

Blending Reactivity and Goal-Directedness in a Fuzzy Controller

Alessandro Saffiotti* Enrique H. Ruspini Kurt Konolige

Artificial Intelligence Center, SRI International
Menlo Park, CA 94025, U.S.A.
saffiotti@ai.sri.com

Abstract— Controlling the movement of an autonomous mobile robot requires the ability to pursue strategic goals in a highly reactive way. We describe a fuzzy controller for such a mobile robot that can take abstract goals into consideration. Through the use of fuzzy logic, reactive behavior (e.g., avoiding obstacles on the way) and goal-oriented behavior (e.g., trying to reach a given location) are smoothly blended into one sequence of control actions. The fuzzy controller has been implemented on the SRI robot Flakey.

I. INTRODUCTION

Autonomous operation of a mobile robot in a real environment poses a series of problems. In the general case, knowledge of the environment is partial and approximate; sensing is noisy; the dynamics of the environment can only be partially predicted; and robot's hardware execution is not completely reliable. Though, the robot must take decisions and execute actions at the time-scale of the environment. Classical planning approaches have been criticized for not being able to adequately cope with this situation, and a number of reactive approaches to robot control have been proposed (e.g., [Firby, 1987; Kaelbling, 1987; Gat, 1991]), including the use of fuzzy control techniques (e.g., [Sugeno and Nishida, 1985; Yen and Pfluger, 1992]). Reactivity provides immediate response to unpredicted environmental situations by giving up the idea of reasoning about future consequences of actions. Reasoning about future consequences (sometimes called “strategic planning”), however, is still needed in order to intelligently solve complex tasks (e.g., by deciding not to carry an oil lantern downstairs to look for a gas leak [Firby, 1987].)

One solution to the dual need for strategic planning and reactivity is to adopt a two-level model: at the upper level, a planner decides a sequence of abstract goals to be achieved, based on the available knowledge; at the lower

level, a reactive controller achieves these goals while dealing with the environmental contingencies. This solution requires that the reactive controller be able to simultaneously satisfy strategic goals coming from the planner (e.g., going to the end of the corridor), and low-level “innate” goals (e.g., avoiding obstacles on the way). A major problem in the design of such a controller is how to resolve conflicts between simultaneous goals.

In this paper, we describe a reactive controller for an autonomous mobile robot that uses fuzzy logic for trading off conflicting goals. This controller has been implemented on the SRI robot Flakey, and its performance demonstrated at the first AAAI robot competition, where Flakey finished second [Congdon *et al.*, 1993]. The formal bases for the proposed controller have been set forth by Ruspini [Ruspini, 1990; Ruspini and Ruspini, 1991; Ruspini, 1991a] after the seminal works by Zadeh (e.g., [Zadeh, 1978]). In a nutshell, each goal is associated with a function that maps each perceived situation to a measure of desirability of possible actions from the point of view of that goal. The notion of a *control structure* is used for introducing high-level goals into the fuzzy controller. Intuitively, a control structure is an object in the robot's workspace, together with a desirability relation: typical control structures are locations to reach, walls to follow, doors to enter, and so on. Each desirability function induces a particular behavior — one obtained by executing the actions with higher desirability. Behaviors induced by many simultaneous goals can be smoothly blended by using the mechanisms of fuzzy logic. In particular, reactive and goal-oriented behaviors are blended in this way into one sequence of control actions.

The next section gives a brief overview of Flakey. Section III sketches the architecture of the controller, and describes the way behaviors are implemented, and how they are blended together. Section IV deals with the introduction of high-level goals into the reactive controller. Section V discusses the results, and concludes.

*On leave from Iridia, Université Libre de Bruxelles, Brussels, Belgium.

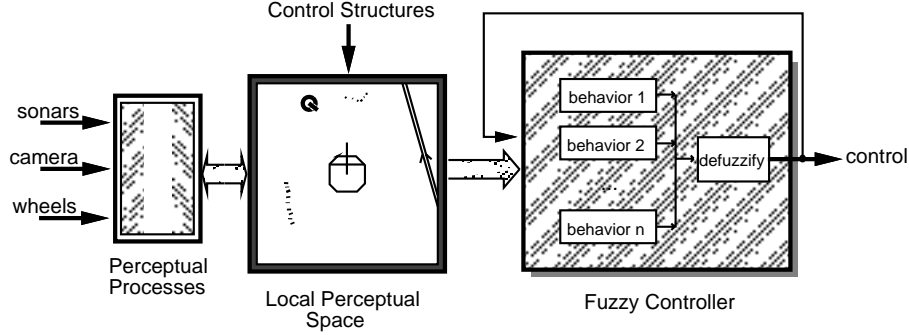


Figure 1: Architecture of Flakey (partial)

II. THE MOBILE ROBOT TEST-BED

Flakey is a custom-built mobile robot platform approximately 1 meter high and .6 meter in diameter for use in an indoor environment. There are two independently-driven wheels, one on each side, giving a maximum linear velocity of about .5 meters/sec. Flakey sensors include a ring of 12 sonars, giving information about distances of objects up to about 2 meters; wheel encoders, providing information about current linear and rotational velocity; and a video camera, currently used in combination with a laser to provide dense depth information over a small area in front of Flakey. On-board computers are dedicated to low-level sensor interpretation, motor control, and radio communication with an off-board Sparc station. Though it is possible to run the high level interpretation and control processes on board, they are normally run remotely for programming convenience.

Figure 1 illustrates the part of Flakey's architecture that is relevant to the controller. The sensorial input is processed by a number of interpretation processes at different levels of abstraction and complexity, and the results of interpretation are stored in the *local perceptual space* (LPS). The LPS represents a Cartesian plane centered on Flakey where all the information about the vicinity of Flakey is registered. In Figure 1, points corresponding to surfaces identified by the sonars and the camera are visible in the LPS — Flakey is the octagon in the middle of the LPS, in top-view. The other objects in the LPS are “artifacts” associated to *control structures*, and are discussed in Section V. The content of the LPS constitutes the input to the controller: this checks its input and generates a control action every 100 milliseconds.

III. REACTIVE FUZZY CONTROLLER

The fuzzy controller is centered on the notion of *behavior*. Intuitively, a behavior is one particular control regime that

focuses on achieving one specific, predetermined goal (e.g., avoiding obstacles). Hence, we can think of a behavior as a mapping from configurations in the LPS to actions to perform. More precisely, and following [Ruspini, 1991b], we say that each behavior B is associated with a desirability function

$$Des_B : LPS \times Control \rightarrow [0, 1]$$

that measures, for each configuration s of the LPS and value c of a control variable, the desirability $Des_B(s, c)$ of applying control values c in the situation s from the point of view of B . Equivalently, we can say that Des_B associates each situation s with the fuzzy set \tilde{C} of control values characterized by the membership function $\mu_{\tilde{C}}(c) = Des_B(s, c)$. Notice that in general, c is a n -dimensional vector of values for all the control variables; in the case of Flakey, the control variables include linear acceleration and turning angle.

In practice, each behavior is implemented by a fuzzy machine structured as shown in Figure 2. The *fuzzy state* is a vector of fuzzy variables (each having a value in $[0, 1]$) representing the truth values of a set of fuzzy propositions of interest (e.g., “obstacle-close-on-left”). At every

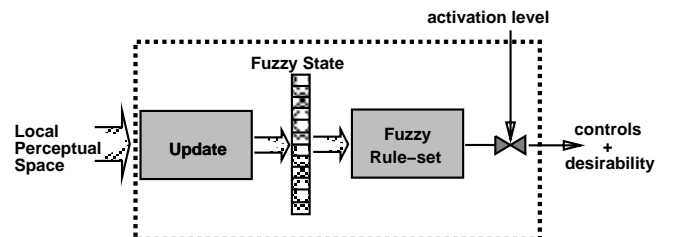


Figure 2: Implementation of a behavior.

cycle, the **Update** module look at the (partially) interpreted perceptual input stored in LPS, and produces a new fuzzy state. The **Fuzzy Rule-Set** module contains a set of fuzzy rules of the form “If A then c ” where A is a fuzzy expression composed by predicates in the fuzzy states plus the fuzzy connectives AND, OR and NOT; and c is a vector of values for the control variables. Max, min, and complement to 1 are used to compute the truth value of disjunction, conjunction and negation, respectively. An example of a control rule is:

```
IF obstacle-close-in-front
AND NOT obstacle-close-on-left
THEN turn -6 degrees
```

Each “If A then c ” rule computes the degree of desirability of applying control value c as a function of the degree at which the current state happens to be similar to A . The outputs of all the rules in a rule-set are unioned using the max T-conorm: the function computed in this way is meant to provide an approximation of the Des_B function above.¹ This desirability function is fed to the **Defuzzify** module for computing one single control value. We presently do defuzzification according to the centroid approach: the resulting control value is given by

$$\frac{\int c Des_B(c) dc}{\int Des_B(c) dc}.$$

As shown in Figure 1 above, many behaviors can be simultaneously active in the controller, each aiming at one particular goal — e.g., one for avoiding obstacles; one for keeping a constant speed; one for heading toward a beacon; etc. Correspondingly, many instances of the fuzzy machine depicted in Figure 2 simultaneously run in the controller, each one implementing one behavior’s desirability function. All these desirability functions are merged into a composite one by the max T-norm; the defuzzification module converts the resulting tradeoff desirabilities into one crisp control decision. Care must be taken, however, of possible conflicts among behaviors aiming at different, incompatible goals. These conflicts would result in desirability functions that assign high values to opposite actions: simple T-norm composition should not be applied in these cases. The key observation here is that each behavior has in general its own *context* of applicability. Correspondingly, we would like that the impact of the control actions suggested by each behavior be weighted according to that behavior’s degree of applicability to the current situation. For instance, the actions proposed by the obstacle avoidance behavior should receive higher priority when there is a danger of collision, at the expense

of the other, concurrent behaviors. In order to do this, the output of each rule-set is discarded by the value of the corresponding *activation level*: typically, the activation level is represented by some variable in the fuzzy state. This corresponds to arbitrate the relative dominance of different behaviors by a set of meta-rules of the form

$$\text{IF } A' \text{ THEN activate_behavior } B \quad (1)$$

where A' is a LPS configuration. Notice that this solution is formally equivalent to transforming each rule “If A then c ” in B into a rule “If A' and A then c ” (see [Berenji *et al.*, 1990] for a similar approach to conflict resolution.)

As an example, consider the way Flakey “wanders” around. In the wandering mode, three behaviors coexist in the controller: AVOID-OBSTACLES, AVOID-COLLISIONS and GO-FORWARD. GO-FORWARD just keeps Flakey going at a fixed velocity, given as a parameter. AVOID-OBSTACLES looks at the last 5 seconds’ sonar readings in the LPS, and guides Flakey away from occupied areas. AVOID-COLLISIONS looks at the nearest sonar readings and proposes drastic actions (immediate stop and turn) when a serious risk of collision is detected. The activation levels of AVOID-OBSTACLES and AVOID-COLLISIONS are given by the fuzzy state variable “approaching-obstacle”; the complement of this value gives the activation level for GO-FORWARD. The visual result for an external observer is that Flakey “follows its nose”, while smoothly turning away from obstacles as it approaches them.

IV. BEYOND PURE REACTIVITY

The behaviors discussed in the previous section are purely reactive: at each cycle, Flakey selects an action solely on the basis of the current state of the world as perceived by its sensors and represented in the local perceptual space. Engaging into more purposeful activities than just wandering around requires more than pure reactivity: we need to take explicit goals into consideration. For example, we may want Flakey to reach a given position at a given velocity, and still (reactively) avoid the obstacles on the way.²

In our approach, a goal is represented by a *control structure*. Intuitively, a control structure is virtual object (an *artifact*) that we put in the LPS, associated with a behavior that encodes the way to react to the presence of this object. For example, a “control-point” is a marker for a (x, y) location, together with a heading and a velocity: the associated behavior Go-To-CP reacts to the presence of a control point in the LPS by generating the commands to reach that position, heading and velocity. In Figure 1 there are two artifacts: a control point to reach (left), and a wall to follow (right).

¹ See [Ruspini, 1991a; Ruspini, 1991b] for an account of fuzzy logic and fuzzy control in terms of similarity and desirability measures, and the use of T-norms and T-conorms in this context.

² Reactive behaviors are also associated with (innate) goals, hard-wired in the definition of the behavior. We are now interested in dynamically assigning specific strategic goals to Flakey.

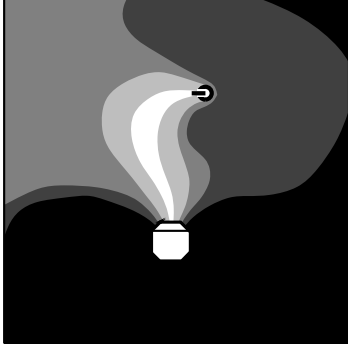


Figure 3: Path families generated by actions with increasing values of Des_F .

More precisely, a control structure is a pair

$$S = \langle Q_S, R_S \rangle,$$

where Q_S is an artifact, and R_S is a fuzzy relation between the position of Flakey and that of the artifact.³ Such a control structure implicitly defines a goal: the goal to achieve, and maintain, the given relation between Flakey and the artifact Q_S . Intuitively, if Q_S is at position q , $R_S(q, p)$ says how much a position p of Flakey satisfies this goal. If the position of Flakey is such that $R_S(q, p) = \alpha$, we say that the control structure S is satisfied to the degree α .

The R_S relation induces a desirability function Des_S in the following way. Given the set P of possible positions of Flakey, and the set C of possible control values, let $Exec(p, c)$ denote the new position reached by applying control c from position p . Then, the desirability *from the viewpoint of the control structure S* of executing c when Flakey is at position p (and Q_S is at q) is given by

$$Des_S(q, p, c) = R_S(q, Exec(p, c))$$

However, not all positions are equally reachable by Flakey: moving to certain positions will require more effort (changes in velocity and/or direction, time, etc.) than moving to others. To account for this, we consider a second desirability function Des_F : $Des_F(p, c)$ measures the desirability *from the viewpoint of Flakey's motion capabilities* of executing control action c when Flakey is at position p . The desirability of control actions from the joint viewpoints of feasibility for Flakey, and effectiveness with respect to the control structure S , is measured by the combination $Des_S \otimes Des_F$ (where \otimes is a T-norm). Figure 3 illustrates one such combined desirability function:

³Positions are actually points in a (x, y, θ, v) 4-D space.

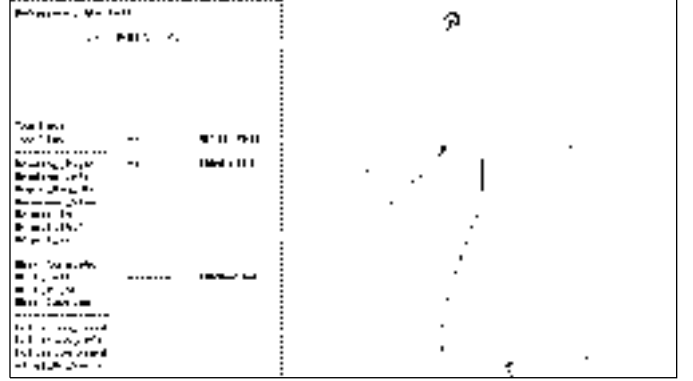


Figure 4: A snapshot of Flakey's control window while achieving a control point.

here, S is a control point, represented by the semi-circle near the top (the “tail” indicates the desired entry orientation.) The fading from black to white illustrates the increase in the value of $Des_S \otimes Des_F$ for some families of possible paths.

We have already seen how Des_S can induce, for each LPS configuration, a fuzzy set of possible controls. As we did in the case of reactive behaviors, we approximate this fuzzy set using rules on the form “If A then c ”. The only difference is that A now refers to artifacts rather than to sensorial input.⁴ We have designed sets of rules for many “purposeful” behaviors, including going to a x, y position; achieving a control point; following a wall; crossing a door; and so forth. Each ruleset consists of a small number (four to eight) of rules. Purposeful behaviors can coexist with other behaviors, either purposeful and reactive: the context-dependent blending of behaviors explained above provides arbitration and guarantees the smooth integration of directed activities and reactivity.

Figure 4 exemplifies the performance of the integration. The picture shows Flakey's control window during an actual run: on the right is Flakey's local perceptual space. Flakey sits in the middle of the window, pointing upwards; the small points all around mark sonar readings, indicating the possible presence of some object; the rectangle on the left of Flakey highlight a dangerously close object. The window on the left lists all the currently active rules, grouped into rule-sets: topmost, the rules for the Go-FORWARD behavior; below, those for Go-TO-CP, for AVOID-OBSTACLES, and for AVOID-COLLISIONS. In the shown situation, Flakey is going too slow and heading right of the CP: hence, some desirability is given to the

⁴Alternatively, these rules can be thought as responding to input from a “virtual sensor” that senses the position of an artifact.

accelerate and the *turn-left* actions. However, the close obstacle on the left causes the activation level of the AVOID-OBSTACLES behavior to be high, at the expenses of the other behaviors; hence, the *turn-right* action suggested by AVOID-OBSTACLES receives high total desirability (as indicated by the 7 stars). The small box in front of Flakey indicates the resulting turning control — some degrees on the right. The overall result of the blending is that Flakey makes its way among obstacles while *en route* to achieving the position and bearing of the given control point. The smoothness of the movement is evident in the wake of small boxes that Flakey left behind it (one box per second). Flakey’s speed was between 200 and 300 mm/sec.

One word is worth spent on the problem of local minima, ubiquitous in approaches to robot navigation based on local combination of behaviors [Latombe, 1991]. The problem is illustrated in Figure 5 (top): the robot needs to mediate the tendency to move toward the goal, and the tendency to stay away from the obstacle. A straightforward combination of these two opposite tendencies (whether they are described by desirability measures, potential fields [Khatib, 1986], motor schemas [Arkin, 1990], or other) may result in the production of a zone of local equilibrium (local minimum): when coming from the left edge, the robot would be first attracted and then trapped into this zone. By using meta-rules like the 1 above to reason about the relative importance of goals, our context-dependent blending of behaviors provides a way around this problem. Figure 5 shows the path followed by Flakey in a simulated run (top), and the corresponding activation levels of the KEEP-OFF and REACH behaviors (bottom). In (a), Flakey has perceived the obstacle; as the obstacle becomes nearer, the KEEP-OFF behavior becomes more active, at the expenses of the REACH behavior. In this way, the “attractive power” of the goal is gradually shaded away by the obstacle, and Flakey responds more and more to the obstacle-avoidance suggestions alone. The REACH behavior re-gains importance, however, as soon as Flakey is out of danger (b).

V. CONCLUSIONS

We have defined a mechanism based on fuzzy logic for blending multiple behaviors aimed at achieving different, possibly conflicting goals. Goals are either built-in, as in most fuzzy controllers, or dynamically set from outside the controller. Typically, the built-in goals correspond to reactive behaviors (like avoiding collisions), while the dynamic ones are strategic goals communicated by a planner. Context-dependent blending of behaviors ensures that strategic goals be achieved as much as possible, while maintaining a high reactivity.

Our behavior blending mechanism has been originally inspired to the technique proposed by Berenji et al.

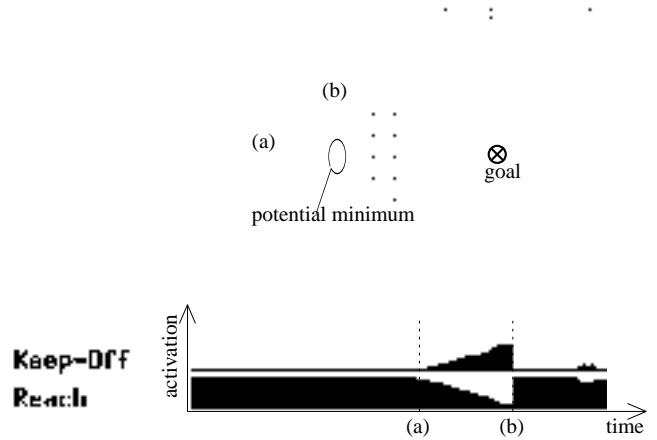


Figure 5: How context-dependent blending of behaviors avoids potential local minima.

[Berenji *et al.*, 1990] for dealing with multiple goals in fuzzy control. There are however two important differences: first, our context mechanism dynamically modifies the degrees of importance of each goal; second, we allow the introduction of high-level, situation-specific goals in the controller.

From another perspective, the work presented here fits in the tradition of the “two level” approaches to robot control, where a strategic planner is used to generate guidelines to a reactive controller (e.g., [Arkin, 1990; Payton *et al.*, 1990; Gat, 1991]). In our case, a plan consists in a sequence of control structures. For example, a plan to exit building E could consist in three successive *corridors* to follow, one *control point* in the entrance hall close to the door, and the exit *door* itself. The context of applicability of each control structure is used to decide when each control structure becomes relevant. (see [Saffiotti *et al.*, 1993; Saffiotti, 1993] for more on this issue). We believe that having based our architecture on fuzzy logic results in improved robustness (e.g., more tolerance to sensor noise and knowledge imprecision), while granting a better understanding of the underlying mechanisms.

Finally, many current approaches to robot control deal with multiple goals using the so-called “potential fields” method [Khatib, 1986]: goals are represented by pseudo-forces, which may be thought of as representatives of most desirable behavior from that goal’s viewpoint. These optimal forces are then combined, as physical vectors, to produce a resultant force that summarizes their joint effect. In our approach, by contrast, the goals’ desirability functions, rather than a summary description, are combined into a joint desirability function, from which a most desired tradeoff control is extracted. Moreover, this com-

bination takes behaviors' context of applicability into account; this provides a key to eliminate the local minima arising from the combination of conflicting goals.

The technique proposed in this paper has been implemented in the SRI mobile robot Flakey, resulting in extremely smooth and reliable movement. The performance of Flakey's controller has been demonstrated at the first AAAI robot competition in San Jose, CA [Congdon *et al.*, 1993]. Flakey accomplished all the given tasks while smoothly getting around obstacles (whose positions were not known beforehand) and people, and placed second behind Michigan University's CARMEL. Flakey's reliable reactivity is best summarized in one judge's comment: "Only robot I felt I could sit or lie down in front of." (What he actually did!)

Acknowledgments John Lowrance, Daniela Musto, Karen Myers and Leonard Wesley contributed to the development of the ideas presented in this paper. Nicolas Helft implemented a first version of Flakey's controller.

The first author has been supported by a grant from the National Council of Research of Italy. Research performed by the second author leading to the conceptual structures used in the autonomous mobile vehicle controller was supported by the U.S. Air Force Office of Scientific Research under Contract No. F49620-91-C-0060. Support for Kurt Konolige came partially from ONR Contract No. N00014-89-C-0095. Additional support was provided by SRI International.

REFERENCES

- [Arkin, 1990] Arkin, Ronald C. 1990. The impact of cybernetics on the design of a mobile robot system: a case study. *IEEE Trans. on Systems, Man, and Cybernetics* 20(6):1245-1257.
- [Berenji *et al.*, 1990] Berenji, H.; Chen, Y-Y.; Lee, C-C.; Jang, J-S.; and Murugesan, S. 1990. A hierarchical approach to designing approximate reasoning-based controllers for dynamic physical systems. In *Procs. of the 6th Conf. on Uncertainty in Artificial Intelligence*, Cambridge, MA.
- [Congdon *et al.*, 1993] Congdon, C.; Huber, M.; Kortenkamp, D.; Konolige, K.; Myers, K.; and Saffiotti, A. 1993. CARMEL vs. Flakey: A comparison of two winners. *AI Magazine*. To appear.
- [Firby, 1987] Firby, J. R. 1987. An investigation into reactive planning in complex domains. In *Procs. of the AAAI Conf.*
- [Gat, 1991] Gat, E. 1991. *Reliable Goal-Directed Reactive Control for Real-World Autonomous Mobile Robots*. Ph.D. Dissertation, Virginia Polytechnic Institute and State University.
- [Kaelbling, 1987] Kaelbling, L. P. 1987. An architecture for intelligent reactive systems. In Georgeff, M.P. and Lansky, A.L., editors 1987, *Reasoning about Actions and Plans*. Morgan Kaufmann.
- [Khatib, 1986] Khatib, O. 1986. Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research* 5(1):90-98.
- [Latombe, 1991] Latombe, J.C. 1991. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, MA.
- [Payton *et al.*, 1990] Payton, D. W.; Rosenblatt, J. K.; and Keirsey, D. M. 1990. Plan guided reaction. *IEEE Trans. on Systems, Man, and Cybernetics* 20(6).
- [Ruspini and Ruspini, 1991] Ruspini, E. H. and Ruspini, D. 1991. Autonomous vehicle motion planning using fuzzy logic. In *Procs. of the IEEE Round Table on Fuzzy and Neural Sys. and Vehicle Appl.*, Tokyo, Japan.
- [Ruspini, 1990] Ruspini, E. H. 1990. Fuzzy logic in the Flakey robot. In *Procs. of the Int. Conf. on Fuzzy Logic and Neural Networks (IIZUKA)*, Japan. 767-770.
- [Ruspini, 1991a] Ruspini, E. H. 1991a. On the semantics of fuzzy logic. *Int. J. of Approximate Reasoning* 5.
- [Ruspini, 1991b] Ruspini, E. H. 1991b. Truth as utility: A conceptual sythesis. In *Procs. of the 7th Conf. on Uncertainty in Artificial Intelligence*, Los Angeles, CA.
- [Saffiotti *et al.*, 1993] Saffiotti, A.; Konolige, K.; and Ruspini, E. H. 1993. Now, do it! Technical report, SRI Artificial Intelligence Center, Menlo Park, California. Forthcoming.
- [Saffiotti, 1993] Saffiotti, A. 1993. Some notes on the integration of planning and reactivity in autonomous mobile robots. In *Procs. of the AAAI Spring Symposium on Foundations of Automatic Planning*, Stanford, CA.
- [Sugeno and Nishida, 1985] Sugeno, M. and Nishida, M. 1985. Fuzzy control of a model car. *Fuzzy Sets and Systems* 16:103-113.
- [Yen and Pfluger, 1992] Yen, J. and Pfluger, N. 1992. A fuzzy logic based robot navigation system. In *Procs. of the AAAI Fall Symposium on Mobile Robot Navigation*, Boston, MA. 195-199.
- [Zadeh, 1978] Zadeh, L.A. 1978. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems* 1:3-28.