Correspondence

A Fuzzy Logic Based Extension to Payton and Rosenblatt's Command Fusion Method for Mobile Robot Navigation

John Yen and Nathan Pfluger

***Abstract*—David Payton and Ken Rosenblatt have proposed a command fusion method for combining outputs of multiple behaviors in a mobile robot navigation system such that information loss due to command fusion can be reduced. Using linguistic fuzzy rules to explicitly capture heuristics implicit in the Payton-Rosenblatt approach, we have extended their approach to a fuzzy logic architecture for mobile robot navigation in dynamic environments, which is simpler and easier to understand and modify. We have also developed and empirically tested a new defuzzification technique for alleviating difficulties in applying existing defuzzification methods to mobile robot navigation control.**

I. Introduction

The ability for a mobile robot to navigate intelligently in a dynamic environment is needed if robots are to be used in areas other than strictly controlled environments. In order to navigate in a dynamic environment, the robot must be able to deal with the issue of uncertainty and incomplete information about the environment in a timely manner. Several approaches have been developed to address this important issue at various levels in mobile robot planning and control.

One approach at the level of intelligent mobile robot control is to use multiple *behaviors* to generate several control suggestions, one of which is selected by a *command arbitrator* based on the behaviors' priorities. The term *behavior* comes from biology and refers to the reaction of an agent to a given situation. Therefore, a behavior in a mobile robot navigation system usually represents a concern of the robot, such as *follow the path* or *avoid obstacles.* To avoid collisions, the obstacle avoidance behavior is usually given a higher priority than other behaviors.

Even though this command arbitration scheme is simple and effective under most situations, Payton and Rosenblatt have found a serious problem with this approach, which is demonstrated in Fig. 1 [1]. The priority-based arbitration scheme would only have a 50% chance to make the correct turn (i.e. left), since the information regarding path following is not available once the command arbitrator selects the collision avoidance behavior. What we need is a way to fuse multiple control commands recommended by different behaviors.

Observing the limitation of a fixed priority arbitration scheme, Payton and Rosenblatt developed a *command fusion* architecture that allows control recommendations of different behaviors to be directly combined to form multiple control recommendations with different weights, from which a final control command is chosen. This paper describes an extension of their work using fuzzy logic [2], [3].

In the next section, we will first briefly review major approaches to mobile robot navigation and, in particular, Payton and Rosenblatt's command fusion scheme. A brief introduction to fuzzy logic and a discussion on its relationship to the Payton-Rosenblatt (P-R) approach is then given. Section III presents our fuzzy logic based architecture for mobile robot navigation that extends the P-R approach. We describe the simulation results of an implementation of the architecture in Section IV. Finally, we summarize the benefits of our approach and outline the directions of our future work.

Manuscript received September 8, 1991; revised April 16, 1993, and August 5, 1994. This work was supported by NASA grant NGT-50837 and NSF Young Investigator Award IRI-9257293.

The authors are with the Center for Fuzzy Logic and Intelligent Systems Research, Department of Computer Science, Texas A&M University, College Station, TX 77843-3112 USA.

IEEE Log Number 9410386.

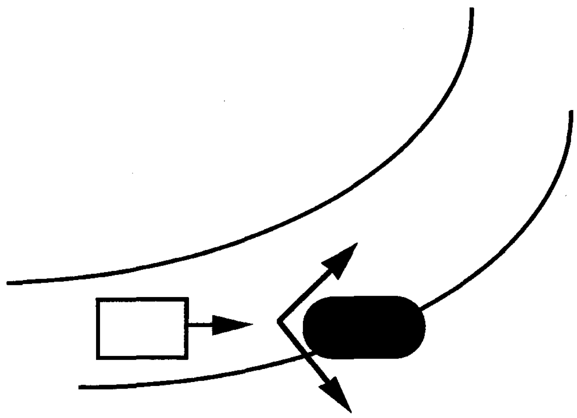

Fig. 1. Which way to turn? Obstacle avoidance does not know.

II. Background

A. Approaches to Mobile Robot Navigation

The basic problem of autonomous mobile robot path planning and control is to navigate safely to one or several target locations.[1] Existing approaches to mobile robot navigation can be classified into three categories: model-based approaches, sensor-based approaches, and hybrid approaches. *Model-based approaches* use a model of the environment to generate a path for the robot to follow. Techniques for model-based path generation include road mapping [4], [5], cell decomposition [5], [6], and potential fields [5], [7]. All of these methods are able to find a path from an initial point to a goal point using a model of the environment. Some of these techniques (e.g. road mapping) can be used to find the shortest path between a starting location and a target location, if such a path exists. However, these methods rely on an accurate model of the environment to generate a safe path (i.e., a path that does not go through or near any obstacles). Since it is usually difficult to obtain an accurate model of a dynamic environment, model-based approaches are primarily used for robot path planning in controlled environments.

Sensor based approaches to mobile robot navigation generate control commands based on sensor data. A promising architecture for sensor-based approaches is the *behavioral architecture*, which consists of multiple behaviors, each one of which reacts to sensor input based on a particular concern of the navigation controller [8], [9]. Examples of typical behaviors include goal-attraction, wall-following, and obstacle-avoidance. The main advantage of sensor-

[1] The problem can be further complicated by other considerations such as deadlines for reaching those locations, safety considerations of paths, reactiveness to emergent situations and uncertainty about the environment.

0018-9472/95$04.00 © 1995 IEEE

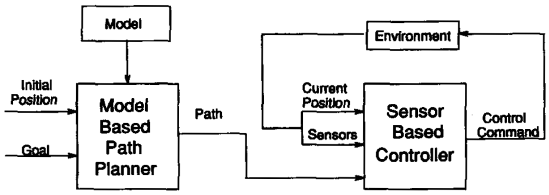

Fig. 2. A hybrid approach to robot motion control.

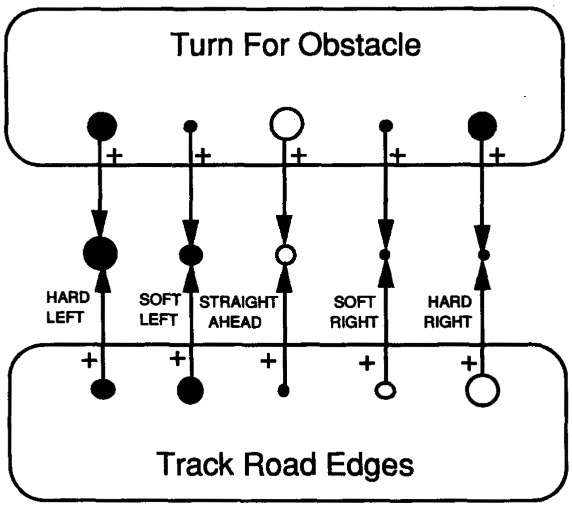

Fig. 3. A Payton-Rosenblatt network for fusing travel direction commands of two behaviors.

based approaches is that the robot can navigate safely in a dynamic environment, because it can easily react to obstacles detected by sensors in real-time. The major limitation of purely sensor-based approaches is that the robot my not reach the goal, even if a path to the goal exists.

Model-based approaches and sensor-based approaches can be combined into *hybrid approaches* to mobile robot navigation [1]. As shown in Fig. 2, a hybrid approach usually uses a model-based planner to generate a path from an incomplete model of the environment. The path is used by a sensor-based controller to navigate the robot such that it follows the path while avoiding obstacles unknown to the model. This approach is thus able to achieve the optimal path of model-based approaches and the reactivity of sensor-based approaches. One important issue of hybrid approaches is how the path information is combined with the sensor data. One way to address this issue is to first generate a set of control commands (e.g. some using the path information and others using the sensor data), which are then fused to an appropriate final control command to be executed by the mobile robot. A suitable scheme for fusing the control commands thus is critical for a hybrid approach to be successful. This need leads to the command fusion technique developed by Payton and Rosenblatt.

B. Payton and Rosenblatt's Command Fusion Network

In the Nov./Dec. 1990 issue of *IEEE Trans. Syst. Man Cyber.*, (vol. 20, no. 6, pp. 1370–1382) David Payton, Ken Rosenblatt and David Keirsey presented an approach for combining control commands of multiple behaviors for an autonomous mobile robot [1]. The approach is motivated by their observation that a fixed-priority-based

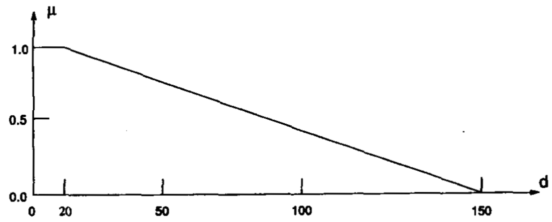

Fig. 4. A fuzzy set representing the concept of near.

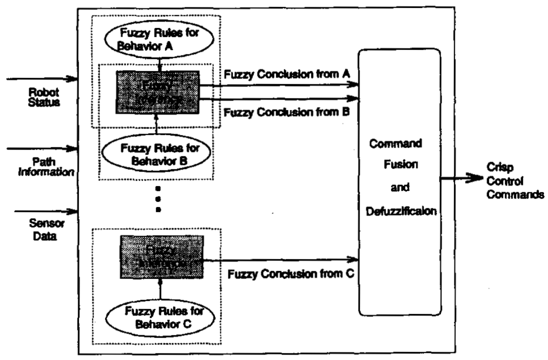

Fig. 5. A behavioral architecture based on fuzzy logic.

command arbitration usually results in loss of information, which makes decision making much more difficult, as illustrated by the example situation in Fig. 1. In this example, the path following behavior suggests to turn left, while the obstacle avoidance behavior suggests to turn either left or right to avoid the obstacle. The priority-based arbitration scheme would only have a 50% chance to make the correct turn (i.e. left), since the information regarding path following is not available once the command arbitrator selects the collision avoidance behavior. Payton and Rosenblatt therefore developed a command fusion method that allows the information conveyed by the sensor data and the path information to be available in choosing the final control command.

In the Payton-Rosenblatt (P-R) method, the output of each behavior is a set of nodes, each of which corresponds to a possible control decision. The confidence and desirability of each control decision is represented by the activation level of the node. To illustrate this, Fig. 3 (which duplicates Fig. 7 in [1]) shows a P-R network for combining two behaviors, *Turn-for-Obstacle* and *Track-Road-Edges*, for the situation given in Fig. 1. The size and color of each node in Fig. 3 represents the behaviors' activation for that command. A node's size represents the magnitude of its activation. Solid black color represents positive activation, while white color represents negative activation. For example, the *Turn-for-Obstacle* behavior in Fig. 3 has a large positive activation level for the "hard left" node, and a large negative activation for the "straight ahead" node.

To obtain the result of combining two behaviors, the activation strengths of the corresponding nodes are combined using a weighted sum scheme, as illustrated by the middle layer in Fig. 3. The weight associated with a behavior reflects the degree of importance of the behavior's suggestion. For instance, the *Turn-for-Obstacle* has a higher weight than the *Track-Road-Edges* behavior because it is more important to avoid hitting obstacles than to follow the track. The final control command is the node with the largest positive activation in the combined behavior based on the winner-take-all selection strategy.

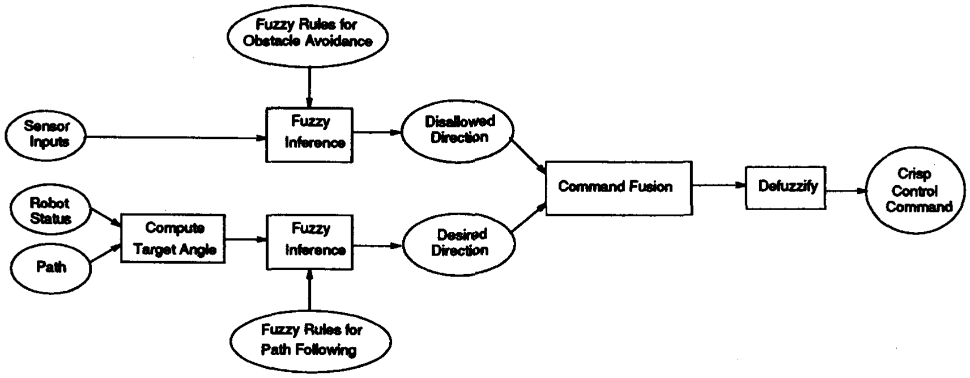

Fig. 6. A fuzzy logic based mobile robot navigation controller.

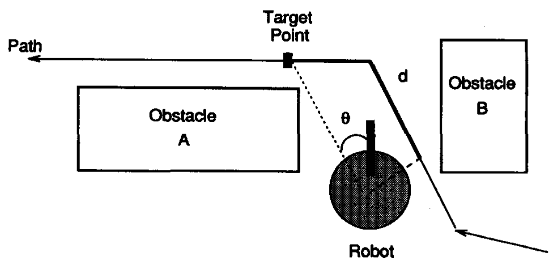

Fig. 7. An example.

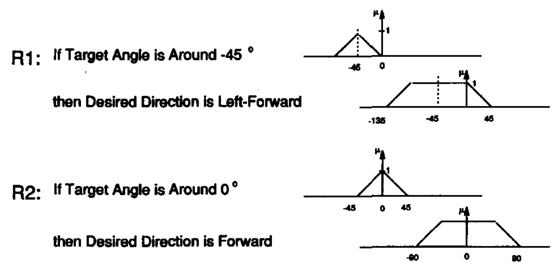

Fig. 8. T_{NO} fuzzy rules for path following.

C. Fuzzy Logic

Motivated by the observation that many concepts in the real world do not have well defined sharp boundaries, Lotfi A. Zadeh developed fuzzy set theory that generalizes classical set theory to allow objects to take partial membership in vague concepts (i.e. fuzzy sets) [10]. The degree an object belongs to a fuzzy set, which is a real number between 0 and 1, is called the *membership value* in the set. The meaning of a fuzzy set, is thus characterized by a *membership function* that maps elements of a universe of discourse to their corresponding membership values.

Fig. 4 shows the membership function of the fuzzy set NEAR in the context of mobile robot navigation control. In this figure, d represents the distance of the closest obstacle detected by a sensor, and μ represents the membership value in the fuzzy set NEAR. As the figure depicts, an object that is 15 units away has a full membership in NEAR, while one that is 46 units away has a membership value of 0.8. Capturing vague concepts such as NEAR using fuzzy sets can improve the robustness of a navigation control system in the presence of sensor noise, because noise in the sensor data can only slightly change the membership degree in NEAR and therefore affects the final control command in a minor way.

Based on fuzzy set theory, fuzzy logic generalizes *modus ponens* in classical logic to allow a conclusion to be drawn from a fuzzy if-then rule when the rule's antecedent is partially satisfied. The antecedent of a fuzzy rule is usually a boolean combination of fuzzy propositions in the form of "x is A" where A is a fuzzy set. The strength of the conclusion is calculated based on the degree to which the antecedent is satisfied. A fuzzy logic controller uses a set of fuzzy if-then rules to capture the relationship (i.e. the control law) between the observed variables and the controlled variables. In each control cycle, all fuzzy rules are fired and combined to obtain a fuzzy conclusion for each control variable. Each fuzzy conclusion is then *defuzzified*, resulting in a final crisp control command. An overview of a fuzzy logic controller and its applications can be found in [11].

Other researchers have also applied fuzzy logic to mobile robot navigation with different focus. Ruspini and his colleagues have developed a fuzzy logic based approach to the explicit representation and execution of complex navigation plan, which was implemented in SRI's award winning robot—Flaky [12]. The use of VLSI fuzzy inferencing chips for implementing sensor-based fuzzy behaviors has also been demonstrated by Pin and Watanabe [13].

D. Relationship Between Payton-Rosenblatt Approach and Fuzzy Logic

Even though Payton and Rosenblatt's (P-R) method was not presented as a fuzzy logic approach, it is similar to fuzzy systems in several ways. First, the activation level of nodes in the P-R network corresponds to membership degrees in fuzzy sets. A high positive activation level, for example, corresponds to a high membership degree in a desirable fuzzy control decision, and a high negative activation level corresponds to a high membership degree in an undesirable fuzzy control decision. Second, the set of nodes representing one behavior, together with their activation levels, corresponds to a discrete fuzzy set. For instance, the nodes of *Turn-For-Obstacle* behavior in Fig. 3 represents a discrete fuzzy set `not forward`. Third, the process of command fusion is analogous to the combination of fuzzy conclusions in fuzzy inference. In particular, the winner-take-all method in the P-R approach for choosing the final control command is similar to the mean-of-maximum defuzzification technique in fuzzy logic controllers.

There are several major differences, however, between the Payton-Rosenblatt approach and a fuzzy system. First, linguistic terms are explicitly used in a fuzzy system, making it easier for designers to understand and modify linguistic fuzzy rules than to comprehend

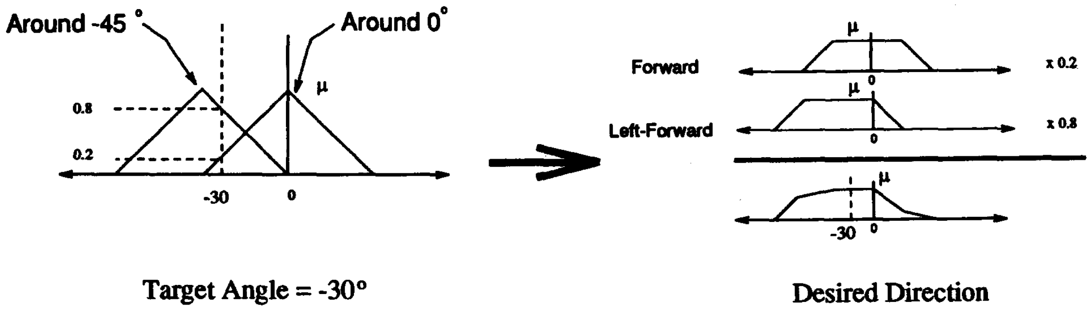

Fig. 9. An example of computing desired direction.

link connections in the P-R command fusion network. Second, the combination of activation levels in the P-R approach can implement combining operations that are difficult to achieve using a single fuzzy operator. However, we will demonstrate in the next section that the same effect can often be achieved using a combination of multiple fuzzy operators. Third, fuzzy logic offers several defuzzification techniques beyond the winner-take-all method. Furthermore, it facilitates the designer to develop new defuzzification techniques suitable for certain applications by extending existing ones. Section III.E describes a new defuzzification technique we developed for applications such as mobile robot navigation control that need to deal with prohibitive information (e.g. traveling directions that should be avoided due to the presence of near obstacles).

III. Our Methodology

A. A Behavioral Architecture Based on Fuzzy Logic

Like other behavioral approaches, the fuzzy logic based architecture of our mobile robot navigation system consists of several behaviors. Each behavior represents a concern in mobile robot control, and relates sensor data, robot status data and path information to control recommendations, as shown in Fig. 5. Unlike other behavioral approaches, however, a behavior in our architecture has two components: (1) a set of fuzzy rules, and (2) a fuzzy inference module. A behavior's fuzzy rules explicitly capture the control strategy of the behavior in the form of linguistic rules. A behavior's fuzzy inference module implements a fuzzy inference scheme appropriate for the behavior. As illustrated in Fig. 5, multiple behaviors could share a common fuzzy inference module. Fuzzy control recommendations generated by all the behaviors are fused and defuzzified to generate a final crisp control command.

Based on the architecture, we have implemented a mobile robot controller, shown in Fig. 6, that consists of two behaviors: one for obstacle avoidance and one for path following. The basic algorithm performed every control cycle by the architecture consists of the following four steps: (1) The path following behavior determines the desired turning directions. (2) The obstacle avoidance behavior determines the disallowed turning directions. (3) The command fusion module combines the desired and disallowed directions. (4) The combined fuzzy command is converted into a crisp command through a defuzzification process. It is worthwhile to note that the desired and disallowed directions are maintained in fuzzy set form to reduce possible loss of information in command fusion.

Fig. 7 shows an example of a situation in which the path following behavior suggests the robot to turn left, but the robot must go straight a little longer to avoid the obstacle on the left (i.e. obstacle A).

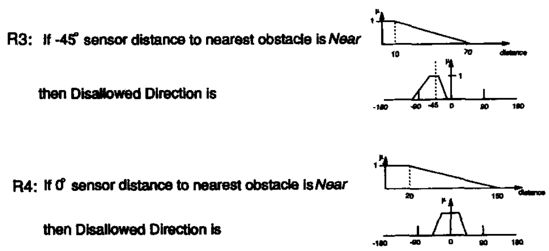

Fig. 10. Fuzzy rules used for obstacle avoidance.

Using this example, we will describe and demonstrate each step of the algorithm in the following subsections.

B. Path Following Behavior

The path following behavior generates a desired turning direction based on the robot's current location, its current heading, and an imperfect path generated from an incomplete and uncertain model of the environment using a simplified road map type method [14]. The path following behavior determines the desired turning direction in three steps. First, it locates a *target point*, which serves as a near term goal, by projecting forward along the path for a predetermined distance d starting from the point on the path closest to the robot. The target point for our example is shown in Fig. 7. Second, the behavior computes the *target angle*, which is the angle between the current heading of the robot and a vector from the robot's current location to the target point. For the example given, with the robot heading toward north, the target angle θ is -30 degrees. Third, the behavior uses a set of fuzzy rules to broaden the specific target angle into a general desired direction, which gives the robot more flexibility in avoiding obstacles while following the path. Two fuzzy rules used by the path following behavior, R1 and R2, are shown in Fig. 8.

The path following behavior's fuzzy inference module combines desired directions recommended by all path following fuzzy rules using weighted sum. This process is illustrated in Fig. 9 for a target angle of -30 degrees using rules R1 and R2 in Fig. 8. The antecedent membership functions (i.e. `Around 0 degrees`, `Around -45 degrees`, etc.) are designed such that the sum of their membership values for an angle is exactly one.[2] We chose weighted sum fuzzy composition instead of other fuzzy reasoning methods (e.g., max-min) for the path following behavior because the behavior performs, in effect, a linear interpolation between rules whose antecedent membership functions are adjacent.

[2] In a study on the stability of fuzzy controllers, it has been shown that this is actually a desirable property for analyzing the stability of fuzzy systems [15].

Sensor Number	Sensor Angle	Distance Returned	Rule Firing Strength
1	-90	200	0.0
2	-45	15	1.0
3	-20	200	0.0
4	0	200	0.0
5	20	50	0.6
6	45	30	0.8
7	90	200	0.0

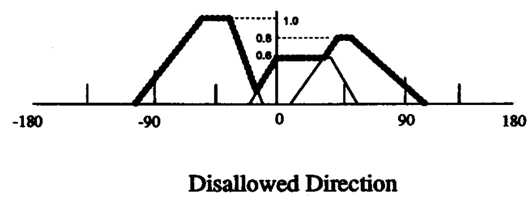

Fig. 11. An example of sensor fusion and the resultant fuzzy set.

C. Obstacle Avoidance Behavior

The obstacle avoidance behavior uses sonar sensor data to generate a fuzzy set that represents the disallowed directions of travel (i.e. directions that lead into or near any obstacles in the short term). The behavior operates by first comparing each sensor input, which measures the distance of the closest obstacle detected by the sensor, to a fuzzy set NEAR associated with the sensor. Based on the result of the comparison, the behavior determines the degree to which the general direction of the sensor is considered disallowed. Examples of fuzzy rules used by the obstacle avoidance behavior are shown in Fig. 10. Note that the membership function of disallowed direction of the rule R3 is not symmetric with respect to -45 degrees. More precisely, the -45 degree sensor has less influence in the forward direction than in the left direction. This is due to the presense of a -20 degree sensor toward the front, while there is no sensor between -45 degrees and -90 degrees. Because the sensors are not uniformly distributed, we have designed the membership function of disallowed turning directions associated with a sensor such that (1) it partially overlaps those of neighboring sensors, and (2) it has a major influence on the sensor's direction.

Once all the fuzzy rules associated with the obstacle avoidance behavior have been fired, their fuzzy conclusions are combined using the max operator. Fig. 11 shows an example of this combination with sensor inputs based on the situation in Fig. 7. The behavior's fuzzy inference module uses the max operator instead of other t-conorm operators (i.e. other union operators in fuzzy set theory) because it is consistent with the intuition that the degree a travel direction is disallowed should be determined by the sensor source that has the strongest opinion about it.

It is worthwhile to point out that the membership functions NEAR for different sensors are different, as illustrated in Fig. 10. This is because an obstacle in the robot's traveling direction poses more of a threat than an obstacle that is on the side. From an obstacle avoidance viewpoint, an obstacle of distance d, detected by the front sensor, is thus considered "closer" to the robot than an obstacle of distance d detected by the left side sensor. In fuzzy logic, the meaning of a linguistic term is always associated with the context in which the term is used. A term could therefore have different meanings in different contexts. This flexibility offered by fuzzy logic makes it possible to capture the desired meanings of the term NEAR for various sensor directions by associating them with the appropriate membership functions.

C. Command Fusion

The third component of the mobile robot navigation controller fuses the fuzzy conclusions about the `desired-direction` and the `disallowed-direction` into a combined fuzzy control command. Since the final turning angle should be both *desired* from the path following viewpoint and *not disallowed* from the obstacle avoidance consideration, the command fusion module uses the min operator in fuzzy logic to form a conjunction of the two behaviors' output as follows:

$$\begin{aligned}\mu_{\text{Turning-Direction}}(x) &= \mu_{\text{Desired AND NOT Disallowed}}(x)\\ &= \min\{\mu_{\text{Desired}}(x),\\ &\qquad \mu_{\text{NOT Disallowed}}(x)\}\\ &= \min\{\mu_{\text{Desired}}(x), 1-\mu_{\text{Disallowed}}(x)\}\end{aligned}$$

For the convenience of our discussion, we will refer to the negated `Disallowed Direction` as the `Allowed Direction` of travel.

Returning to our example situation in Fig. 7, Fig. 12 illustrates the command fusion step under the situation. Even though the target angle in the example is approximately -30 degrees, most of the combined fuzzy command for turning direction is around 0 degrees, which is the correct direction for the robot to take given the presence of obstacle A. The next section will describe how our system converts a combined fuzzy command regarding turning direction into a crisp one.

E. Defuzzification

Defuzzification is the process of converting a fuzzy command into a crisp command, (e.g. turn 4.3 degrees to the right). The two major methods of defuzzification are (1) the Mean of Maximum (MOM) method and (2) the Center of Area (COA) method. The MOM defuzzification method computes the average of those values with the highest membership degree in the fuzzy command. The COA method computes the center of gravity of the entire fuzzy command.

As we have pointed out in Section IID, the Mean of Maximum (MOM) method is similar to the winner-take-all command arbitration scheme in Payton and Rosenblatt's approach. The major drawback of the MOM method for our application is that it does not use all of the information conveyed by the fuzzy command, and thus has difficulty in generating commands that turn the robot smoothly over time.

The COA defuzzification method also has problems when applied to mobile robot control. Let us consider the situation shown in Fig. 13. The combined fuzzy command for the mobile robot's turning direction in this situations has a twin-peak membership function. Applying the COA defuzzification technique to this twin-peak fuzzy set yields a bad command that will bring the robot even closer to the obstacle. In general, the COA method could create problems when it is applied to applications that involve prohibitive information,

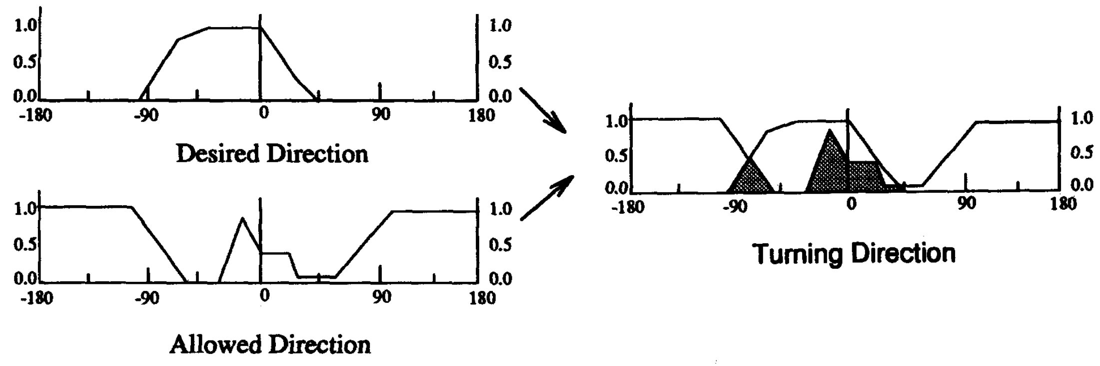

Fig. 12. An example of command fusion.

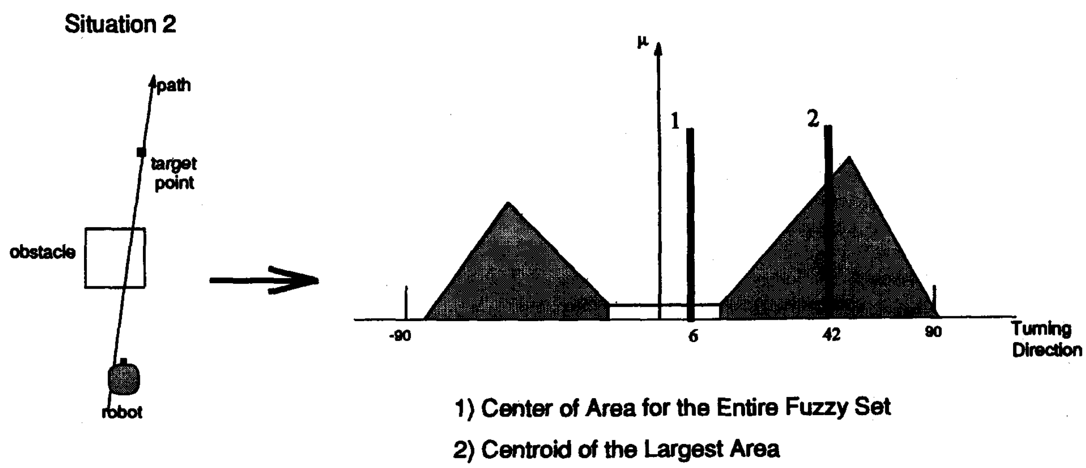

Fig. 13. An example of defuzzification.

for COA does not ensure that the defuzzified decision avoids those regions that are prohibited.

To alleviate this difficulty, we have developed a new defuzzification method, called Centroid of Largest Area (CLA), that partitions a multiple-peak fuzzy command into several disjoint fuzzy subsets, each corresponding to a feasible fuzzy command [16]. The fuzzy subset with the largest area is then selected and defuzzified using the COA method. This new defuzzification technique is illustrated in Fig. 13. The twin-peak fuzzy control command is partitioned into two fuzzy subsets, one corresponding to each feasible fuzzy command. The fuzzy subset on the right is then chosen because its area is larger than the left one. Applying COA defuzzification to the right fuzzy subset results in a turn of 42 degrees to the right. Consequently, the robot will go around the obstacle while maintaining its proximity to the path at the same time.

IV. Simulation Results

We have implemented the mobile robot controller and tested it through extensive simulations. One of these simulations is shown in Fig. 14. These simulations show that the robot can safely navigate in the environment even if the path goes through or near obstacles. The simulations demonstrated in Fig. 14 uses a robot with 7 simulated sonar sensors. We have conducted two experiments to evaluate the system. The first experiment tests the robustness of the system in the presence of sensor noise. The second experiment compares the performance of our defuzzification technique with MOM and COA defuzzification methods.

A. Experiment 1

The first experiment tested the controller's robustness in the presense of various degrees of sensor noise. We tested the controller using 90 imperfect paths. The robot was asked to navigate along each path using seven different sensor noise rates (0, 0.1, 0.2, 0.4, 0.6, 0.8 and 1.0). The simulated sensor noises are characterized by a uniform probability distribution in the interval $[-d \times n, d \times n]$ where d is the actual distance, and n is the sensor noise rate.

We use the following two metrics to measure the performance of the mobile robot controller. (1) *Safety Measure:* The safety measure of the controller is the percentage of simulation runs in which the robot successfully reaches the goal without hitting any obstacles. (2) *Smoothness Measure:* The smoothness measure of the controller is calculated based on the average accumulative turning angles made by the robot.

The result of our empirical evaluation is summarized in Table I, which shows a graceful degradation of the smoothness and safety of the mobile robot controller as the amount of sensor noise increases. Even when the sensor noise rate is as high as 100 percent, the controller is still able to avoid obstacles in most cases. These results suggest that the controller can successfully navigate to the goals using imperfect path information and noisy sensor data.

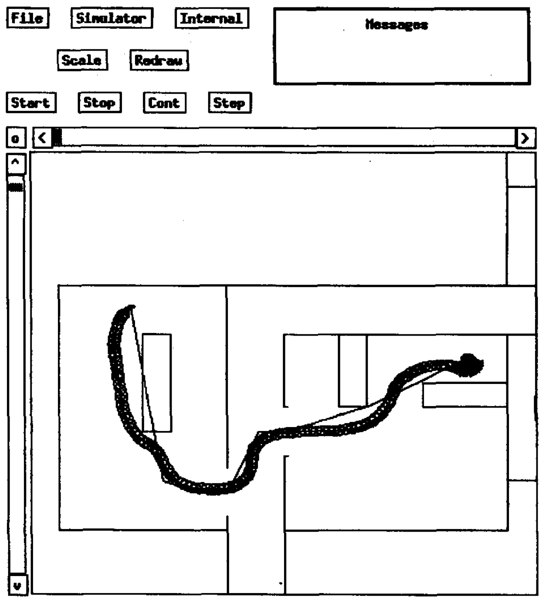

Fig. 14. An example simulation runs.

TABLE I
COMPARISON OF DIFFERENT DEGREES OF ERROR IN SENSOR RETURN

Sensor Noise Rate	Smoothness Measure	Safety Measure
0.00	0.137	1.000
0.10	0.408	1.000
0.20	0.820	0.989
0.40	1.559	0.978
0.60	2.146	0.933
0.80	2.683	0.944
1.00	3.075	0.911

B. Experiment 2

We have also compared our defuzzification technique, Centroid of Largest Area (CLA), with the Mean of Maximum (MOM) and the Center of Area (COA) technique in a similar way using 270 imperfect paths and a sensor noise rate of 0.1. The result of this evaluation is shown in Table II, which indicates that our CLA defuzzification technique achieves a safer and smoother control of the mobile robot than MOM or COA does.

TABLE II
COMPARISON OF DEFUZZIFICATION TECHNIQUES

Defuzzification Technique	Measure of Smoothness	Safety Measure
MOM	2.698	0.415
COA	0.062	0.907
CLA	0.055	1.000

V. BENEFITS

Our fuzzy logic based extension to Payton-Rosenblatt's command fusion scheme offers three important benefits: simplicity, extensibility and understandability.

1. *Simplicity:* The fuzzy logic based navigation controller only needs a small number of rules. In our current implementation, the path following behavior uses seven rules, and the obstacle avoidance behavior uses one rule for each sonar sensor. The simplicity of the system is due to its modular design and the interpolative reasoning capability of a fuzzy system.
2. *Extensibility:* Our approach is easily extensible. For instance, we have successfully modified the controller using seven sensors into one using fifteen sensors in an hour. This is because the desired interaction between sensors is easily achieved by (1) designing the membership functions of the sensors' disallowed directions such that they partially overlap, and (2) choosing the appropriate fuzzy inference scheme (i.e., the max-min inference).
3. *Understandability:* The knowledge of each behavior is easy to comprehend because it is captured in linguistic form by fuzzy rules.

VI. CONCLUSION

We have used fuzzy logic to extend Payton and Rosenblatt's behavioral architecture for mobile robot control. By using fuzzy rules to explicitly capture heuristics implicit in Payton and Rosenblatt's behaviors, we obtain a mobile robot controller that is simple, extensible and understandable, yet can effectively cope with unknown obstacles and imperfect paths in a dynamic environment. We have also developed the CLA defuzzification technique that solves a problem in applying existing defuzzification methods to mobile robot navigation control. The simulation results of our system indicates that it is able to navigate safely and smoothly in the presence of sensor noise. The simulation also shows that our defuzzification technique is superior to existing ones for mobile robot navigation and control.

REFERENCES

[1] D. W. Payton, J. K. Rosenblatt and D. M. Keirsey, "Plan guided reaction," *IEEE Trans. Syst. Man Cyber.*, vol. 20, no. 6, pp. 1370–1382, Nov. 1990.
[2] J. Yen and N. Pfluger, "Designing an adaptive path execution system," in *Proc. IEEE Int. Conf. Syst., Man, and Cyber.*, Carlottesville, VA, pp. 1459–1564, Oct. 1991.
[3] J. Yen and N. Pfluger, "A fuzzy logic based robot navigation system," in *Proc. AAAI Fall Symp. Application of Artificial Intelligence to Real-World Autonomous Mobile Robots*, Cambridge, MA, pp. 195–199, Oct. 1992.
[4] R. C. Luo and T. J. Pan, "An intelligent path planning system for robot navigation in an unkown environment," in *SPIE*, vol. 1195, *Mobile Robots IV*, pp. 316–326, 1989.
[5] J. Latombe, *Robot Motion Planning.* Norwell, MA, Kluwer Academic, 1991.
[6] L. Dorst and K. Travato, "Optimal path planning by cost wave propagation in metric configuration space," in *SPIE*, vol. 1007, *Mobile Robots III*, pp. 186–197, 1988.
[7] M. Okutomi and M. Mori, "Decision of robot movement by means of a potential field," *Advanced Robotics*, vol. 1, no. 2, pp. 131–141, 1986.
[8] T. Anderson and M. Donath, "Synthesis of reflexive behavior for a mobile robot based upon a stimulus-response paradigm," in *Mobile Robots III: Proc. of the SPIE*, vol. 1007, Cambridge MA, Nov. 1988.

[9] R. Brooks, "A robust layered control system for a mobile robot," in *Readings in Uncertain Reasoning*, Morgan Kaufmann, G. Shafer and J. Pearl, Eds., pp. 204–213, 1990.
[10] L. A. Zadeh, "Fuzzy sets," *Inform. Contr.*, vol. 8, no. 3, pp. 338–353, June 1965.
[11] C. C. Lee, "Fuzzy logic in control systems: Fuzzy logic controller—Part I and II," *IEEE Trans. Syst. Man Cyber.*, vol. 20, no. 2, pp. 404–435, Mar. 1990.
[12] E. Ruspini, A. Saffiotti and K. Konolige, "Progress in research on autonomous vehicle motion planning," in *Industrial Applications of Fuzzy Logic and Intelligent Systems*, J. Yen, R. Langari and L. A. Zadeh, Eds. New York: IEEE Press, 1995, chapter 8.
[13] F. G. Pin and H. Watanabe, "Autonomous navigation of a mobile robot using the behaviorist theory and VLSI fuzzy inferencing chips," in *Industrial Applications of Fuzzy Logic and Intelligent Systems*, J. Yen, R. Langari and L. A. Zadeh, Eds. New York: IEEE Press, 1995, chapter 9.
[14] N. Pfluger, "A fuzzy logic approach to command arbitration," Master's thesis, Texas A&M University, 1993, pp. 1–48.
[15] G. R. Langari and M. Tomizuka, "Analysis and synthesis of fuzzy linguistic control systems," in *Intelligent Control 1990, DSC-Vol. 23*, ASME, R. Shoureshi, Ed., pp. 35–42, Nov. 1990.
[16] N. Pfluger, J. Yen and R. Langari, "A defuzzification strategy for a fuzzy logic controller employing prohibitive information in command formulation," *Proc. First IEEE Int. Conf. on Fuzzy Systems*, San Diego, CA, pp. 717–723, Mar. 1992.

The Use of Negative Printing and Attributed-Token State Modelin Character Recognition

D. C. Douglas Hung

Abstract— **This article proposed a new scheme for presenting and recognizing of printed character. The method employs the concept of negative painting i.e. uses surrounding pieces of information as feature elements instead of analyzing the target character directly. Curvature continuity is first parameterize as one of six *symbolic consistencies* (feature primitives). Then, surrounding pieces are represented as attributed-token state sequences (ATS-sequences) of symbolic consistencies. The advantage of the proposed ATS-sequences is that it converts characters into token-sequences without loosing their structural uniqueness wherein the classification can be done by a computational hierarchical classifier.**

I. Introduction

Character recognition is a topic of interest which has attracted numerous research works in recent years. It is a fact that character recognition is one of the major paper-to-computer interfaces that transforms printed information into a symbolic representation. The most crucial issue in character recognition is how accurately and efficiently the system detects similarities as well as dissimilarities between characters. At present, a document can be obtained in a computer-manipulatable digitized form by cameras or optical scanners. From an image processing's point of view, each character can be presented as a collective set of 4-connected pixels, while the background is stated as 8-connected. This implicitly defines a character consisting of two components, such as letter 'i' and 'j', as two disjoint "characters." This undesired disconnection can be resolved easily due to the fact that one of the components is geometrically placed on top of the other one. As a compounded entity, a character of 4-connected pixels is identified and recognized as a whole.

Manuscript received June 20, 1993; revised December 22, 1993; August 5, 1994.

D. C. Hung is with Computer Vision Laboratory, Department of Computer Information Science, New Jersey Institute of Technology, Newark, NJ 07102 USA.

IEEE Log Number 94010387.

A feature-based text recognition algorithm generally can only efficiently handle text with limited fonts and sizes. The same character in fonts, say Roman and Helvetica, will be featured differently because the decorative effect of Roman character creates extraneous pixels on both sides of the vertical and tilting strokes. When a character appears in different sizes, the undersampling effect will distort the silhouette of the digitized character, as well as its description. The features extracted from the character, which are challenged in a multi-font and multi-size environment, will suffer from disturbance. In trying to solve this problem, a considerable amount of earlier research was done. Most of it works on the target character itself—the positive painting [12]. Techniques, such as the line adjacency graph [6], [14] which represents the mutual connectedness among decomposed stroke segments, orientation- and size-invariant descriptor [15], [16] use circular or squared grids to obtain normalized character shape characteristics, attributed graph [7] which parametrizes decomposed stroke segments by a number of geometrical attributes, typographical analysis [8] uses the lower- and upper-baselines of a character's layout to preclassify character, rule-based analysis [10], [13] use grammar-like rules to describe characters, the stream description [9] is the order of a top-to-down, right-to-left raster scanning, the projections of white pixels [11], circular harmonic expansion [5], and a combination of multiple methods [1], [2], [4] have been developed in the last decade.

Most of these approaches need to thin the character images in order to obtain the skeletons and to construct structural descriptions. Both of these two procedures provide useful features, as well as spurious characteristics. This implicitly suggests that the resulting descriptions may not be unique. In addition, the classification of structural or syntactic descriptions is not purely computative. The rest of the methods employ statistic features to which sophisticate discrimination functions are developed to perform the classification with an inevitable certain degree of nonclassification.

On the other hand, the negative painting of a character provides stable structural and quantitative information that can be categorized into simple primitives. Most significantly, the establishing description is less sensitive to the scattering noise, though still affects by regional smudginess that causes structural distortion. In this article, the author focuses his attention on the representation and recognition of multi-font and multi-size machine-printed characters using the concept of negative painting.

Regarding of touched character, Kahan *et al.* [6] separated the merged character by looking for a local maximum in the gradient of the vertical projection profile, Tsujimoto and Asada [17] evaluate the degree of contact using a break cost function, and Fujisawa *et al.* [3] measure the vertical thickness of a character. A new approach which inherits the significant concepts revealed by the above methods and takes advantage of the structural uniqueness of the negative painting is proposed in this paper.

II. Character Puzzle—The Negative Painting

What is a *character puzzle*? It is a character image cut into a number of pieces with irregular shapes. To recognize the original pattern of a puzzle, one needs to put together as many pieces

0018–9472/95$04.00 © 1995 IEEE