent, in each operational unit, has knowledge of the world environment in which it must function. This knowledge includes state variables, maps, images, and symbolic descriptions of the state of the world. It also includes entities, events, patterns, and situations that exist in the environment, their attributes and relationships. Knowledge is kept current and accurate through sensors that detect events and sensory processing systems that compute attributes of objects and situations in the world. Knowledge of the world includes laws of nature that describe how the environment behaves under various conditions, as well as values and cost functions that can be used to evaluate the state of the world and the performance of the intelligent control system itself.

At the bottom of the hierarchy, the system performs physical actions (e.g., the movement of effectors such as wheels, arms, legs, or propellers) that affect the environment, while its sensors measure phenomena – including the effects of the system itself – in the environment. The process is a continuous feedback loop of the environment affecting the robotic system and the robotic system affecting the environment.

For any chain of command, there is an organizational configuration that describes functional groupings and defines who reports to whom. However, organizational charts typically do not show all the communication pathways by which information flows throughout the organization. Much information flows horizontally between agents and operational units, through both formal and informal channels. Multiple agents within operational units share knowledge about objects and events in the world, and status of other agents. For example, agents operating on the battlefield often can see each other and may respond to requests for help from peers without explicit orders from superiors. Also, plans developed in one operational unit may be communicated to other units for implementation.

The 4D/RCS explicitly allows for the exchange of information between organizational units and agents at the same level or different levels. Commands and status reports flow only between supervisor and subordinates, but queries, replies, requests, and broadcasting of information – by posting in common memory or by messaging mechanisms – may be used to convey information between any of the units or agents in the entire 4D/RCS architecture.

The 4D/RCS organizational chain of command is defined by the duties and responsibilities of the various organizational units and agents and by the flow of commands and status reports between them, not by access to information or the ability to communicate. This means that while the relationships between supervisors and subordinates has the form of a tree, the exchange of information between units and agents is a graph that, in principle, could be fully connected. In practice, however, the communication network is not typically fully connected because many of the units and agents simply have nothing to say to each other.

7. THE DEMO III EXPERIENCE

4D/RCS has been implemented in part on the Demo III experimental unmanned vehicle (XUV) program and related research.[8, 9, 10, 11] Three levels (Servo, Primitive, and Autonomous Mobility Subsystem) of the 4D/RCS behavior generation, world modeling, and value judgment processes were more or less fully implemented on the Demo III XUVs, and the first two levels of sensory processing were partially implemented. Even this partial implementation enabled performance of the Demo III vehicles that was extremely impressive. The vehicles were regularly able to traverse more than a kilometer of challenging terrain including dirt roads and trails, woods and fields, hills and valleys, filled with tall grass, weeds, stumps, fallen trees, and brush – without human intervention. Occasionally, the vehicles had to be stopped to prevent a dangerous situation, or because they became hopelessly trapped by difficult terrain features. But more often, they were able to keep out of trouble, and accomplish their mission of reconnaissance, surveillance, and target acquisition. They were usually able to avoid stumps and logs, climb up and down steep hills, and deal with standing water and mud as well as, and in some cases better than, human drivers. Particularly at night, the XUVs were able to traverse difficult terrain faster and more reliably than human drivers.

The Demo III experience has validated the basic principles of the 4D/RCS reference model architecture. The architecture has proven to be modular, upgradable, scalable, and easy to use. As additional features of 4D/RCS are implemented, the performance of autonomous vehicles is expected to improve dramatically. For example, when Vehicle level behavior generation algorithms are installed, the vehicle will be able to register a priori maps with sensor data and execute tactical behaviors. When the Section level is installed, the vehicle will be able to conduct cooperative tactical behaviors with other autonomous or manned vehicles. When additional levels of sensory processing are installed, the vehicle will be able to track targets, recognize a large number of objects, and respond with tactically relevant behaviors.

The Demo III experience also pointed up some areas that need improvement. For example, the current LADAR range imaging camera is not able to see through clouds of dust or smoke. This causes the XUV to treat a cloud of dust or smoke as if it were a solid obstacle. An interim solution is to stop and wait for the dust to clear. A long range solution is to enable the LADAR to see objects beyond the first detected return signal. The LADAR also needs more range and higher resolution. These are problems that can be expected to yield to further engineering development in the near future.

The November 2001 performance of the Demo III vehicles at Fort Indiantown Gap far exceeded the expectations of most military observers.[12] As a result of this demonstration, several general officers revised downward their estimate of how long it will be before autonomous vehicles can perform useful military tasks. It now is widely anticipated that autonomous vehicles will be capable of useful military missions in laying smoke, delivering supplies, serving as pack mules, and acting as forward observers for scouts by the year 2010.

8. ACKNOWLEDGEMENTS

This work was supported by the U. S. Army Research Lab, Charles Shoemaker, Program Manager. This paper is a product of government funding and is therefore not subject to copyright.

9. REFERENCES

1. J. A. Bornstein, “U.S. Army ground robotics research program”, *Proceedings of SPIE Aerosense Conference Vol. 4715,* Orlando, Florida, 1-5 April, 2002.
2. C. Shoemaker, J. Bornstein, S. Myers, and B. Brendle, “Demo III: Department of Defense testbed for unmanned ground mobility”, *Proceedings of SPIE Vol. 3693, AeroSense Session on Unmanned Ground Vehicle Technology*, Orlando, Florida, April 7-8, 1999.
3. J. S. Albus and A. M. Meystel, *Engineering of Mind: An Introduction to the Science of Intelligent Systems*, John Wiley & Sons, New York, 2001.
4. E. D. Dickmanns, et. al., “The Seeing Passenger Car ‘VaMoRs-P’,” *International Symposium on Intelligent Vehicles’94*, Paris, October 24-26, 1994.
5. J. S. Albus, “4-D/RCS Reference Model Architecture for Unmanned Ground Vehicles”, *Proceedings of SPIE Vol. 3693, AeroSense Session on Unmanned Ground Vehicle Technology*, Orlando, Florida, April 7-8, 1999.
6. A. M. Meystel and J. S. Albus, *Intelligent Systems: Architecture, Design, and Control*, John Wiley & Sons, New York, 2002.
7. V. Gazi, M. Moore, K. Passino, W. Shackleford, F. Proctor, J. Albus, *The RCS Handbook: Tools for Real-Time Control Systems Software Development*, John Wiley & Sons, NY, 2001.
8. T. H. Hong, S. Balakirsky, E. Messina, and M. Shneier, “A Hierarchical World Model for an Autonomous Scout Vehicle”, *Proceedings of SPIE Aerosense Conference Vol. 4715,* Orlando, Florida, 1-5 April, 2002.
9. T. H. Hong, T. Chang, C. Rasmussen, and M. Shneier, “Road Detection and Tracking for Autonomous Mobile Robots”, *Proceedings of SPIE Aerosense Conference Vol. 4715,* Orlando, Florida, 1-5 April, 2002.
10. A. Lacaze, “Hierarchical Real-Time Path Planning for Obstacle Avoidance and Path Optimization for One or More Autonomous Vehicles, *Proceedings of SPIE Aerosense Conference Vol. 4715,* Orlando, Florida, 1-5 April, 2002.
11. S. Balakirsky and A. Lacaze, “Value-Driven Behavior Generation for an Autonomous Mobile Ground Robot”, *Proceedings of SPIE Aerosense Conference Vol. 4715,* Orlando, Florida, 1-5 April, 2002.
12. Video CD of Demo III demonstration at Fort Indiantown Gap, available from Army Research Laboratory Demo III Program Office, Attn. AMSRL-WM-RP, Aberdeen Proving Ground, MD, November, 2001.