Fuzzy Behavior Hierarchies for Multi-Robot Control

Edward Tunstel,[1,*] Marco A. A. de Oliveira,[2] Sigal Berman[3]
[1] NASA Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109
[2] NASA Center for Autonomous Control Engineering, University of New Mexico, Albuquerque, NM 87131
[3] Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel

Hierarchical approaches and methodologies are commonly used for control system design and synthesis. Well-known model-based techniques are often applied to solve problems of complex and large-scale control systems. The general philosophy of decomposing control problems into modular and more manageable subsystem control problems applies equally to the growing domain of intelligent and autonomous systems. However, for this class of systems, new techniques for subsystem coordination and overall system control are often required. This article presents an approach to hierarchical control design and synthesis for the case where the collection of subsystems is comprised of fuzzy logic controllers and fuzzy knowledge-based decision systems. The approach is used to implement hierarchical behavior-based controllers for autonomous navigation of one or more mobile robots. Theoretical details of the approach are presented, followed by discussions of practical design and implementation issues. Example implementations realized on various physical mobile robots are described to demonstrate how the techniques may be applied in practical applications involving homogeneous and heterogeneous robot teams. © 2002 Wiley Periodicals, Inc.

1. INTRODUCTION

Autonomous control and navigation of mobile robotic vehicles are fundamental enabling technologies for automation in a variety of operating domains ranging from industrial environments to remote planetary surfaces. The engineering problem to be solved generally consists of achieving real-time sensor-based motion control among obstacles in the environment while performing useful tasks throughout its accessible regions. In many instances, mobile robots are required to do so using limited resources (e.g., power, computation, sensors, etc.) that are resident onboard the vehicle. Recent successful approaches have been based on a behavioral decomposition of tasks with quasi-parallel execution. The seminal work by Brooks,[1] exemplified

*To whom all correspondence should be addressed.

INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS, VOL. 17, 449–470 (2002)
© 2002 Wiley Periodicals, Inc.
Published online in Wiley InterScience (www.interscience.wiley.com). • DOI: 10.1002/int.10032

by the subsumption architecture, provided the impetus for these behavior-based solutions. Since the introduction of the subsumption architecture, a number of variants have been proposed for behavior-based control including hierarchical architectures of fuzzy rule-based systems.[2,3] Related work has been proposed in the literature on single mobile robot control using multiple behaviors and/or hierarchical control structures.[4,5,6] The ideologies of each of these research activities are mutually similar. The approach presented in this article differs mainly in its use of a fuzzy-behavior hierarchy, and the implementation of a dynamic behavior coordination mechanism as a source of adaptive behavior. It can generally be applied as a control methodology for distributed intelligent systems that can be represented as hierarchical or decentralized computational structures. Examples of such systems are autonomous robotic agents, corporate decision-making entities, social systems, electric power systems, and other large-scale systems[7] in general.

This article presents some theoretical details of a hierarchical fuzzy behavior-based control architecture that has been successfully applied to autonomous mobile robot navigation problems. The theoretical foundation permits design flexibility and extensibility of the architecture for application to increasingly complex problems. As such, the approach may be used to implement hierarchical controllers for individual mobile robot navigation, and is easily extended to serve as a core architecture for coordination and control of multi-robot teams.[9,10] Architectures that integrate fuzzy logic techniques with behavior-based control can offer robust and reliable solutions for the autonomous navigation problem. In a fuzzy-behavior control hybrid, additional advantages are gained with regard to representation and handling of uncertain and imprecise knowledge about the robots' environment, and in behavior coordination and conflict resolution. Fuzzy control implicitly accounts for uncertainty by virtue of the approximate reasoning capability of fuzzy logic. In general, fuzzy logic controllers (FLCs) provide robustness to perturbations, design simplicity, and efficiency in dealing with continuous variables.[8] These are all desirable attributes for autonomous vehicle control systems.

Example implementations of the architecture on various physical mobile robots are briefly described, which demonstrate how the techniques may be applied in practical applications involving homogeneous and heterogeneous robot teams. The article is organized as follows. Section 2 details the underlying theory of hierarchical fuzzy-behavior synthesis and coordination. This is followed by discussion of practical issues related to hierarchy design and implementation for individual mobile robots in Section 3. Sections 4 and 5 describe the application of the approach in the context of multi-robot control, followed by example implementations for physical robots in Section 6.

2. FUZZY BEHAVIOR HIERARCHIES

Behavior controllers that are based on fuzzy rule-based systems can be configured in a number of ways. The alternatives are governed by issues such as the fuzzy set resolution selected for system variables and the complexity of decision-making, or reasoning, demanded by the task environment. The fuzzy set resolution of the system variables (inputs) determines the total number of rules (i.e., the rule-base cardinality) necessary to cover all possible combinations of fuzzy controller inputs. The collection of individual rule outputs for a given rule base produces a control/decision surface, the nonlinearity of which is a measure of the decision-making complexity. Thus, the resolution of the state space and the nonlinearity of the control surface are interrelated with regard to the interpolation necessary to produce desired behavior via approximate reasoning. If the rule-base cardinality is relatively small, then it is feasible to realize the fuzzy behavior controller as a monolithic, or single-rule-base controller. Otherwise, alternative rule structures may be in order. Hierarchical rule structures are a viable alternative for dealing with autonomous behavioral systems such as mobile robots, which require many rules and/or complex decision-making.

The architectural design of a hierarchical system of behaviors is based on the premise that autonomous navigation behavior can be decomposed into a finite number of special-purpose task-achieving behaviors.[1] Behaviors can be arranged as a hierarchical network of distributed fuzzy rule bases, each responsible for some integral aspect of system functionality. A collection of *primitive* behaviors resides at the lowest level, which is referred to as the primitive level. These are encoded as fuzzy rule bases with distinct control policies governed by fuzzy inference. They are typically simple and self-contained behaviors that serve a single purpose while operating in a reactive (non-deliberative) or reflexive (memoryless) fashion. Examples include simple obstacle avoidance and motion towards commanded subgoals. Primitive behaviors perform mappings from different subsets of the available sensor suite to common actuators. When operating alone, each would be insufficient for performing complex navigation tasks. Such primitive behaviors are building blocks for higher-level coordination behaviors, referred to as *composite behaviors,* such as goal-seeking or route-following.[11] That is, their capabilities can be combined through synergistic coordination to produce composite behavior(s) suitable for goal-directed navigation. Hereafter, references to primitive and composite fuzzy behaviors will be abbreviated as *p-behaviors* and *c-behaviors,* respectively.

The hierarchical architecture differs from the conventional monolithic FLC implementation in that a multi-level structure of fuzzy rule bases is employed and a mechanism for adaptive behavior is provided. The behavior of the system is adaptive in the sense that the control surface generated by the hierarchy constantly changes in response to sensor input and perception of the environment. The control surface of the conventional monolithic FLC is usually fixed and represented by a static nonlinear input-output map. Note, however, that an FLC with a mechanism for adaptive behavior should not be mistaken for an adaptive controller. The fuzzy control hierarchy assumes the role of an intelligent supervisory controller over conventional linear controllers. That is, the hierarchy generates control set-points as input to low-level motor controllers in support of autonomous local navigation. Figure 1 is a conceptual illustration of a general behavior hierarchy consisting of a primitive level of individual motion behaviors, β_i, coordinated by higher-level c-behaviors, B_j, via a weight-adaptive scheme called behavior modulation (described in Section 2.1.1). The interconnecting circles between c-behaviors and the primitive level represent weights and activation thresholds associated with p-behaviors. Each p-behavior maps inputs to a vector of fuzzy control outputs. Higher-level behaviors act as fuzzy

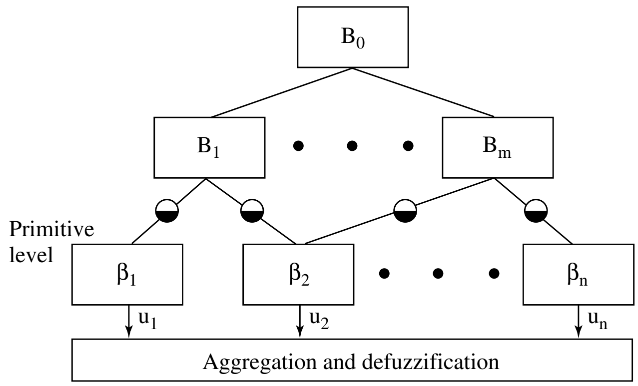

Figure 1. Generic fuzzy behavior hierarchy.

decision systems that map goal information and other available input (which may include sensory data, memory, and symbolic knowledge) to dynamically adaptive, scalar weights associated with each p-behavior.

2.1. Theoretical Framework

Let X and U be input and output universes of discourse of a p-behavior with a rule base of size N. We describe the generic fuzzy if-then rule as follows

$$\text{IF } x \text{ is } \tilde{A}_i \quad \text{THEN } u \text{ is } \tilde{B}_i \tag{1}$$

where x and u represent input and output fuzzy linguistic variables, respectively, and $\tilde{A}_i$ and $\tilde{B}_i$ $(i = 1, 2, \ldots, N)$ are fuzzy subsets representing linguistic values of x and u. In our mobile robot controllers, x often refers to sensory data or goal information; u refers to set-points for motor control velocities. In general, the rule antecedent consisting of the proposition "x is $\tilde{A}_i$" could be replaced by a compound fuzzy proposition consisting of a conjunction (and/or disjunction) of similar propositions. Similarly, the rule consequent "u is $\tilde{B}_i$" could include additional rule-base output propositions. Primitive fuzzy behaviors are synthesized as a finite set of such rules. Formally, the output of the ith fuzzy rule is represented by a fuzzy relation, $\tilde{u}_i \in X \times U$, which is a fuzzy set itself. Moreover, the output of a fuzzy rule base can be characterized as a single fuzzy relation, $\tilde{\beta}$, which is a union of fuzzy relations $\tilde{u}_i, i = 1, 2, \ldots, N$:

$$\tilde{\beta} = \bigcup_{i=1}^{N} \tilde{u}_i \tag{2}$$

The output of a primitive fuzzy behavior, then, can also be represented as a fuzzy set. Thus, the mathematical operations of inference in fuzzy controllers are closed for fuzzy sets. This fact serves as the basis for extending fuzzy set and logic operations used for monolithic fuzzy control to multi-rule-based hierarchical fuzzy control.

In monolithic FLCs, Equation 2 represents the aggregated result of individual rule outputs and undergoes defuzzification to yield a crisp (numerically precise) output that serves as input to the robot. In a similar manner, outputs from multiple p-behaviors are aggregated to yield a resultant fuzzy set as the output of the overall behavior hierarchy. In order for this to work effectively, defuzzification of p-behavior outputs must be deferred until after the aggregation takes place. Therefore, in the fuzzy-behavior hierarchy the output of each p-behavior is a fuzzy set of recommended wheel velocities. Alternatively, the output fuzzy sets could represent recommended vehicle steering and speed, depending upon one's selection of control variables.

2.1.1. Behavior Modulation

When more than one p-behavior is active, interactions can take the form of behavioral cooperation and/or competition. These forms of behavior are not perfectly distinct; they are extremes along a continuum.[12] Variations in the state of interaction throughout this continuum are governed by *behavior modulation,* which we define as continuous adjustment or adaptation of behavior activation levels in a multi-behavior or multi-agent system. This is the underlying mechanism used to regulate behavior coordination in the architecture. It is achieved by weighted control decision-making embodied in a concept called the *degree of applicability* (DOA)—a measure of the instantaneous level of activation of a motion behavior. Fuzzy rules of c-behaviors are formulated to include weighting consequents that modulate the DOAs of behaviors at a lower level. We refer to these as applicability rules. The DOA, α, of a p-behavior is specified in the consequent of applicability rules of the form

$$\text{IF } x \text{ is } \tilde{A}_i \quad \text{THEN } \alpha \text{ is } \tilde{D}_i \tag{3}$$

where $\tilde{A}_i$ is defined as in Equation 1. $\tilde{D}_i$ is a fuzzy subset representing the linguistic value (e.g., *high*, *low*, etc.) of the behavior's DOA to the situation prevailing during the current control cycle. It is defined over the closed unit interval [0, 1], and defaults to zero if unspecified by a c-behavior. In general, a composite behavior, c, will include applicability rules with a consequent of the form "α_p is $\tilde{D}_i$" for each primitive behavior p modulated by c. Thus for all p, $\alpha_p \in [0, 1]$ is determined by fuzzy inference as the output of an associated composite behavior. This feature allows certain robot behaviors to influence the overall behavior to a greater or lesser degree as required by the current situation and goal. It serves as a form of adaptation since it causes the control policy to dynamically change in response to goal information and sensory input. The behavior hierarchy, then, is a dynamic nonlinear mapping from situations to actions rather than a static nonlinear mapping represented by a fixed set of fuzzy rules.

We have established Equation 2 as an expression for the output fuzzy set for motor control set-points recommended by a p-behavior. Let us denote the fuzzy output for primitive behavior p as $\tilde{\beta}_p$, and its corresponding DOA as α_p. Let P be the set of all primitive behaviors in a given behavior hierarchy. Then the modulated fuzzy output of p is given by $(\alpha_p \cdot \tilde{\beta}_p)$. At this point the use of an appropriate t-conorm will take care of aggregating individual modulated fuzzy outputs to produce a resultant output of the behavior hierarchy. The arithmetic sum t-conorm, and hence, center-of-sums defuzzification Ref. 13, has been chosen for this purpose. This is an instance of the *weight-counting* property described in Ref. 13. The arithmetic sum will be denoted here by the symbol $\uplus$. Finally, if we denote the resultant output fuzzy set of the behavior hierarchy as $\tilde{\beta}_H$, then its computation is performed using the following fuzzy-behavior hierarchy equation:

$$\tilde{\beta}_H = \biguplus_{p \in P} \alpha_p \cdot \tilde{\beta}_p \tag{4}$$

The crisp control output, $u^* \in U$, which serves as the velocity set-point input to the robots's wheel motors, is computed by center-of-sums defuzzification of $\tilde{\beta}_H$. That is,

$$u^* = \frac{\int_{u \in U} u \cdot \mu_{\tilde{\beta}_H}(u)}{\int_{u \in U} \mu_{\tilde{\beta}_H}(u)} \tag{5}$$

$$= \frac{\int_{u \in U} u \sum_{p \in P} \alpha_p \cdot \mu_{\tilde{\beta}_p}(u)}{\int_{u \in U} \sum_{p \in P} \alpha_p \cdot \mu_{\tilde{\beta}_p}(u)} \tag{6}$$

This expression is the nonlinear input-output mapping of the fuzzy-behavior hierarchy, which causes its control surface to dynamically change due to continuous fluctuations in α_p, $\forall p \in P$. In this procedure, multiplication by α_p expresses the relative *applicability* of a p-behavior to the current situation, while the scalar α_p itself represents the *weight* of the behavior in the aggregated control decision. Operators other than multiplication can be used to achieve a similar effect. Yager[14] refers to such operators as importance transformations and suggests a general class of them for both t-norm and t-conorm aggregations.

To gain a better understanding of the effect of weight counting, consider two primitive behavior output fuzzy sets (modulated or not), $\tilde{\beta}_1$ and $\tilde{\beta}_2$. Suppose that when these output recommendations are aggregated, the union, $\tilde{\beta}_1 \cup \tilde{\beta}_2$, results in an overlap of portions of $\tilde{\beta}_1$ and $\tilde{\beta}_2$. Weight-counting defuzzification methods consider membership values in the overlap region contributed by both $\tilde{\beta}_1$ and $\tilde{\beta}_2$, thereby processing the full recommendations of both behaviors and forming a true consensus. This also holds for greater than two output fuzzy sets. The weight-counting property enforces the *weighted* decision-making intended in the philosophy of our approach; i.e., control actions should result from a consensus of recommendations from all behaviors applicable in the current context. The arithmetic sum, as an aggregation operator, affords a behavior arbitration strategy that retains and uses all available information from the individual output fuzzy sets.

3. DESIGN CONSIDERATIONS AND PRACTICAL ISSUES

A number of practical issues serve to complicate the design and development of the necessary autonomous control algorithms. First, each vehicle is required to achieve the desired functionality within the data processing limitations of the computational resources available onboard. Second, sensors commonly used in mobile robotics to measure range to objects in the workspace have inherent accuracy and reliability problems that are well known in the literature.[15] As a result, the quality of range data operated on by an associated robot controller is suspect. Third, several sources of error and uncertainty contribute to inaccurate dead reckoning. Systematic errors caused by unequal wheel diameters, uncertainty about the vehicle's wheel-base, or other mechanical imperfections are common.[16] In addition, non-systematic errors such as irregularities in the terrain contribute significantly to the problem of estimating the vehicle position and orientation. The problem is further complicated by unpredictable effects due to less-than-perfect execution of actuator commands (e.g., wheel slip).

For mobile robots that are designed primarily for path tracking or teleoperation, classic control techniques can be applied without much difficulty. If there is sufficient reason to employ fuzzy control techniques for these problems, monolithic FLCs often provide sufficient solutions. However, as environmental structure and task constraints are removed from the problem domain, the need for increased autonomy mandates the development of higher-level intelligent controllers. The methodology detailed in the previous section can be used to synthesize such controllers using a well-managed hierarchical structure of fuzzy behaviors. Fuzzy logic is particularly well suited for implementing such controllers due to its capabilities of inference and approximate reasoning under uncertainty. In the context of mobile robotics, fuzzy logic techniques permit easy modeling of the intuitive nature of sensor-based navigation through the use of fuzzy sets and linguistic terminology. There are certain aspects of non-fuzzy solutions to the problem for which the application of fuzzy techniques is more germane. These include conflict resolution, cooperation among alternatives, and behavior coordination. The proposed architecture features fuzzy implementations of these capabilities. Advantages to be gained include generally smoother control response in transitions from one activated behavior to others, and more democratic arbitration with less loss of available information. This is not to mention the "built-in" capacity for handling uncertainties inherent in perception and actuation for mobile robot controllers.

At a higher level, solutions to complex navigational problems must handle the fulfillment of multiple, and sometimes conflicting, goals for which the priorities might change with time. Thus, a control system capable of realizing a number of task-achieving behaviors that can be integrated to achieve different control objectives is desirable. In the face of these formidable practical concerns, solutions for autonomous navigation must be both robust and adaptable. The requirements for developing a system that deals with multiple objectives subject to imprecise sensory data and uncertain actuator performance lead us to employ approximate reasoning capabilities as a resource for autonomy. We advocate fuzzy logic control techniques as a means to this end because they facilitate integration of the various layers of hierarchical control systems. Software control architectures that employ fuzzy logic techniques and behavioral subsystems can mitigate complications associated with developing a suitable interface between high-level and low-level processes. This is primarily due to the intrinsic capability of fuzzy logic to represent both numerical and symbolic aspects of reasoning (in the form of analytical membership functions and rules expressed in natural language, respectively). As such, fuzzy systems are a powerful tool for addressing the integration of deliberative and reactive control layers in hierarchical software architectures for distributed robot coordination. Furthermore, fuzzy control structures are amenable to integration with other soft computing methodologies such as neural networks and evolutionary algorithms. Fuzzy systems also provide significant reasoning capability that can be realized with compact code and memory sizes. In contrast to traditional production rules in AI expert systems, the symbolic expressions of if-then rules in fuzzy systems are manipulated through simple numerical computations based on fuzzy set-theoretical operations. Since fuzzy sets and fuzzy logic are generalizations of classic sets and logic theory, quantitative references to physical attributes can be represented with coarse granularity. As such, significantly fewer fuzzy if-then rules (relative to rule base sizes for expert systems) can be formulated to represent general situations, while the interpolative mechanics of fuzzy logical inference handle intelligent decision-making for specific situations that arise. In contrast, a conventional expert system requires a rule for each possible situation.

3.1. Simulation Example

To provide insight into the inner workings of the control strategy, we present an example behavior hierarchy used to control a single mobile robot in a simulated indoor navigational task. The capabilities necessary for goal-directed navigation include avoidance of collisions with obstacles, self and goal localization, and traversal through indoor features such as halls, doorways, and densely cluttered spaces. A behavior hierarchy encompassing some of these capabilities is shown in Figure 2. This figure implies that goal-directed navigation can be decomposed as a behavioral function of *goal-seek* (collision-free navigation to some location) and *route-follow* (assuming some direction is given perhaps in the form of waypoints or a path plan). These behaviors can be further decomposed into the p-behaviors shown, with dependencies indicated by the adjoining lines. The purposes of *wall-follow* and *avoid-collision* are implied by their names. The *doorway* behavior guides a robot through narrow passageways in walls. The *go-to-xy* behavior performs a position-based homing motion that directs a robot to navigate along a straight-line

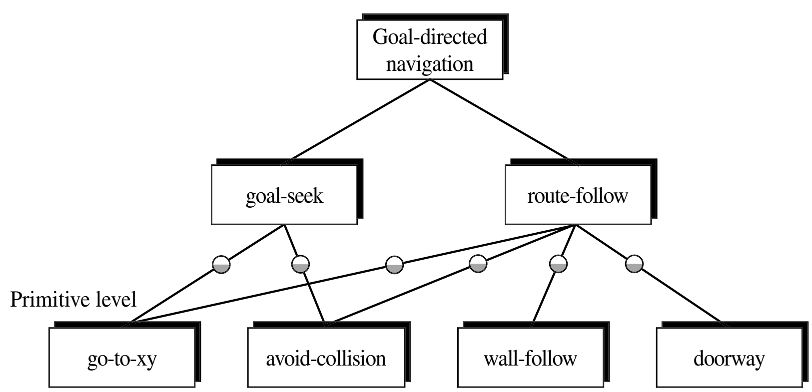

Figure 2. Mobile robot behavior hierarchy.

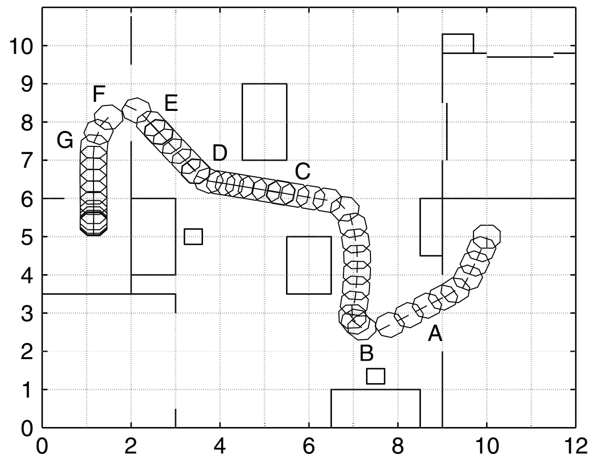

Figure 3. Route-following using waypoints.

trajectory to a particular location. Interconnecting circles between c-behaviors and the primitive level represent DOAs of associated p-behaviors.

In this hierarchy, *route-follow* employs capabilities of several primitive behaviors, each contributing its own capability while being modulated in response to sensor input and goal information. For this example, the robot's task is sensor-based navigation from a start location to a designated goal inside a hypothetical indoor layout (See Figure 3). The robot is not provided with an explicit map, however, it is cognizant of the notion of a two-dimensional Cartesian coordinate system. Its path is not preplanned; it is executed in response to instantaneous sensory feedback via modulation of *avoid-collision* (*ac*), *go-to-xy* (*gt*), and *wall-follow* (*wf*). Its initial state is at a docking/charging station with pose $(x \;\; y \;\; \theta)^T = (10\,\text{m} \;\; 5\,\text{m} \;\; -\frac{\pi}{2}\,\text{rad})^T$; the goal is located at (1.2 m, 5.2 m). A designated route to the goal is specified by the following three additional waypoints or subgoals: (7.5, 2.5) → (3.5, 6.5) → (2.0, 8.5). The resulting route is shown in Figure 3 and the corresponding DOAs (α_p) for each p-behavior are shown separately in Figure 4 (labeled DOA_p on the ordinate of each graph). Labels *A*–*G* in each figure indicate a correlation between robot position along the route and the DOAs applied at that instant.

At point *A* as the robot exits the start room, all three primitive behaviors compete for control. At *B*, α_{ac} exceeds α_{gt} and α_{wf}, allowing *avoid-collision* to take over as the dominant behavior while approaching the first waypoint. After avoiding an obstacle, α_{gt} increases while *go-to-xy* becomes dominant at *C* on approaching the second waypoint. Dominance alternates between *avoid-collision* and *go-to-xy* as they compete while traversing through *D* and *E* where the robot adjusts its heading towards the goal room. Interactions among the three p-behaviors resurface at *F* where α_{wf} exceeds α_{ac} and α_{gt}, allowing *wall-follow* to briefly dominate. It becomes inactive at *G*, giving way to *avoid-collision*, and finally to *go-to-xy* on direct approach to the goal. During the majority of the task each p-behavior is active to varying degrees influencing the overall robot behavior.

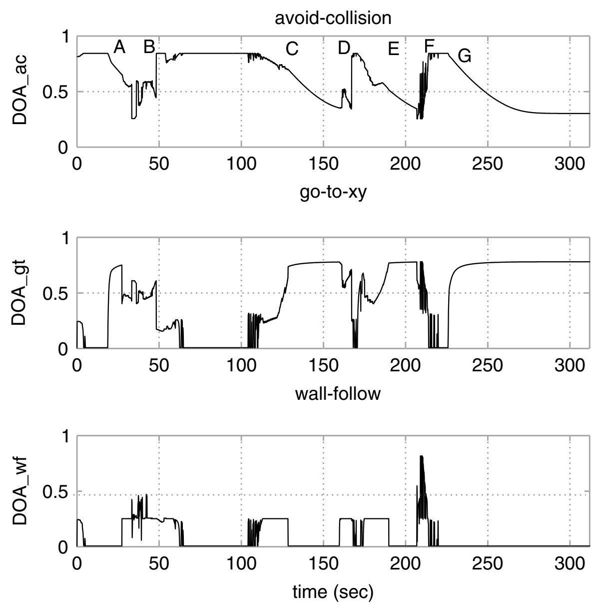

Figure 4. Behavior modulation.

4. APPLICATION TO MULTI-ROBOT SYSTEMS

In order to extend the approach to be viable in multi-robot domains, additional considerations are necessary to address the new demands and challenges on the control system.[17] These include efficient communication for data sharing and robot cooperation, scalable cooperation strategies, and efficient sensor fusion and data representation. The extensibility of the architecture facilitates the straightforward incorporation of the necessary features. In this section, we describe the *core* fuzzy behavior hierarchy developed for multi-robot systems and its associated design methodology.

In the design phase, the generic hierarchy of Figure 1 is outfitted with competencies that span the necessary behavioral repertoire for fulfilling the robot's purpose. That is, a behavior hierarchy is constructed that includes competencies sufficient to support the robot's task(s), and that are executable in the target operating environment. Sufficiency of the behavioral repertoire is based on the designer's subjective assessment of the problem.

The design procedure begins with establishment of the overall behavioral purpose of the desired navigation system. The problem is then assessed from the point of view of the robot(s) situated in the target operating environment. The behavior hierarchy is built to encompass motion capabilities necessary for goal-directed local navigation. Based on the designer's assessment and intuition, a number of fundamental behavioral requirements are identified leading to definition of the primitive level. Useful composite behaviors at the level above are then identified, which can conceptually emerge from combining fundamental capabilities of two or more identified primitive behaviors. This process continues until the designer is satisfied that the suite of behaviors is sufficient to realize the overall purpose of the navigation system. Inputs, outputs, and associated membership functions are defined for each fuzzy behavior. Rules are then formulated that govern the desired response of the behavior for all practical inputs. This is followed by an iterative test-debug-tune cycle, which might be performed off-line using computer simulation. Additional refinement takes place on the actual robot when inadequacies are discovered that were not evident in simulation. This trial-and-error procedure involves fine-tuning of the shapes of membership functions used to express uncertainty in inputs and outputs, as well as modifications to the fuzzy rule base. The duration of this process is a function of system complexity and the availability of expert knowledge or intuition. Manual formulation of rules for primitive behaviors is straightforward because the control policies are relatively simple. Manual formulation of rules for composite behavioral capabilities such as goal-seeking and route-following is a considerably more complex and arduous task. It is perhaps the most challenging aspect of applying the approach. We will elaborate on this issue later in Section 7.

The total number and individual purpose of fuzzy behaviors in a given behavior hierarchy is indicative of the problem complexity and can be conveniently modified as required. Such modifications may become necessary in the event that tests with the physical robot(s) and environment reveal a need for additional behaviors or an alternative decomposition. Having established a nominal hierarchical arrangement of behaviors, the design proceeds as outlined above. Descriptions of the resulting suite of fuzzy behaviors are provided below.

4.1. Core Behavior Hierarchy

The behavior of each mobile robot is implemented using the hierarchical control architecture described in Section 2, wherein multiple behaviors are arranged as a hierarchical network of distributed fuzzy rule bases. This base architecture for individual robots[3] can be extended to enable cooperative control of multiple mobile robots by exploiting its flexibility and scalability.[10] Thus far, this has been achieved by employing a common core hierarchy of behaviors that enables cooperative group behaviors such as moving towards a given position without colliding with stationary obstacles or other robots. A core hierarchy for this *safe-homing* group behavior is depicted in Figure 5. It is composed of the p-behavior *homing* and the c-behavior *safe-wander*. *Homing* consists of moving towards a given position based on knowledge of the current robot position and its goal/target position. *Safe-wander* enables random motion without colliding with stationary obstacles or other robots. *Safe-wander* is composed of three p-behaviors: *avoid-obstacle*, *avoid-kin*, and *wander-randomly*. As its name implies, *avoid-obstacle* provides the survival capability for avoiding collisions with stationary obstacles. This p-behavior takes as inputs the distances to the nearest sensed obstacle in front and on both sides of the robot and recommends appropriate actuator outputs. *Avoid-kin* augments this capability with respect to robot team members by enabling a robot to avoid collisions with other robots. It operates in a similar fashion, considering only the distance and angle to the nearest robot team member. Finally, *wander-randomly* ensures avoidance of deadlock

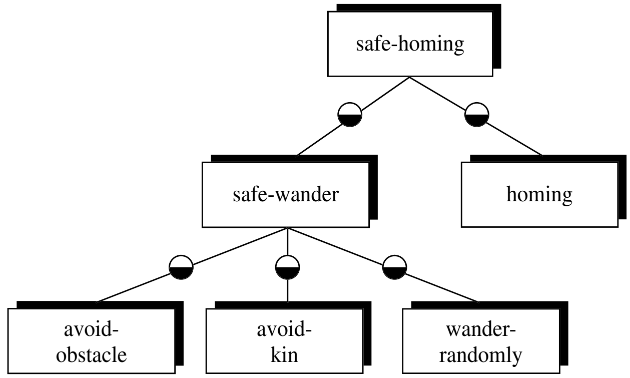

Figure 5. *Safe-homing* behavior hierarchy.

situations by directing a robot to move forward with rotational directives issued randomly.

Executing on each robot during group motion, *safe-wander* modulates the DOAs of *avoid-obstacle*, *avoid-kin*, and *wander-randomly*, in response to distance measurements to the nearest obstacle and robot. At a higher level, *safe-homing* modulates the DOAs of *homing* and *safe-wander*. This core behavior hierarchy aggregates basic navigational and survival behaviors with communications and cooperative response behavior. With a hierarchical fuzzy controller executing on each robot, it is possible to realize various intelligent group behaviors such as safe wandering, homing, and flocking, which emerge as a result of robot interactions.[10]

Note that decomposition of behavior for a given mobile robot system is not unique. Consequently, suitable behavioral repertoires and associated hierarchical arrangements are arrived at following a subjective analysis of the system and the task environment. Note also that there is no requirement for behaviors in the primitive level to be implemented using fuzzy logic. In fact, behaviors with crisp inputs and outputs that are implemented using other techniques (e.g., classic control or neural networks) can easily be accommodated by the hierarchy. This is possible because their crisp outputs can be folded into Equation 4 as singleton fuzzy sets.

5. MOBILE ROBOT GROUP ARCHITECTURES

We describe two group architectures that have been implemented using the hierarchical fuzzy behavior-based approach. The first consists of a homogeneous group of mobile robots that work collectively to achieve a common mission generally classified as foraging. The focus of the second group architecture is a heterogeneous group of robots that are organized as a hierarchy themselves to achieve missions involving area coverage.

The homogeneous group is comprised of two identical robotic platforms. For this pair of robots, mission objectives are achieved via parallel execution of a common

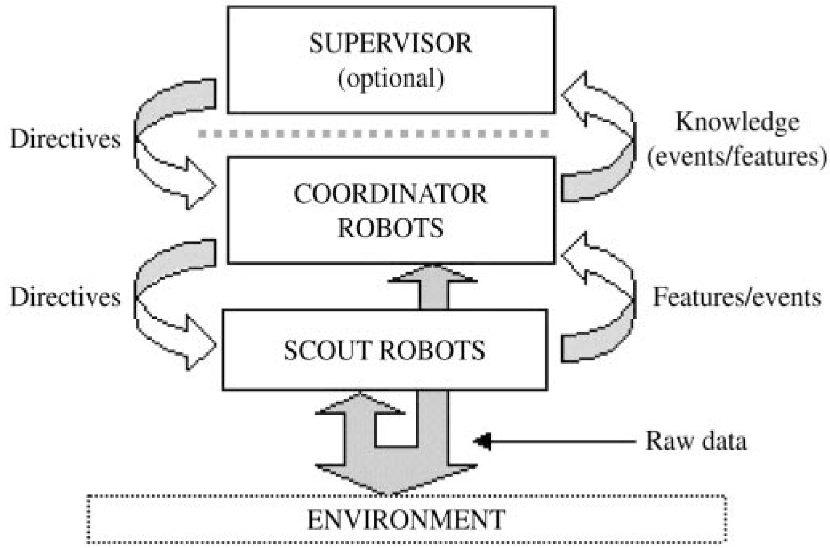

Figure 6. Group hierarchy for coordinator-scout robot ensemble.

core behavior hierarchy. The robots receive common directives from a centralized source of environmental sensory data and execute the directives collectively until the mission is accomplished. An example is provided in the following section.

The heterogeneous group is comprised of two different types of mobile robot, each with different sensors and actuators. In this case, the multiple robots are themselves organized as a loose hierarchy, with the cooperative control strategies performed by the robot(s) at the highest level. Cooperation among robots is mediated through a two-tier hierarchy, based on two classes of robots: *coordinators* and *scouts*. A cluster of scout robots is associated with a coordinator robot. We refer to the coordinator-scout cluster as an *ensemble* and its hierarchical and functional disposition is depicted in Figure 6. This figure also shows an optional supervisor layer, for situations where remote telemetry or human-in-the-loop are required.

In our conception, scout and coordinator robots employ the same robotic platform software and core behavior hierarchy, although coordinators are typically encoded with more deliberative capabilities. Differentiation is manifested in the subset of behaviors employed, allowing a scout to take over the role of a coordinator, if the latter is absent. Likewise, if the capabilities of a coordinator are compromised in some manner, it can be reconfigured as a (less-capable) scout. This reconfigurability is a feature of the hierarchical architecture that can be exploited by presetting or adjusting behavioral DOAs—in essence, enabling only those behaviors in the hierarchy that are useful for coordinators versus scouts. A similar concept is proposed for modular design of robot teams in ref. 18. Scouts are configured to utilize minimal autonomous decision-making capability, sufficient only for self-preservation and basic navigation. They utilize their resources primarily for mission-relevant data collection. On the other hand, coordinator robots are configured for more sophisticated navigation, and multiple scout coordination and control capabilities. They

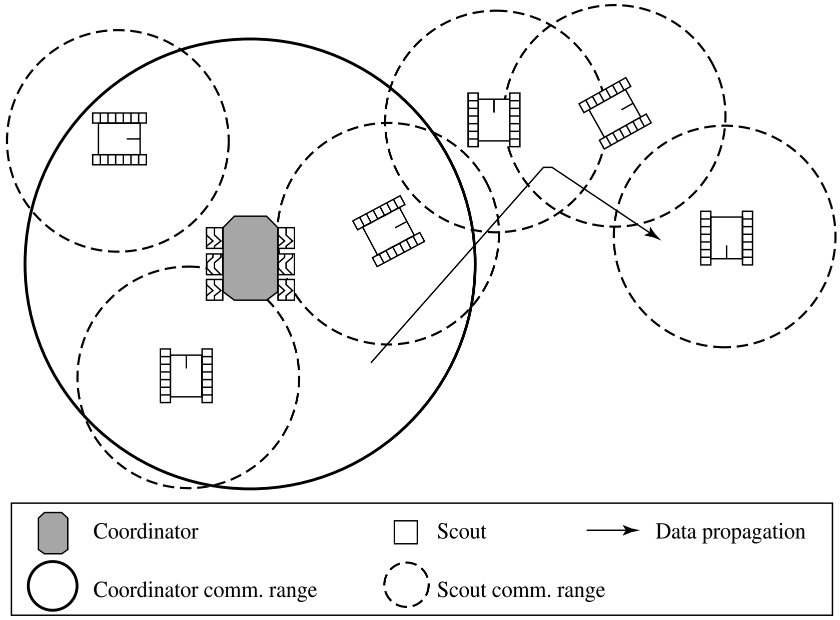

Figure 7. Data propagation within robot ensemble.

limit their sensory resources to self-preservation and navigation uses, rather than data collection.

The ensemble is defined by the coordinator robot and the scouts that are within the coordinator range of communication (and therefore, range of influence). Inter-robot communication is realized through intermittent, short bursts of data. In general, since the communication range of each scout is limited, data originating at a certain location will gradually propagate to other locations via robots in the group (see Figure 7). In this manner, incoming environmental information (e.g., a priori or acquired map data) is integrated into the world representation of each robot and propagated, as part of its current database, in posterior bursts.[19] Furthermore, communication between coordinator and scout units may also contain mission directives that modify the basic scout behaviors of data collection and self-preservation, resulting in collectively directed action. Examples of group directives include go-to-position, follow, halt, and go-direction, which are application-dependent.

6. PHYSICAL MOBILE ROBOT APPLICATIONS

Two examples are presented to describe how the hierarchical fuzzy control approach can be used in practical applications to coordinate and control multi-robot group behavior. The first example describes a foraging application involving the homogeneous group of two robots.[20] The foraging application has a variety of practical parallels in robot application domains. Some of these include the search and disarming of landmines (demining), search and collection of particular soil samples at hazardous waste sites or on remote planetary surfaces, collection and disposal of trash, and so on. In the example, foraging is performed in a laboratory setting wherein the objects of the search are represented by aluminum soda cans distributed on the laboratory floor. The second illustrative example describes a scenario involving the task of area coverage using the heterogeneous group of robots organized as an ensemble hierarchy.

6.1. A Foraging Example

The foraging mission of identifying target objects (soda cans) in an indoor environment was chosen since it requires a relatively simple setup, it allows comparison to other projects (see, for example, Ref. 21) due to its popularity as a multi-robotic application, and it lays a foundation for more interesting applications such as those mentioned above.

Two identical Pioneer 2-DX mobile robots (named LACHISH and NEGEV) are used to forage for target objects. These commercially available mobile robots are 38 cm long, 22 cm wide, and 44 cm tall, with mobility provided by two driven wheels and a passive caster. Each is equipped with front and rear sonars, bump sensors, encoders, a photoelectric sensor, and wireless Ethernet communication devices (with a range up to 50 meters in optimal conditions). In this example, the robots reach the soda cans and subsequently return to a home position using the core behavior hierarchy for *safe-homing*. An additional p-behavior, *constant-velocity*, was added at the same level of the hierarchy as the *safe-wander* and *homing* behaviors. The *constant-velocity* behavior simply guides each robot to move forward at a constant velocity. We found this behavior useful for increasing the rate of progress during foraging, which was relatively slow due to frequent interactions between *homing* and *avoid-obstacle*.

In this case, the c-behavior *safe-wander* modulates the DOAs of *avoid-obstacle* and *wander-randomly*, according to the distance to the nearest obstacle and robot and the probability of the robot being in a deadlock situation (based on persistent activation of *avoid-obstacle*). The c-behavior *safe-homing* modulates the DOAs of its underlying behaviors according to the distance to the nearest obstacle, the distance to the target, as well as the DOA of the *safe-wander* behavior and its recommended turn angle from the previous run.

Experiments were conducted in a 6 m $\times$ 8 m relatively open region of the laboratory (see Figure 8). Soda cans are targeted by two fixed overhead cameras, each viewing a different subregion of the laboratory. One region is 2 m $\times$ 2 m and the other is 1.2 m $\times$ 1.6 m. The robots' home position is situated approximately in between the two camera subregions. One region is directly in front of the home position and the other is to the right (Figure 9). Once a minute, black-and-white images of the regions are sampled and then thresholded and dilated. The soda cans are identified based on their size in the processed image. The locations of the detected soda cans are then broadcast to the robots and are expected to lie within only one of the two subregions.

Prior to task execution, each robot starts at the home position. They wait until a target is acquired before executing the *safe-homing* behavior to reach the target. When the target vicinity is reached, the robots verify that a soda can object is indeed

Figure 8. Foraging experimental environment.

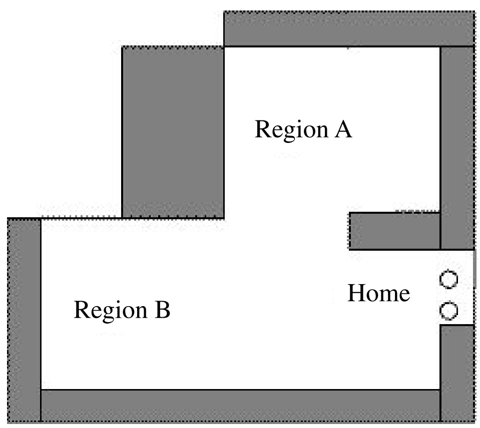

Figure 9. Schematic diagram of experiment.

present using a photoelectric range detector. If successful, the robot announces its findings and returns to the home position. The robots do not actually pick up the soda cans because they are currently not equipped with grippers; the cans are manually removed by the experimenters. Ten successive runs in which the robots set out to find cans and return to the home position are summarized in Table I. In all runs, except runs 3 and 10, LACHISH left the home position 15 seconds after NEGEV. After each run, the robots were reset to avoid accumulation of encoder errors. Nine runs terminated successfully; i.e., the robots found the soda cans and returned successfully to the home position. When a can had already been found by the other robot (and manually removed by the experimenter), the robots correctly identified that it was not there. Each run was completed within one to four minutes. However, run 5 had

Table I. Group foraging results.

Run	Region (NEGEV)	Time (sec)	Region (LACHISH)	Time (sec)	Remarks
1	A	110	A	165	
2	B	150	A	110	
3	A	240	A	180	No reset after run 2
4	A	120	A	120	
5	A	—	A	—	Robots collided
6	B	130	A	120	
7	A	90	B	165	
8	A	135	A	270	LACHISH collided with table
9	B	165	B	240	
10	A	165	B	180	

to be manually stopped because the robots collided with each other due to a failure to resolve inter-robot behavioral interactions. This occurred during an instance in which LACHISH and NEGEV were moving in opposite directions while close to an obstacle on their side. The behavioral failure occurred while attempting to satisfy the dual objective of avoiding collision with each other and the obstacle. Such an event is not unlike the situation wherein two people collide while walking toward one another after committing to conflicting decisions about avoidance maneuvers. This revealed a need for traffic rules and/or inter-robot communication to facilitate group navigation when the robots collectively execute parallel tasks in a shared workspace. Formal multi-objective decision-making approaches for behavior-based systems[22] are particularly useful in such cases.

6.2. Area Coverage Scenario

The ensemble architecture is being implemented in a heterogeneous group of three robots. A commercially available Koala robot named MAX is used as a coordinator robot, and two identical custom-built laboratory robots are used as scouts due to the greater computational resources of the former. As mentioned earlier, however, the flexibility of the hierarchical fuzzy control architecture permits each robot to be configured for alternative group roles. As shown in Figure 10, the Koala robot is a six-wheeled vehicle and the scouts employ a tracked mobility system. MAX is the larger robot with dimensions of: length: 32 cm, width: 32 cm, and height: 20 cm; respective dimensions for the scout robots are 21 cm, 20 cm, and 13 cm. Each of these robots possesses only range-finding sensors for obstacle detection/avoidance and encoder-based odometry for navigation. MAX has 12 peripherally mounted infrared (IR) sensors, while the scout robots currently have four perimeter-mounted IR sensors. The robots are equipped with similar radio frequency transceivers (mounted atop each robot). These are UHF band FM transceivers with a range of about 30 meters in optimal conditions and a data rate up to 38,400 bits per second. MAX utilizes a communication scheme executed by an ancillary Motorola 68HC11 microcontroller coupled to its transceiver.

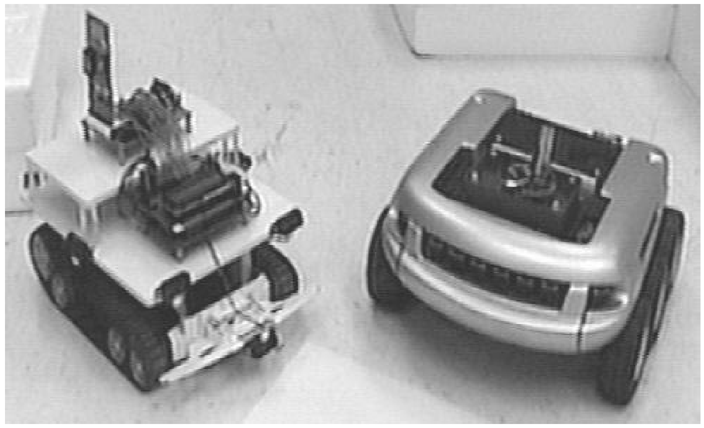

Figure 10. Ensemble scout robot (left) and coordinator robot (right).

This heterogeneous group of robots can be used to achieve tasks of area coverage for exploration. In an exploration scenario, scouts commence distributed coverage of a region of interest after being led to the region by a coordinator. The group could collectively navigate using a core behavior hierarchy that supports motion in particular formations (not supported in the current design). Alternatively, the coordinator can execute the navigation while the scouts follow using a *follow-leader* behavior added to the hierarchy described above. A simple way to implement this is to install a light source or beacon on MAX, which is tracked by light sensors on the scout robots. The coordinator illuminates its beacon when leading scouts to a region of interest, and shuts it off when the location is reached. So the directive to start work is communicated to scouts optically by the absence of the coordinator beacon. When the coordinator beacon is on, the nominal approach to area coverage by the scouts is based on execution of the *safe-wander* behavior in the region of interest; otherwise scouts activate a *follow-leader* behavior.

6.2.1. Example

In the experimental scenario, the ensemble's task is coverage of a designated area (for example, to gather data from all reachable parts of some interesting terrain). The terrain has subareas that the coordinator cannot reach or venture into due to its larger size or poor traversability for wheels versus tracks, for instance. Coverage of such subareas within the general area of interest must be achieved by the scouts. The coordinator is responsible for leading the scouts to the designated area, while the scouts are tasked with the coverage job.

This area coverage scenario is illustrated in Figure 11, and it was carried out in a laboratory setting using MAX and the scout robots. Figure 12 shows the ensemble operating in a representative experimental trial as recorded by a sequence of overhead camera images. The trial run was executed in an area bounded by four walls enclosing the robots and styrofoam boxes used as generic obstacles/barriers. This figure records the first 3.5 minutes of area coverage trajectories executed by the scouts upon activation of the *safe-wander* behavior. The coordinator, MAX, is situated at

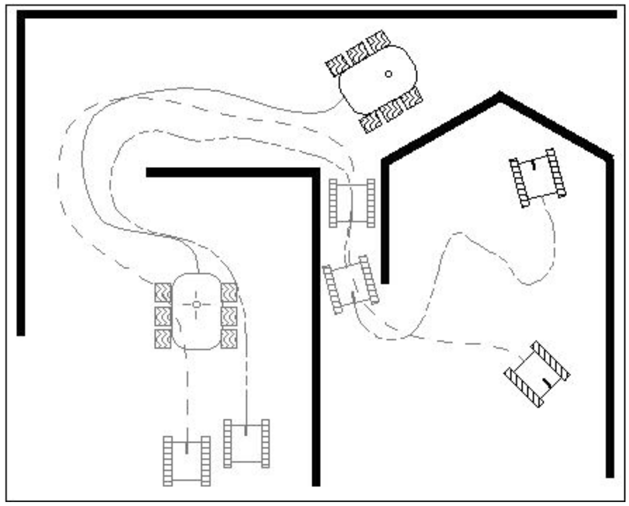

Figure 11. Ensemble area coverage scenario.

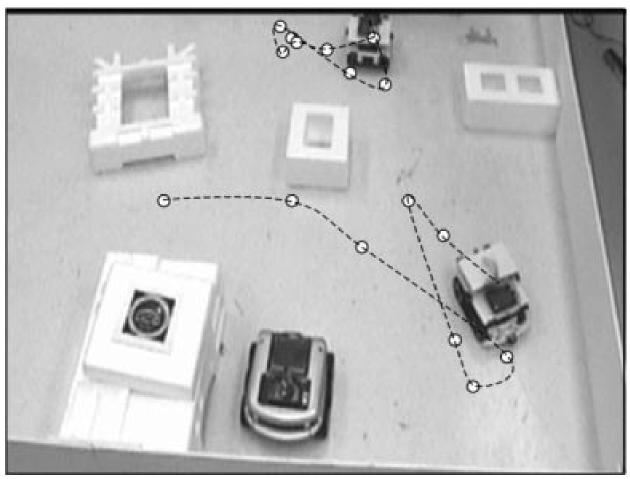

Figure 12. Area coverage experimental run.

the bottom center of the figure, having issued a directive to initiate area coverage. Scout trajectories are depicted as dashed lines with circles marking periodic positions along the trajectory approximately every 25 seconds. The random-walk nature of the *safe-wander* behavior is evident in Figure 12. Indeed, more structured exploration behaviors could be employed for systematic coverage of designated areas in real applications.

7. DISCUSSION AND CONCLUSIONS

The illustrative example implementations for foraging and area coverage provide basic demonstrations of how the proposed multi-robot control scheme might be applied. The hierarchy of fuzzy behaviors offers an efficient approach to synthesis of behavioral capabilities necessary for these and other autonomous navigational tasks. Its practical utility lies in the hierarchical decomposition of overall behavior into sub-behaviors that are activated only when applicable. This approach, along with its mechanism for weighted decision-making, provides a suitable framework for situated adaptation in single and multiple autonomous vehicles. As in the philosophy of our approach, the "trick" is in applying these fuzzy logic techniques when, and where, they are most feasible.

One of the salient attributes of the approach, in the spirit of the subsumption architecture, is the solipsist view that each primitive has of the "world." The functionality of the system depends on a combined effect of the behavioral functionality of each primitive and the competence of the composite behaviors that coordinate them. As mentioned earlier, perhaps the most difficult aspect of applying the approach is the formulation of applicability rules for c-behaviors, particularly when they modulate several underlying p-behaviors. In earlier work,[23] we successfully addressed the issue of which behaviors to activate, and to what degrees, using genetic programming[24] to computationally evolve fuzzy coordination rules off-line. However, the evolutionary approach leaves something to be desired with regard to the speed with which solutions can be obtained. Behavior evolution is one among several computational alternatives to manual formulation of the applicability rule bases responsible for appropriately modulating p-behaviors. Other learning or optimization approaches may be applied to facilitate the process. For example, this problem has been addressed in the contexts of other coordination schemes by using reinforcement learning[25] and hybrids of reinforcement and neural networks.[26,27]

Low-level and high-level behaviors, as well as the behavior coordination mechanisms of the hierarchical structure, are based on fuzzy set theory and fuzzy logic. This uniformity in representation permits design flexibility and extensibility of the architecture for application to increasingly complex problems. In addition, it facilitates the integration of the various layers of hierarchical control systems. The practical applications to basic mobile robot navigational tasks demonstrate the scalability of the core architecture from single to multiple robot control. It is conceivable that the coordination mechanisms could be useful at a larger scale where individual agents/robots are treated in the same way as individual behaviors in the hierarchy.

A significant feature of the architecture that could be the focus of future extensions is behavior threshold activation. Thresholds imposed on DOAs would allow filtering of undesirable interbehavioral influences. Threshold activation has not been fully exploited in the research reported here. The feature remains as an additional degree of freedom of the architecture that deserves further attention. A simple approach to exercising threshold activation within the hierarchy as described here is presented in Ref. 28. A preliminary assessment of its impact on local navigational performance is provided there as well. In addition, the computational mechanism for behavior modulation deserves further attention. Namely, the availability of various formulations for computing the information and control interface between hierarchical layers should be exploited to determine their relative merits in different problem domains.

The fuzzy-behavior approach is being adopted for planetary rover navigation in rough terrain.[29] It extends the approach by accounting for third-dimensional characteristics of terrain such as roughness, slope, and discontinuities, as well as vehicle safety against hazards such as tip-over and excessive wheel slippage. A fuzzy logic rule set is added to reason about these characteristics and infer a degree of terrain traversability, which is then factored into behavioral decisions for goal-seeking. Finally, the approach presented here is also being extended for application to multiple cooperating mobile robots in the competitive robot soccer domain.[9,30] In this extension, robots are controlled by a behavior hierarchy of strategic motion behaviors meant to emulate soccer players. Both of these extensions have been implemented and tested on physical mobile robots.

Acknowledgment

The work described in this article was started at the Center for Autonomous Control Engineering, University of New Mexico, sponsored by NASA under contract # NCCW-0087, and completed in part at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with NASA. Marco A. A. de Oliveira is supported by the CNPq scholarship 200267/97-3(NV)-GDE, UFG contract 23070.003737/96-81. Research conducted at BGU was partially supported by the BGU Paul Ivanier Center for Robotics & Production Management.

References

1. Brooks RA. A robust layered control system for a mobile robot. IEEE J Robot Automat 1986;RA-2(1):14–23.
2. Saffiotti A, Konolige K, Ruspini EH. A multi-valued-logic approach to integrating planning and control. Artif Intell 1995;76(1–2):481–526.
3. Tunstel E. Mobile robot autonomy via hierarchical fuzzy behavior control. In: 6th International Symposium on Robotics and Manufacturing, 2nd World Automation Congress. Montpellier, France; May 1996. pp 837–842.
4. Michaud F, Lachiver G, Le Dinh CT. A new control architecture combining reactivity, planning, deliberation and motivation for situated autonomous agent. In: IEEE Int Conf on Fuzzy Systems, 1996. pp 258–264.
5. Correia L, Steiger-Garçao A. A useful autonomous vehicle with a hierarchical behavior control. In: Merelo JJ, Móran F, Moreno A, Chácon P, editors. Advances in artificial life, 3rd European Conference on Artificial Life. Springer, Granada, Spain, 1995. pp 625–639.
6. Moreno L, Moraleda E, Salichs MA, Pimentel JR, de la Escalera A . Fuzzy supervisor for behavioral control of autonomous systems. In: Int Conf on Industrial Electronics, Control, and Instrumentation IECON'93, 1993. pp 258–261.
7. Jamshidi M. Large-scale systems: Modeling, control, and fuzzy logic. Environmental and Intelligent Manufacturing Systems, volume 8. Upper Saddle River, NJ:Prentice Hall PTR; 1997.
8. Mamdani EH. Twenty years of fuzzy control: Experiences gained and lessons learnt. In: IEEE Int Conf on Fuzzy Systems; 1993. pp 339–344.
9. de Oliveira M, Barrera R. A micro-controller-based hierarchical fuzzy controller for cooperative multiagent autonomous robots. In 7th International Symposium on Robotics with Applications, 3rd World Automation Congress. Anchorage, Alaska, May 1998.
10. Berman S, de Oliveira MAA, Edan Y, Jamshidi M. Hierarchical fuzzy behavior-based control of a multi-agent robotic system. In: 7th IEEE Mediterranean Conf on Control and Automation. Haifa, Israel, June 1999.
11. Tunstel E. Fuzzy-behavior synthesis, coordination, and evolution in an adaptive behavior hierarchy. In: Saffiotti A, Driankov D, editors. Fuzzy Logic Techniques for Autonomous

Vehicle Navigation, Studies in Fuzziness and Soft Computing, chapter 9. Springer-Verlag; 2000.
12. Staddon JER. Adaptive behavior and learning. New York: Cambridge University Press; 1983.
13. Driankov D, Hellendoorn H, Reinfrank M. An introduction to fuzzy control. Berlin, Germany: Springer-Verlag; 1993.
14. Yager RR, Filev DP. Essentials of fuzzy modeling and control. New York: Wiley and Sons; 1994.
15. Everett HR. Sensors for mobile robots: Theory and application. Wellesley, MA: AK Peters, Ltd.; 1995.
16. Borenstein J, Feng L. UMBmark—a method for measuring, comparing, and correcting dead-reckoning errors in mobile robots. Technical Report UM-MEAM-94-22. Ann Arbor, MI: The University of Michigan; 1994.
17. Cao YU, Fukunaga AS, Kahng AB. Cooperative mobile robotics: Antecedents and directions. Autonomous Robots 1997;4:1–23.
18. Navarro-Serment LE, Grabowski R, Paredis CJJ, Khosla PK. Modularity in small distributed robots. In: SPIE Conf on Sensor Fusion and Decentralized Control in Robotic Systems II. Boston, MA, Sep. 1999.
19. de Oliveira MAA, Berman S, Tunstel E, Jamshidi M. Remote surface exploration with multiple soft computing based cooperative rovers. In: 8th International Symposium on Robotics with Applications, 4th World Automation Congress. Maui, HI, Jun. 2000.
20. Berman S, Edan Y, Jamshidi M. A foraging group of autonomous, mobile robots: Implementation of hierarchical fuzzy behavior-based control. In: 21st IEEE Convention of the Electrical and Electronics Engineers in Israel, Tel-Aviv, Israel, April 2000.
21. Balch T, et al. Io, Ganymede and Callisto—a multiagent robot trash collecting team. AI Magazine 1995;16(2):39–51.
22. Pirjanian P. Multiple objective behavior-based control. Robotics and Autonomous Systems 2000;31(1–2):53–60.
23. Tunstel E, Lippincott T, Jamshidi M. Behavior hierarchy for autonomous mobile robots: Fuzzy-behavior modulation and evolution. Int J Intell Automation Soft Computing 1997;3(1):37–49.
24. Koza JR. Genetic Programming: On the programming of computers by means of natural selection. Cambridge, MA: MIT Press; 1992.
25. Bonarini A, Basso, F. Learning the suitability of simple behaviors to obtain composite behaviors for autonomous agents. In 1st On-Line Workshop on Soft Computing, 1996.
26. Koh KC, Beom HR, Cho HS. Behavioral control in mobile robot navigation using fuzzy decision making approach. In: IEEE Int Conf on Intelligent Robotic Systems, 1994. pp 1938–1945.
27. Gachet D, Salichs MA, Moreno L, Pimentel JR. Learning emergent tasks for an autonomous mobile robot. In: IEEE Int Conf on Intelligent Robotic Systems, 1994. pp 290–297.
28. Tunstel E. Fuzzy behavior modulation with threshold activation for autonomous vehicle navigation. In: 18th Int Conf of the North American Fuzzy Information Processing Society, New York, Jun 1999. pp 776–780.
29. Tunstel E, Seraji H, Howard A. Soft computing approach to safe navigation of autonomous planetary rovers. In: Zilouchian A, Jamshidi M, editors. Intelligent Control Systems Using Soft Computing Methodologies, chapter 11. CRC Press LLC; 2000.
30. Delgado SM, et al. Soft computing based approach to cooperative autonomous robotics. In: NASA University Research Centers Technical Conference, Hunstville, AL, 1998.