Fuzzy Control for Behavior-Based Mobile Robots

Jane Yung-Jen Hsu* Der-Chiang Lo Shih-Chun Hsu
Department of Computer Science and Information Engineering
National Taiwan University
Taipei, Taiwan 106, R.O.C.

Abstract— This paper presents an implemented control architecture for behavior-based mobile robots in partially known environments. In addition to coordinating primitive behaviors, the fuzzy controller is able to integrate reactive behaviors with goal-directed planning capabilities so that the robot can reach its goals more efficiently. The fuzzy controller was tested on a mobile robot equipped with sonars and infrared sensors to perform collision-free navigation toward any given goal position. The performance of the navigation system was compared with two existing algorithms, and the experimental results showed that the proposed architecture provides an efficient and flexible solution to higher level control problems.

I. INTRODUCTION

A mobile robot needs to be reactive in order to cope with uncertainties due to imperfect sensing and control, as well as incomplete knowledge about the world. Many traditional path planning algorithms rely on complete information and accurate robot hardware to find a collision-free path to the goal [9], [4]. Unfortunately, while the real world usually contains many static known objects such as walls and hallways, unknown and/or moving objects do exist. The dynamics in the world demand a controller to have a high degree of flexibility. Many existing navigation algorithms can handle unknown obstacles, but they still assume perfect sensing and control [13], [12].

In recent years, *behavior-based robots* have successfully demonstrated their competence in reacting to dynamic environments [1], [8]. Unlike symbolic AI that emphasizes high-level deliberative reasoning, a behavior-based robot emphasizes a more direct coupling of perception to action, whose overall task performance emerges as a result of the coexistence and cooperation of many individual behavior generating modules. For example, a mobile robot may be constructed from the primitive behaviors for wandering, path tracking, wall following, and obstacle avoidance etc., each of which may become active or inactive according to the current world situation. Several primitive behaviors have been implemented for a Nomad 200 mobile robots, which is equipped with sonar and infrared sensors.

Behavior-based robots using the subsumption architecture are simple and powerful. However, there are several problems associated with this approach. First of all, by eliminating explicit internal representations of the world, it is more difficult to take advantage of prior domain knowledge without carefully programming specific knowledge into the robot. Secondly, (variable) goal directed activi-

ties such as path planning are fundamentally incompatible with a fully reactive system. In particular, path planning can generate helpful information even if the environment is only partially known. One possible solution is to integrate a map representation into a behavior-based robot [2]. Last but not least, controlling the activations and inhibitions of the various behaviors is a complex task. For example, a mobile robot may start by finding an initial path using the *path planning* module. It can then use *path tracking* to move toward its goal. If any unknown object blocking the current path is encountered, the *obstacle avoidance* module will be activated. Once the obstacle is out of the way, the robot then tries to get back to the nearest intermediate points originally planned. If the robot has deviated too much from its planned path, *replanning* has to be done. In addition, due to errors accumulated in its mechanical movements, a mobile robot needs to reduce the differences between its encoder and actual positions by invoking the *localization* module.

Fuzzy control systems have been widely used for many applications that call for inexact and approximate reasoning, including low-level motion control of robots [10], [15], [16]. In the simple scenario above, the decisions about whether an object is blocking a path, when an object is out of the way, or how far the robot has to deviate from its planned course before replanning becomes necessary etc., often involve reasoning with inexact and incomplete information. As a result, fuzzy control provides a plausible solution for high level coordination of the primitive behaviors for such robots. A fuzzy controller takes the sensor readings as input, converts such crisp data into appropriate fuzzy sets and the corresponding values of the membership function. The system then infers conclusions based on a set of fuzzy control rules. Finally, a unique behavior is selected through the defuzzification interface. Our experiments have shown the advantages of the fuzzy control approach for programming a mobile robot in environments with unknown static objects.

This paper defines a behavior-decision fuzzy controller and reports the results from computer simulation as well as experiments with real robots. In Section 2, we will describe the architecture of the fuzzy controller for behavior-based robots. The detailed design of the fuzzy controller, including the database, the rulebase, and the decision-making logic will be given in Section 3. Section 4 summarizes the results from a set of experiments, and the performance of our fuzzy controller is compared with several existing navigation algorithms.

*This research was supported in part by the ROC National Science Council under grant NSC82-0422-E002-046. The authors can be reached at jyhsu@csie.ntu.edu.tw.

II. ARCHITECTURE

In general, a behavior interacts with the world based on a direct mapping from its sensory data to actions. For example, a light-seeking behavior will choose to move toward the direction with the strongest light. Similarly, an obstacle-avoidance behavior will choose a direction with the lowest probability of having an obstacle. In addition to strictly reactive behaviors, our framework also allows behaviors that utilize simple models of the world.

The current implementation includes behaviors for obstacle avoidance, path tracking, localization, and path planning. The architecture is as shown in Figure 1.

Fig. 1. Navigation by fuzzy control of behaviors.

This architecture can be generalized to perform more complex tasks than simple navigation by including more behaviors, where each behavior can perform a specialized task. At each cycle, a single behavior will be selected for execution. The individual behaviors are described below.

Path tracking: Given a sequence of subgoal points specifying a planned path, this simple behavior compares its current odometer readings with the position of its next subgoal, and moves in the corresponding direction [3].

Collision avoidance: Avoiding collisions with both known and unknown obstacles should be a basic competence of a mobile robot, which is usually equipped with sensors to detect obstacles in its immediate neighborhood. Because sensors can be inaccurate, obstacle detection based on a single sample of sensory information is not very reliable. A heuristic and efficient method, Vector Field Histogram (VHF) [5], is chosen to achieve real-time obstacle avoidance. The VHF method creates a probability distribution of obstacles by continuously and rapidly sampling each sensors while the robot is moving. Each sampling will

increase the certainty value of any cell and its neighboring cells corresponding to the actual location(s) of obstacles. The robot will choose to move in a direction closest to goal and whose obstacle density is below a certain threshold.

Localization: Dead reckoning is typically used to compute the position of a robot during path tracking and other behaviors. However, due to cumulative errors in its motor control, a robot that navigates by dead reckoning alone will eventually lose track of its correct position. The problem can be prevented by periodically re-establishing the absolute position of the robot using localization techniques [6], [7]. The localization behavior works by building an environment model based on its current sensor readings. The model is then transformed and matched with an existing model (such as a given map) to identify potential position(s) of the robot, which are then clustered based on their distances. The average values of the largest cluster will be taken to be the actual position of the robot.

Path planning: This behavior module finds a free path for the robot that avoids all known obstacles. The general problem of path planning is computationally intractable, and many seemingly simple classes of planning problems have been shown to be PSPACE-hard or worse [9]. Given incomplete information and unpredicted changes in the environment, the utility of the "perfect" path is only tentative. Therefore, it makes more sense to use an approximate but efficient method for finding a path. The *numerical potential field* method NF1 introduced in [4] was employed in our system. The time complexity of the planning algorithm is $O(k)$, where k is the number of rectangloid grids in the configuration space.

Given a goal position, the path planning module can generate a collision-free path based on information contained in the map. Since an exact world model is not available in most application, the planned path will be considered as recommended rather than mandatory. The navigation controller should take the current sensory data into account and select the appropriate behavior according to the fuzzy rule base.

III. FUZZY CONTROLLER DESIGN

Our fuzzy controller is a multiple input, single output (MISO) system, which is divided into three parts: **database**, **rulebase**, and **decision-making logic**. The former is used to define the membership functions of fuzzy sets, the latter two are used to define the relation between the input fuzzy sets and the output fuzzy set, and to generate a unique result as the input to the controlled system.

A. Database

There are five factors affecting behavior selection in the current implementation. The input linguistic variables and their fuzzy sets are described as follows (more detailed definitions can be found in [11]):

- PA – the possibility that the robot can pass without touching any obstacle when moving forward to its next subgoal. Its fuzzy sets are *very high*, *high*, *normal*, *low*,

and *very low*. The value of PA combines both sonar and infrared readings.

- SI – the difficulty of the current situation. Its fuzzy sets are *very difficult*, *difficult*, *normal*, *easy*, and *very easy*.
- SD – the difference between real sensor readings and predictive sensor readings. The fuzzy sets are *very small*, *small*, *medium*, *large*, and *very large*.
- DS – the distance from the robot to the next subgoal. Its fuzzy sets are *very near*, *near*, *medium*, *far*, and *very far*.
- TL – the time interval since the last localization. The fuzzy sets are *very short*, *short*, *medium*, *long*, and *very long*.

The fuzzy sets are defined in terms of certain computed numeric values. For example, the SI values is defined as:

$$SI_value = \sum_{i=1}^{16} \#(\text{infrared}_i < 15) + \sum_{j=1}^{16} \#(\text{sonar}_j < 30)$$

where # is the number of sensors whose readings satisfied the condition. Similarly, the SD value is defined as:

$$SD_value = \sum_{i=1}^{16} (\text{infrared}_i - \text{predict_infrared}_i)^2 + \sum_{j=1}^{16} (\text{sonar}_j - \text{predict_sonar}_j)^2$$

The single output linguistic variable is behavior (BH). Its fuzzy sets are PT (path tracking), OA (obstacle avoidance), RP (replanning), and LO (localization). Each fuzzy set of BH is a fuzzy singleton.

B. Rulebase

All the rules in our rulebase are state evaluation fuzzy control rules. They are of the form:

$$R_i : \text{if } x \text{ is } A_i \text{ and } \dots \text{ and } y \text{ is } B_i \text{ then } z \text{ is } C_i$$

where R_i is the i -th rule in the rulebase, x and y are input linguistic variables, and A_i and B_i are their fuzzy sets, respectively. The output linguistic variables z has C_i as its fuzzy set.

The initial design and subsequent modifications of the fuzzy control rules are heuristic and based on experience. It is often necessary to go through a trial-and-error process. The initial rules are constructed based on our knowledge about navigation control. The set of rules were thoroughly tested to identify their shortcomings and the database and rulebase definitions were repeatedly modified and improved.

The final rulebase contains the following seven rules, which are used in the decision-making process.

Conditions	Consequence
IF PA is very low	THEN BH is OA
IF PA is low and DS is far	THEN BH is OA
IF SD is large	THEN BH is LO
IF SD is medium and TL is long	THEN BH is LO
IF SI is hard and TL is medium	THEN BH is LO
IF PA is very low and TL is short	THEN BH is RP
IF PA is low and TL is short	THEN BH is RP

C. The Decision-making logic

The Mamdani's minimum operation rule is used as the implication function in our implementation. For any given rule in the rulebase:

$$R_i : \text{IF } x \text{ is } A_i \text{ and } \dots, y \text{ is } B_i \text{ THEN } z \text{ is } C_i$$

Let $\mu_{A_i}(u), \dots, \mu_{B_i}(v)$ and $\mu_{C_i}(w)$ be the degrees of membership of $A_i, \dots, B_i$ and C_i , respectively, where the input values of the linguistic variables $x, \dots, y$ are $u, \dots, v$, and w is an ambiguous element of C_i . Let $\vee$ be the maximum operation and the $\wedge$ be the minimum operation. The degree of membership function of C_i due to R_i is computed by:

$$\forall w, \mu_{C_i'}(w) = (\mu_{A_i}(u) \wedge \dots \wedge \mu_{B_i}(v) \wedge \mu_{C_i}(w))$$

The membership function μ_C of the inferred consequence C is defined, in a point-wise fashion, by

$$\forall w, \mu_C(w) = \bigvee_{i=1}^n \mu_{C_i'}(w).$$

A simple defuzzification strategy is adopted to choose the behavior that has the highest degree of membership. The degree of membership for the behavior PT is fixed at 0.5, so that it is chosen as the default output if none of the other consequences has a degree of membership higher than 0.5. The fuzzy controller executes an interactive control loop continuously until the goal is achieved.

IV. EXPERIMENTAL RESULTS

The fuzzy controller has been developed on a Sun Sparc station, and the resulting system was tested both on a real robot and using a simulator.

A. Robot : Nomad 200

The Nomad 200 robot has four sensory modules: tactile, infrared, ultrasonic, and a 2D laser ranger. The last module was not used in the experiments. Its drive system utilizes a two-servo, three-wheel synchronous drive mechanical system which has a zero gyro-radius. The user control program can be executed on a SUN workstation or a Macintosh computer, which communicates with the robot via a radio

modem to send control commands and to receive sensor data. The robot's bumper consists of two rings of pressure sensitive switches, with ten switches on each ring. On the robot's body is mounted a ring of 16 independent infrared sensors, ranging from 0 to 36 inches; similarly mounted a ring of 16 sonar sensors, ranging from 17 to 255 inches. The maximum translational speed of Nomad 200 is 4 feet per second, and the maximum rotational speed is 60 degrees per second.

The fuzzy controller was used successfully to control the Nomad 200 navigating in the hallways around our laboratory. The robot was able to reach its goal positions without colliding with unknown objects.

B. Simulation

In order to verify and measure the performance of our fuzzy controller, it was further tested on a set of 200 randomly generated environments. A small degree of motor and sensor uncertainties were included in the simulation. For comparison, three existing navigation algorithms Bug1, Bug2(L), and Bug2(R) [12], [13] were also implemented to run on the same examples. The resulting traces from our system and those of the others are shown in Figure 2. In this case, although all the algorithms can achieve their goals, the fuzzy controller was the most efficient.

To illustrate that the relatively better performance of the fuzzy controller was not coincidental, the experiment was repeated for all 200 different environments, and the elapsed time for reaching the goal position was recorded for each run. Table 1 shows the average time for each method to achieve its goal, given that the ratio of unknown obstacles vs. known obstacles was 0%, 60%, 100%, and 140%. The overall performance of the fuzzy controller is about 25% to three times better than the other methods.

TABLE I
AVERAGE TIME (SEC) FOR REACHING THE GOAL.

unknown/known obstacles	0%	60%	100%	140%
fuzzy controller	167	189	223	235
Bug 1	512	553	602	650
Bug 2(R)	277	327	361	363
Bug 2(L)	227	262	305	297

Unfortunately, the performance was gained at the expense of completeness. While the other three algorithms are guaranteed to find a path if one exists, the fuzzy controller may fail. Table 2 shows the failure rate for the fuzzy controller given the different ratios of unknown obstacles.

The experiments illustrated that our fuzzy controller can achieve its goal over 90% of the time even when unknown obstacles outnumbered known obstacles in the environment. It should be noted that the completeness of the other navigation algorithms is obtained under the strong assumption of perfect sensing and control, which is very unlikely in the real world.

TABLE II
FAILURE RATE

unknown/known obstacles	0%	60%	100%	140%
fuzzy controller	0%	1.8%	1.7%	7.9%

V. CONCLUSION

In this paper, we presented a fuzzy controller for behavior-based mobile robots. The fuzzy controller allows run-time arbitration of the next behavior to be executed in a tight interaction loop with the environment. It is robust with respect to unknown obstacles and motor uncertainty. The fuzzy controller satisfies two requirements: reacting to its local dynamic environment and achieving its global goal, which are not achievable at the same time by either centralized or reactive methods alone.

REFERENCES

- [1] R.A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1), pp. 14-23, 1986.
- [2] M.J. Mataric. Integration of representation into goal-driven behavior-based robots. *IEEE Transactions on Robotics and Automation*, 8(3), pp. 304-312, 1992.
- [3] R.C. Arkin. Motor schema-based mobile robot navigation. *The International Journal of Robotics Research*, pp. 92-112, August, 1992.
- [4] J. Barraquand, B. Langlois, and J.-C. Latombe. Numerical potential field techniques for the robot path planning. *IEEE Transactions on Systems, Man, and Cybernetics*, pp. 224-241, 1992.
- [5] H. Borenstein and Y. Koren. Real-time obstacle avoidance for fast mobile robots. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(5), pp. 1179-1186, 1989.
- [6] M. Drumheller. Mobile robot localization using sonar. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 9(2), pp. 325-332, 1987.
- [7] A.A. Holenstein, M.A. Muller and E. Badreddin. Mobile robot localization in a structured environment cluttered with obstacles. *Proceedings of IEEE Conference on Robotics and Automation*, pp. 2576-2581, Nice, France, 1992.
- [8] L.P. Kaelbling and S.J. Rosenschein. Action and Planning in Embedded Agents. *Robotics and Autonomous Systems*, 6, pp. 35-48, 1990.
- [9] J.-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.
- [10] C.C. Lee. Fuzzy logic in control system: fuzzy logic controller, part 1& 2. *IEEE Transactions on System, Man, and Cybernetics*, 20(2), pp. 419-435, 1990.
- [11] D.C. Lo. *A Fuzzy Controller for Behavior-Based Mobile Robots*, MS Thesis, CSIE, National Taiwan University, June, 1993.
- [12] V.J. Lumelsky and A.A. Stepanov. Path planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape. *Algorithmica*, Springer-Verlag, 1987.
- [13] A. Sankaranarayanan and M. Vidyasagar. A new path planning algorithm for moving a point object amidst unknown obstacles in a plane. *Proceedings of 29th IEEE Conference on Decision and Control*, pp. 1111-1119, 1990.
- [14] R.P. Sobek and R.G. Chatila. Integrated planning and execution control for an autonomous mobile robot. *Artificial Intelligence in Engineering*, 3(2), pp. 103-113, 1988.
- [15] M. Sugeno and M. Nishida. Fuzzy control of model car. *Fuzzy Sets and Systems*, 16, pp. 103-113, 1985.
- [16] M. Sugeno. An introductory survey of fuzzy control. *Information Sciences*, 36, pp. 59-83, 1985.

(a) Fuzzy controller

(b) Bug1 algorithm

(c) Bug2(R) algorithm

(d) Bug2(L) algorithm

Fig. 2. A sample comparison among four navigation algorithms