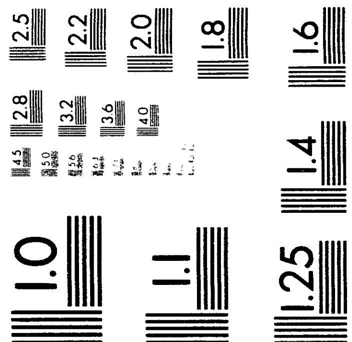
2.5
2.2
2.0
1.8
1.6
2.8
3.2
3.6
4.0
1.4
1.0
1.1
1.25

1 of 1

Conf-931107--2

Sensor-Based Navigation of a Mobile Robot Using Automatically Constructed Fuzzy Rules*

Yutaka Watanabe and François G. Pin
Autonomous Robotic Systems Group
Oak Ridge National Laboratory
P.O. Box 2008
Oak Ridge, TN 37831-6364

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

"The submitted manuscript has been authored by a contractor of the U.S. Government under contract DE-AC05-84OR21400. Accordingly, the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes."

Contact: Dr. François G. Pin, Oak Ridge National Laboratory, P.O. Box 2008, Oak Ridge, TN 37831-6364; Telephone: (615)574-6130; Fax: (615)574-7860, E-mail: pin@ornl.gov

Paper to appear in Proceedings of: ICAR '93, The Sixth International Conference on Advanced Robotics, Tokyo, Japan, November 1–2, 1993, special session on "Advanced Technology for Intelligent Robotic Systems"

* Research sponsored by the Engineering Research Program, Office of Basic Energy Sciences, of the U.S. Department of Energy, under contract DE-AC05-84OR21400 with Martin Marietta Energy Systems, Inc.

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

Sensor-Based Navigation of a Mobile Robot Using Automatically Constructed Fuzzy Rules

Yutaka Watanabe and François G. Pin
Autonomous Robotic Systems Group
Oak Ridge National Laboratory
P.O. Box 2008
Oak Ridge, TN 37831-6364

Abstract

A system for automatic generation of fuzzy rules is proposed which is based on a new approach, called "Fuzzy Behaviorist," and on its associated formalism for rule base development in behavior-based robot control systems. The automated generator of fuzzy rules automatically constructs the set of rules and the associated membership functions that implement reasoning schemes that have been expressed in qualitative terms. The system also checks for completeness of the rule base and independence and/or redundancy of the rules to ensure that the requirements of the formalism are satisfied. Examples of the automatic generation of fuzzy rules for cases involving suppression and/or inhibition of fuzzy behaviors are given and discussed. Experimental results obtained with the automated fuzzy rule generator applied to the domain of sensor-based navigation in *a priori* unknown environments using one of our autonomous test-bed robots are then presented and discussed to illustrate the feasibility of large-scale automatic fuzzy rule generation using our proposed "Fuzzy Behaviorist" approach.

1. Introduction

Several research groups have studied approximate reasoning techniques as a means to mimic human reasoning capabilities in the sensor-based control of complex systems (e.g. see examples in [1], [2]); in particular, several researchers have investigated the use of fuzzy logic ([3], [4]) for the autonomous sensor-based navigation of mobile robots in unstructured environments (e.g., see [5], [6], [7]) including outdoor environments [8],[9]. In all these applications, the sensor-based decision making process has been implemented as a set of fuzzy rules which, together, express the desired navigation behaviors of the robot for various combinations of the sensory input data. Very successful results have been achieved when the number or complexity of the behaviors embodied in the fuzzy rule base was small. When the number or complexity of the behaviors increased, however, and/or the perception system grew more sophisticated (i.e., more sensory input data is provided), the typical difficulties encountered with large rule base systems emerged: the lack of established formalism for the development of rule bases, in particular to provably address completeness, interaction, and redundancy of the rules, made the actual coding of the fuzzy rules an iterative empirical process, requiring lengthy trial-and-error experiments.

In an attempt to alleviate this general shortcoming of rule-base system development, we recently proposed a "Fuzzy Behaviorist" approach [7],[8],[9],[10] which provides a formalism for the development of fuzzy rule systems for the behavior-based control of autonomous robots. This approach and its associated formalism have introduced features which allow for automatic generation of fuzzy rule bases, once the desired set of behaviors (i.e., the overall decision-making strategy) has been expressed *qualitatively* by a user. The objective of this paper is to present our approach and recent results in developing this automatic system for generation of fuzzy rules, including some experimental results obtained with the system in the context of sensor-based navigation in *a priori* unknown environments using one of our autonomous test-bed robots [11],[12]. The next section briefly reviews the features of our proposed "Fuzzy Behaviorist" approach which directly support automatic generation of fuzzy rules, while the following sections present the system's approach, development, and experimental results. The last section includes our conclusion and proposed future work on the automated system.

2. Fuzzy behaviorist approach for sensor-based robot control

The basic premises underlying our proposed Fuzzy Behaviorist approach [7],[9],[10] and pertinent to the development of the automated system for fuzzy rule generation are as follows:

- Each action of the robot results from the concatenation of elemental behaviors.
- Each elemental behavior is a direct mapping from a single stimulus mode to a single output control.
- Each behavior is represented by one or a set of fuzzy rules which are defined by the membership functions of the rule's antecedent (stimuli) and consequence (output controls).
- Each mode of stimulus corresponds to a dimension of the input space and is independent of other stimulus modes.
- Stimuli (antecedents of rules) are represented by membership functions on their respective input dimension, with membership $\mu(x) = 1$ corresponding to regions where the behavior is triggered with "full strength." Input regions with membership $0 < \mu(x) < 1$ correspond to conditions where there is a "tendency" for the behavior to trigger.

- For behaviors effecting the same output control dimension, the regions of "full strength" stimulus of the various behaviors do not overlap in the multidimensional input space. However, regions of tendencies can overlap, resulting in several behaviors possibly triggering for an input data in these regions.
- Each type of input data provided by the sensors is fuzzified with a membership function expressing, as a *possibility* distribution, the uncertainty associated with the specific measurement or calculation.
- Triggering of any behavior takes place when the current input data and the antecedents of the behavior's fuzzy rules have non-empty fuzzy intersection. The triggering strength transferred to the output control is dictated by the fuzzy intersections of the input data and the rule antecedents.
- When several fuzzy rules trigger simultaneously for a given set of input data, all resulting consequences are taken into account in the concatenated output membership function, which is obtained directly from the laws of combinatorial inferencing of the Fuzzy Sets Theory [3],[4]. Consequently, no "conflict resolution" or "arbitrator" between behavior's output is required in a Fuzzy Behaviorist system, therefore alleviating one of the major hurdles which the Behaviorist community has long been struggling with.
- In the current system discussed below, a typical "center of area" defuzzification scheme is used to generate "crisp" control set points for those output variables that represent direct actuator commands.

With these features, rule bases embodying the elemental fuzzy behaviors can be very easily generated (e.g., see [7], [9]) or augmented with additional behaviors to handle situations of increasing complexity [9],[10]. In this latter case, however, a very important aspect of the formalism which needs to be emphasized here is the requirement for independence and non-overlapping of the full strength regions of the behaviors. This often leads to the newly added behaviors having to "be dominant," or "be dominated by," some of the existing behaviors in one or more regions of the input space. This requirement simply expresses the fact that for a single output control, only one "full strength" action can be commanded for any given point (stimulus) in the input space. Section 3 discusses how the automated system implements the dominance between behaviors using the suppression or inhibition mechanisms that are inspired from two concepts which we illustrate here using a simple example: consider two behaviors acting on the *same* output control, say the gripper of a robot or, for analogy, the two object-holding fingers of a child. Initially, the robot or the child only has one behavior which is a "don't get burned" behavior and which could consist of one fuzzy rule expressing that IF *the object being held is hot* THEN *release the object*. The input dimension for this behavior is "temperature of object being held," and a possible membership functions for the antecedent of the rule is shown on the left side of Fig. 1. Suppose that the child is being taught the value of things or assume that the robot is being given a new perception device so that it can recognize the value of objects. A new behavior could be given to the robot or to the child stating that IF *the object being held is expensive* THEN *don't release the object*. The input dimension for this behavior is "value of the object being held" and a possible membership function for the rule antecedent is shown on the bottom of Fig. 1. Taken separately, the two behaviors are fully independent since each has a different one-dimensional input space. They could therefore be developed and tested independently of each other. If these two behaviors were merged to create a more complex system, then the new input space for the system would become two-dimensional. Within that new system, the two behaviors are still "independent" since they trigger from stimuli on different input dimensions. However, every time the robot or the child will handle one of great grandmother's priceless china cups or plates filled with very hot tea or soup, the implicit dominance of one behavior over the other in their overlapping region of the input space will be readily apparent: the cup or plate will, or will not, be dropped, signifying respectively that the "don't get burned" behavior dominates, or is dominated by, the "don't drop valuable objects" behavior at that particular point in the input space.

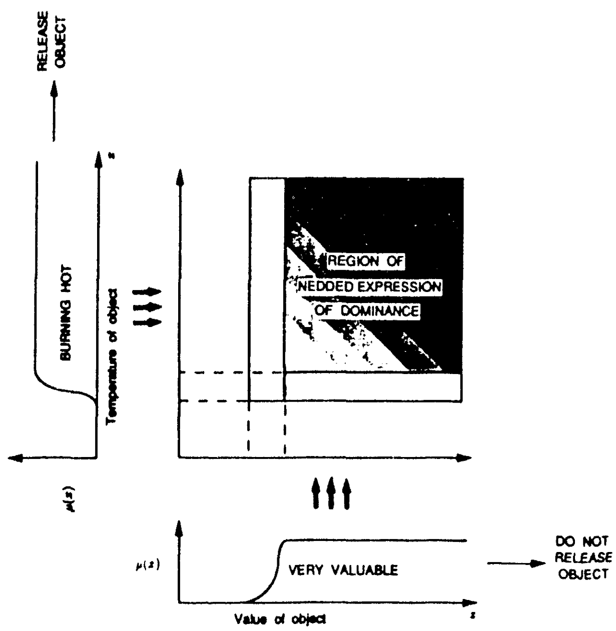

Fig. 1. Example of rule base augmentation through merging of behaviors operating on different input dimensions and the same output dimension.

It is clear that for both the child and the robot, some instruction, reinforcement learning, experience, etc. may cause the dominance process and its resulting outcome to vary over time. However, if some "mistakes" and/or "trial and error experiments" are not allowed to the child or the robot, then an indication of what behavior dominates the other in what region of their overlapping areas in the input space, must be given to the child or the robot at the time the second behavior is added and merged with the other in the system's reasoning module (see Fig. 1). In the automated system, this is accomplished through the "suppression" mechanism, in which the output membership function of the dominant behavior is modified, so that its weight will appropriately overpower that of the other behavior in the c.g. calculation. As will be discussed in Section 3, if the membership function can not be made adequately overpowering, due to previous dominances and/or suppression requirements with respect to other behaviors, then the system uses the concept of "inhibition" to express the behavior's dominance. Applying inhibition within the overlapping region of input space consists in partially truncating the input membership function of the "weaker" behavior so that the dominant behavior always triggers with a greater strength. Although both suppression and inhibition mechanisms result in expressing the desired dominance, their concepts are quite different: one basically operates on the relative weight of the behaviors in the output space while the other modifies the triggering conditions of the behaviors in the input space. This difference, which is transparent in the automated system, becomes significant when one seeks to couple the system with learning modules for refinement of the behaviors and/or of the rule membership functions through reinforcement learning.

3. Automated system

Just like different people may use different strategies, different rules, and different qualitative variables to express their navigation process, and still navigate efficiently "in their own way," several strategies may be used to embody a particular process in a rule base, i.e., there is not a single or unique rule base representation of a given process. For example, a rule base for obstacle avoidance may be built on the basis of an obstacle-distance strategy, as was done in [7], with rules organized and developed for input conditions in which obstacles are very near, near, far, very far, etc.; or on the basis of the direction of obstacles, with rules organized and developed for obstacles located on the right, center, or left of the travel direction. Because of the requirement of our Fuzzy Behaviorist approach for each behavior to trigger from a single input dimension, the expression into rules of various possible strategies may appear quite different, even though the overall process and resulting actions of the robot may be similar. In the automated system, the user inputs the strategy for the rules in a "qualitative" form using the format shown in Fig. 2.

```
name : RN
suppressing list :  G? LF
E                :  50
Outputs          :  20 -
Inputs           :  - Near - -
```

Fig. 2. Input format for the automated rule generation system.

The five-line format in Fig. 2 describes one rule, with the first line giving the "reference name" of the rule, the second line listing the names of the rules or behaviors which are suppressed (or inhibited) by this rule, the third line giving the suppression parameter E (which will be described in the following paragraphs), and the last two lines specifying what the desired input and output of the behavior, or rule within the behavior, are. In the current version of the automated system, each rule is assumed to be of the form

$$\text{IF } (A \text{ is } A_1 \text{ and } B \text{ is } B_1 \text{ and } C \text{ is } C_1 \text{ and } D \text{ is } D_1) \text{ THEN } (E \text{ is } E_1 \text{ and } F \text{ is } F_1)\ , \tag{1}$$

therefore operating on four input and two output channels. Although extension to any number of input and output channels is possible, this configuration was chosen in the initial version of the automated system because it corresponds to what is available on the custom-designed VLSI fuzzy inferencing chips and boards (see [7], [13], [14]) which we utilize in our experimental work. The line labeled "outputs" indicates which of the two outputs is effected by the behavior, with the number, expressed in bit numbers over the scaled output ranges of 0 to 63 bits (see [7], [13], [14]) specifying the desired center of mass of the membership function of the output (E_1 or F_1 in Eq. (1)) of the rule. The line labeled "inputs" specifies which one of the four input channels is the behavior's input dimension, and the qualitative name indicates which fuzzy set constitutes the antecedent, (A_1, B_1, C_1 or D_1 in Eq. (1)) of the rule on the input dimension. The membership functions defining the input related fuzzy sets are defined by the user on a behavior-by-behavior basis (e.g. see [7] or [9]), and can be stored according to their "name" in a "membership function library."

When the user has listed all the rules of the desired behaviors in the format of Fig. 2, the automated system can generate a "skeleton" of the rule base and check if it verifies the input-related requirements of the approach (see Section 2). In particular, the system constructs the four-dimensional input spaces for each of the two output dimensions (see the example 2-dimensional space in Fig. 1), so that it can evaluate completeness of and redundancy in the rule base and report all instances to the user. For any region of incompleteness, i.e., regions of the input space not covered by any of the behaviors stimuli, the user decides on either the addition of a behavior to cover these possible stimuli, extension of the current behaviors (through extension of their input membership function) to include these input regions, or no modification if input data within these uncovered regions or "blind spots" are never expected to occur (for example if these regions correspond to values outside the operating range of the sensors). For the regions of redundancy, i.e., areas where stimuli from two or more behaviors are overlapping, the system reports every rule for which a dominance has not been specified but may be required because of the input overlapping. The user can then interactively add to or modify the dominance specifications in lines 2 and 3 of each rule, until all requirements of the approach are verified and all needed information has been input to express the desired dominances in the rule base. The actual generation of the rule base, including the suppression and/or inhibition mechanism, can then proceed as follows: initially, all rules are given a "standard" output membership function equal to 1 over a width of one bit, centered at the bit value expressed in line 4 of the "qualitative" expression of the rule (see Fig. 2). The system checks the sets of rules that are effecting the *same* output dimension. If no suppression mechanism has been expressed between the rules because dominance is not necessary, then the output membership functions are unconstrained and they remain at their "standard" value. If a dominance has been expressed between two or more rules, using lines 2 and 3 of either rule, then an overlap exists in the input space and the rules need to be modified to reflect the dominance. In the automated system, the dominant rule is the one that is modified if suppression is possible, otherwise the dominated rules are modified using inhibition, as explained in the following paragraphs.

Given a rule i, and its output function $\mu_i(y)$ on the y output dimension, the center of mass y_i and weight m_i of its output are given by

$$y_i = \int y\mu_i(y)dy \tag{3}$$

$$m_i = \int \mu_i(y)dy \tag{4}$$

and, by definition, $m_i > 0$ and $y_{\min} \leq y_i \leq y_{\max}$, where $y_{\min}$ and $y_{\max}$ are the extremum values of the y output dimension. Assume that two rules, which control the same output y, have respective output membership functions $\mu_i(y)$ and $\mu_j(y)$, with corresponding centers of mass and weights given by Eqs. (3) and (4). When both rules trigger under the same input conditions, the resulting output membership function is the union (or max operation) of $\mu_i(y)$ and $\mu_j(y)$ and, since a "center of gravity" defuzzification scheme is used on the VLSI chips [11], [14], [17], [18], the overall output y_o is

$$y_o = \frac{m_i y_i + m_j y_j}{m_i + m_j} \tag{5}$$

From Eq. (5), it is clear that the *relative* value of the output weight of the rules can change the output y_o. In particular, if the weight of one rule is strong enough, the output y_o can be "attracted" *within* the output membership function of the rule, *even though the other rule still "works" and contributes to the overall output*. This constitutes the basis for implementing the suppression mechanism.

3.1. Suppression mechanism

In the general case, a rule may be required to suppress a set S of other rules i, $i \in S$. The suppression mechanism can be expressed as the requirement that, when all the rules h and i, $i \in S$, trigger, their combined output y_o must fall within an "allowable error" ϵ_h of the center of mass y_h of rule h. The suppression condition thus can be written as

$$|y_o - y_h| \leq \epsilon_h \tag{6}$$

This allowable error, ϵ_h, can take any positive value between 0 and 63 (the maximum bit number of the output membership function range), and is the user-specified number appearing in line 3 of the qualitative rule description shown in Fig. 2. Typically, a small value ϵ_h indicates a "strong" dominance of rule h over rules i, while a large ϵ_h represents a "weak" dominance or what we will refer to as a "tendency."

Since

$$y_o = \frac{m_h y_h + \sum_{i\in S} m_i y_i}{m_h + \sum_{i\in S} m_i} \tag{7}$$

we have

$$\frac{\sum_{i\in S} |y_h - y_i| m_i}{m_h + \sum_{i\in S} m_i} \leq \epsilon_h \tag{8}$$

from which a minimum value for m_h can be calculated. Since the distances between the centers of mass $|y_i - y_h|$ are always less than or equal to $Y_r = y_{\max} - y_{\min}$, selection of m_h as

$$\left(\frac{Y_r}{\epsilon_h} - 1\right) \sum_{i\in S} m_i \leq m_h \tag{9}$$

guarantees that rule h suppresses the other rules i, $i \in S$. From m_h, and the selected shape (rectangular in the current system) of the output membership function, the width of, and/or the full $\mu_h(y)$, can be easily determined.

3.2. Inhibition mechanism

In some cases, the suppression mechanism described above may not be feasibly implementable because of the discretization of the membership functions in 64 bits. This will typically occur when ϵ_h is specified very small or equal to zero, or if several dominance mechanisms need to be implemented within the rule base resulting in progressively large weights calculated from Eq. (9). In this case, the inhibition mechanism is used instead of the suppression mechanism, and the dominance of rule h over rules i, $i \in S$, is forced by appropriately truncating the *input* membership functions of the rules i so that these rules do not trigger when rule h does. IF (A_1, B_1, C_1, D_1, E_1, F_1) defines the rule (see Eq. (1))) which dominates rule (A_2, B_2, C_2, D_2, E_2, F_2), and B_2 is the overlapping input to be truncated, then the truncated condition B_2^* is expressed as $B_2^* = B_2 \cap \overline{B_1}$, where $\overline{B_1}$ represents the complement of condition B_1. This effectively removes from the dominated rules' input the regions with membership function $\mu(x) = 1$ that are overlapping.

4. Experimental results

The automated fuzzy rule generation system was utilized to generate rule bases for the sensor-based navigation of an autonomous robot, and the resulting rule bases were tested on navigation problems in a variety of *a priori* unknown environments. In this section, sample results from these experiments are presented to illustrate the automatic rule generation process including the suppression and inhibition mechanisms. Actual paths taken by the robot in test environments are displayed to illustrate the sensor-based navigation behaviors resulting from various rule bases. Note that, except for specifying the goal, no information on the environment is given *a priori* to the robot, nor is any map generated during the motion. The navigation, therefore, is purely reactive, involving no memory or real-time information storage of any type. As discussed in details in Refs. [7], [8], [9], the four sensory input to the fuzzy inferencing system are the goal direction (or target direction), which is updated at each loop rate using the odometry system; and the minimum distances to obstacles obtained using groups of acoustic range finders in three 75° wide sectors at the left, center, and right of the robot travel direction. The two output of the inferencing system are the commands for turn increments and speed of the robot. Thus, navigation behaviors with these data will involve Turn Control (T.C.) and Speed Control (S.C.) from Goal Orientation (G.O.) and Obstacle Proximity (O.P.) input.

As mentioned previously, several rule bases, representing various "strategies," may be developed to solve a complex problem or to embody a complex behavioral process. Figure 3 shows the rules for one of the rule bases we investigated for sensor-based navigation, in which front obstacle proximity is only used for speed control, while side obstacle proximity is only used for turn control. The behaviors shown in Fig. 3 are thus as follows:

G.O.	$\longrightarrow$ T.C.	(3 rules)
G.O.	$\longrightarrow$ S.C.	(1 rule)
"front" O.P.	$\longrightarrow$ S.C.	(4 rules)
"left" O.P.	$\longrightarrow$ T.C.	(4 rules)
"right" O.P.	$\longrightarrow$ T.C.	(4 rules)

Each of the 16 rules in the figure is displayed as a vertical arrangement of six graphs of membership function. The four top graphs in each rule show the membership functions corresponding to the input variables A_1, B_1, C_1, D_1 in Eq. (1) (the direction angle to the goal and the distances returned by the left, center, and right "wide blurry eyes") while the bottom two correspond to the output variables E_1 and F_1 (turn command and speed command). Note that every behavior, and consequently every rule, represents a mapping from only one input dimension to only one output control. In the other input dimensions, membership functions uniformly equal to 1 over their entire range signify that the behavior is not effected by stimuli in these dimensions or, in other words, that the data in these dimensions is "unimportant" or "can be anything." For the output

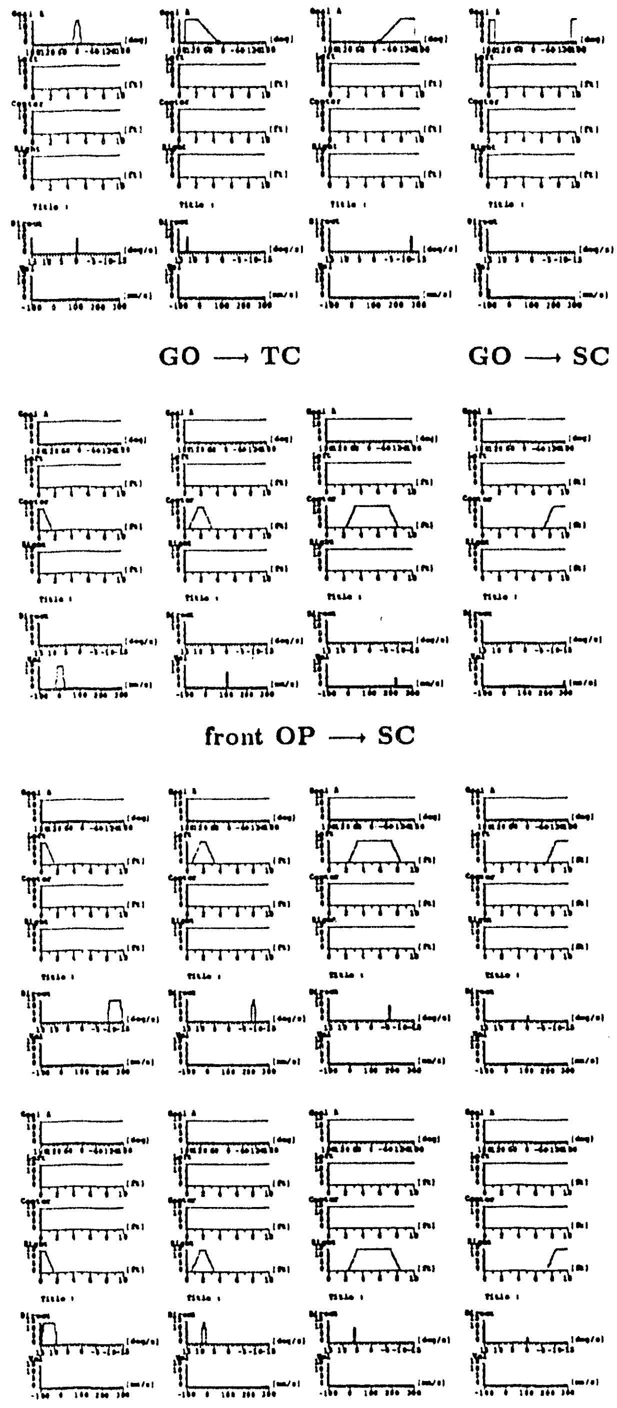

Fig. 3. Rule base for sensor-based navigation.

controls that are not effected by a given behavior, a membership function uniformly equal to 0 over the entire range of that output ensures no contribution of the behavior to that particular actuator control. The three rules of the G.O. $\longrightarrow$ T.C. behavior simply express that if the goal is to the center, left, or right respectively, then the robot should make a zero, positive (i.e., left), or negative (i.e., right) increment of turn, respectively, and there is no command for speed change. The G.O. ⟶ S.C. rule states that if the goal is toward the back sector of the robot, the robot should slow down (i.e., apply a negative contribution of speed). The four rules of the front O.P. ⟶ S.C. behavior express that if the obstacles in front are very far, then a speed close to the maximum possible can be applied, and that the closer the frontal obstacles, the slower the robot should go, eventually stopping when the obstacles are dangerously close. Note that the weight of the velocity command (i.e., the membership function) increases with "increasing danger." Similarly, the eight rules of the O.P. ⟶ T.C. behaviors express that the closer the obstacles on a side, the greater the increment of turn in the opposite direction and the "heavier" the turn command should count in the output control calculation. Note the large weight of the output membership function of the "very near" O.P. ⟶ T.C. rules, which results from this behavior having been selected as suppressing the G.O. ⟶ T.C. behavior, i.e., expressing that when obstacles are very close, their avoidance is always of greater importance than tracking the goal. It is clear that without this expression of dominance, the G.O. ⟶ T.C. and O.P. ⟶ T.C. behaviors would often result in dead-lock or oscillatory situations in which the robot would not turn at all or would oscillate between two orientations. This type of situation constitutes one of the very serious drawbacks of the reactive navigation methods using potential field techniques, and has been alleviated here using the suppression mechanism.

Figures 4 and 5 show plots of actual runs made with the robot to illustrate the overall reactive navigation obtained with the automatic generation of fuzzy rules. These plots are also given here to provide an example of the effect on the navigation behaviors which a dominance mechanism (suppression or inhibition) can produce. In the figures, the shaded areas represent the obstacles which were placed in the room, while the path of the robot is illustrated using the succession of circles showing the position of the robot every 20 loop rates. In Fig. 4, the rules shown in Fig. 3 were used, which embody a very strong dominance of the obstacle avoidance (O.P. ⟶ T.C.) rules over the goal tracking (G.O. ⟶ T.C.) rules. Consequently, due to the almost constant proximity of the corridor walls, the suppression mechanism is quite effective in the early part of the run and the robot wanders around for quite a long time, guided principally by obstacle avoidance. It eventually gets positioned ideally to enter the corridor and then turns right, in a direction closest to the goal direction. It follows the corridor, and when reaching the end of the wall, turns left toward the goal. Clearly the dominance of the obstacle avoidance rules over the "move to the goal" behavior may be too strong in this environment. For the sample run shown in Fig. 5, this dominance has been decreased, and a modified rule base has been generated using the automated system. The robot is seen to negotiate the entrance of the corridor much more rapidly because of the greater effect of the goal tracking behavior, resulting in a much shorter run to the goal.

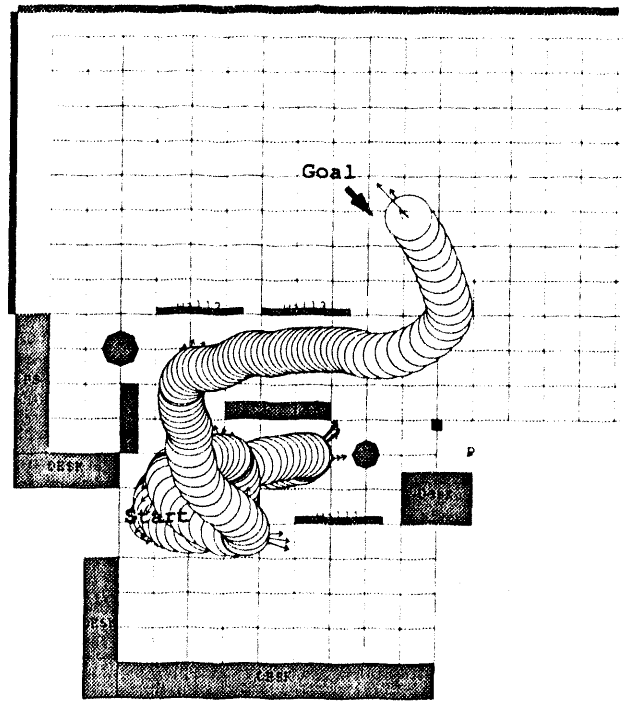

Fig. 4. Actual run of the robot using an automatically generated fuzzy rule base with strong behavioral dominance of obstacle avoidance over goal tracking.

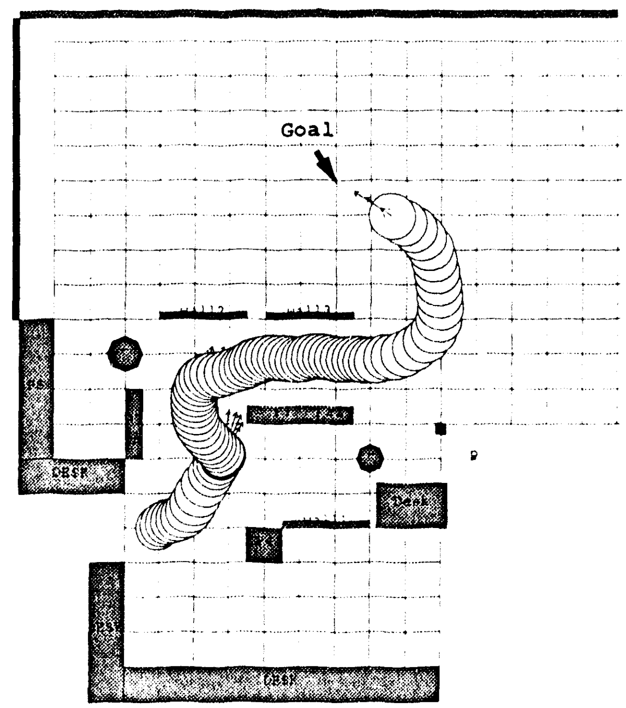

Fig. 5. Same as Fig. 4 with a lesser behavioral dominance of obstacle avoidance over goal tracking.

In both sample runs, note the very similar change in speed when the robot exits the corridor and faces the open space in the upper right section of the environment or, in general, when obstacle proximity varies, which illustrates the effect of the speed control behaviors.

5. Conclusion

An automated system to generate fuzzy rules from the qualitative description of a reasoning process has been developed. The automated system is built on the basis of the Fuzzy Behavior formalism which we proposed to ease the development of fuzzy rule bases embodying "human-like" behaviors in sensor-based decision-making systems. The concepts of suppression and inhibition of behaviors and the inclusion of corresponding mechanisms in the automated system have been described. These mechanisms allow the system to handle situations when potentially conflicting behaviors are merged or progressively added in a single rule base.

Examples of the use of the automated system to generate fuzzy rule bases for the sensor-based navigation of an autonomous robot have been given. Sample runs of an actual robot have been presented to illustrate the navigation behaviors obtained with the automatically generated fuzzy rule base as well as the effect of a change in the inter-behavior dominance expressed through the suppression and/or inhibit mechanism.

References

[1] "Uncertainty in Artificial Intelligence," eds. L. N. Kanal and J. F. Lemmer, North-Holland (1988).

[2] "Industrial Applications of Fuzzy Control," ed. M. Sugeno, North-Holland (1985).

[3] L. A. Zadeh, "Outline of a New Approach to the Analysis of Complex Systems and Decision-Making Approach," *IEEE Transactions on Systems, Man, and Cybernetics* **SMC-3**(1), 28–45 (1973).

[4] L. A. Zadeh, "Fuzzy Logic," *IEEE Computer* **21**(4), 83–93 (1988).

[5] Takeuchi et al., "Fuzzy Control of a Mobile Robot for Obstacle Avoidance," *Information Science* **45**, 231–239 (1988).

[6] J. Yen and N. Pfluger, "A Fuzzy Logic Based Robot Navigation System," in Proceedings of the AAAI Symposium on Applications of Artificial Intelligence to Real-World Autonomous Mobile Robots, Cambridge, Massachusetts, October 23–25, 1992, 195–199.

[7] F. G. Pin, H. Watanabe, J. R. Symon, and R. S. Pattay, "Using Custom-Designed VLSI Fuzzy Inferencing Chips for the Autonomous Navigation of a Mobile Robot," in Proceedings of IROS '92, the 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems, Raleigh, North Carolina, July 7–10, 1992, 790–795.

[8] F. G. Pin and Y. Watanabe, "Driving a Car Using Reflexive Fuzzy Behaviors," in Proceedings of the Second IEEE International Conference on Fuzzy Systems, San Francisco, California, March 28–April 1, 1993, 1425–1430.

[9] F. G. Pin and Y. Watanabe, "Steps Toward Sensor-Based Vehicle Navigation in Outdoor Environments Using a Fuzzy Behaviorist Approach," *International Journal of Intelligent and Fuzzy Systems* **1**(2) (1993). (In press.)

[10] F. G. Pin and Y. Watanabe, "A Fuzzy Behavior Approach for Approximate Perception Data-Based Control of Autonomous Robots," submitted to *IEEE Transactions on Systems, Man and Cybernetics*.

[11] S. M. Killough and F. G. Pin, "Design of an Omnidirectional and Holonomic Wheeled Platform Prototype," in Proceedings of the 1992 IEEE International Conference on Robotics and Automation, Nice, France, May 10–15, 1992, 84–90.

[12] F. G. Pin and S. M. Killough, "A New Family of Omnidirectional and Holonomic Wheeled Platforms for Mobile Robots," accepted for publication in *IEEE Transactions on Robotics and Automation*.

[13] H. Watanabe, W. Dettloff, and E. Yount, "A VLSI Fuzzy Logic Inference Engine for Real-Time Process Control," *IEEE Journal of Solid State Circuits* **25**(2), 376–382 (1990).

[14] J. R. Symon and H. Watanabe, "Single Board System for Fuzzy Inference," in Proceedings of the Workshop on Software Tools for Distributed Intelligent Control Systems (September 1990), 253–261.

DATE FILMED

12/30/93

END