

# Implementation of Multi-valued Fuzzy Behavior Control for Robot Navigation in Cluttered Environments

Majura F. Selekwa, Damion D. Dunlap, and Emmanuel G. Collins, Jr.  
*Center for Intelligent Systems, Control and Robotics (CISCOR)*  
*Florida A& M University - Florida State University College of Engineering*  
*Tallahassee, Florida USA*  
*majura@eng.fsu.edu; dunlap@eng.fsu.edu, ecollins@eng.fsu.edu*

**Abstract**—Navigation in environments that are densely cluttered with obstacles is still a challenge for Autonomous Ground Vehicles (AGVs), especially when the configuration of obstacles is not known a priori. Reactive local navigation schemes that tightly couple the robot actions to the sensor information have proved to be effective in these environments, and because of the environmental uncertainties, fuzzy behavior systems have been proposed. The most difficult problem in applying these systems is that of arbitrating or fusing the reactions of the individual behaviors. This paper presents an architectural design of a fuzzy behavior based system for navigation control of robotic vehicles using multivalued reactive fuzzy behaviors. This design allows the robot to thoroughly use the available sensor information when choosing the control action to be taken. Experimental results show that the proposed method can smoothly and effectively guide a robot through cluttered environments such as dense forests.

**Index Terms**—Navigation Algorithms, Reactive Control, Fuzzy Behavior Systems

## I. INTRODUCTION

NAVIGATION through a forest attracts special interest from the military community due to the lack of stealth and concealment found in open environments. This is typical of challenges that autonomous ground vehicles (AGVs) must be able to tackle, i.e., navigation in unstructured complex environments, densely cluttered with obstacles. This navigation problem is a multiobjective control problem that seeks to ensure that the robot not only reaches its goal without hitting obstacles, but also does so at safe speeds that ensure stability. The problem is particularly difficult because some of the navigational objectives may be in opposition to one another.

It is important that algorithms for navigation control in cluttered environments not be too computationally expensive as this would result in a sluggish response. It has been acknowledged that the traditional Plan-Sense-Model-Act approaches are not effective in such environments; instead,

Prepared through collaborative participation in the Robotics Consortium sponsored by the U. S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD 19-01-2-0012. The U. S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

local navigation strategies that tightly couple the sensor information to the control actions must be used for the robot to successfully achieve its mission [1]. The control complexity is overcome by decomposing the navigation control problem into simpler and well-defined subproblems that can be controlled independently and in parallel. These subproblems and their controllers are known as reactive behaviors, and this approach is known as behavior robotics [2]. It has attracted the interests of many roboticists and has even been used in industrial process control applications [3].

Since its introduction in [4] behavior robotics has grown quickly, resulting in the development of reactive fuzzy behavior methods that use fuzzy logic controllers, which can handle uncertainty in the robot information [5], [6], [7], [8], [9], [10], [11], [12]. Fuzzy logic also allows a continuum of control variables of heading angles and speeds to be considered as opposed to the discrete numbers used in crisp behaviors. In addition, it allows the navigation algorithm to be programmed using linguistic terms, which is the way a designer naturally thinks.

The concept of behavior control was initially seen as a special form of decentralized switching control in which each behavior is fully autonomous, and when allowed, can control the robot on its own without regard to other behaviors. Under that paradigm, the behaviors are designed to be univalued, i.e., each behavior triggers a single control command that best meets the control responsibilities specific to that behavior. Hence, the behaviors are essentially competing. This ‘switched parallel’ structure works fairly well when the switching is relatively rare, but the performance of the robot becomes very poor if the behavior switching frequency becomes high, which can lead the robot to be indecisive [13]. If used in cluttered environments where, behavior switching is likely to be high, such an approach is also likely to fail.

Over time, there have been concerted efforts to make behaviors run cooperatively so that the overall robot reaction is generally an amalgamation of the commands from the individual behaviors through some form of command fusion [14], [9], [12], [15], [5]. However, this univalued structure has been found to have serious flaws [16]. In

particular, by treating behaviors as fully autonomous, this structure tends to cause the robot to be indecisive when the behaviors have mutually exclusive interests with nearly equal importance. This observation led to the introduction of multivalued behavior control systems [16], [17], [18]. The first fuzzy implementation of multivalued fuzzy behavior systems was in [17] in which two fuzzy values were used: *allowed* and *disallowed*. The method presented in this paper extends the *allowed-disallowed* structure to include more hedged terms such *may-be-allowed*, *highly-allowed*. Initial experimentation with the *allowed-disallowed* structure produced very good results when obstacles were not closely spaced; the method generated a non-smooth response when the obstacle spacing was decreased, which was handled by adding intermediate hedged terms. This multivalued method was initially proposed in [19] without any experimental validation. Experimental results with the system of [19] illustrated the need to redesign the behaviors proposed therein in order to enable them to behave more realistically and be more computationally efficient. These observations are now included in the system presented in this paper.

This paper presents a detailed basic structure of a reactive fuzzy behavior system for navigation control of a robotic vehicle using multivalued fuzzy behaviors. The practical implementation of this system on a Pioneer 2 robot equipped with a Sick laser range finder is described thoroughly and the resulting performance is also discussed in sufficient detail to show the capabilities of this algorithm.

The paper is organized as follows: Section II presents the detailed structure of the proposed multivalued behavior system for navigation control. The implementation procedure and experimental accommodations are described in Section III. Section IV presents the experimental results that show the performance of the proposed algorithm on a laboratory robot. Concluding remarks are given in Section V.

## II. THE PROPOSED CONTROL SYSTEM BY MULTIVALUED FUZZY BEHAVIORS

This section describes the design of the proposed multivalued fuzzy behavior system. The overall system is broken down into two control actions: heading control and speed control. The heading control is achieved using four behaviors: 1) goal-seeking, 2) obstacle avoidance, 3) left edge tracking, and 4) right edge tracking. The speed control uses two behaviors only: 1) obstacle avoidance and 2) overturning avoidance. Each of these behaviors uses sensory information to determine its course of action. The obstacle avoidance and edge tracking behaviors use range finding sensors to determine distances to the nearest obstacle or path edges; the goal seeking behavior uses compass measurements to determine the direction of the goal; and the overturning avoidance behavior uses a speedometer reading to determine the robot speed. The fuzzy rules for

each behavior  $i$  are expressed in short hand form as

$$\mathbf{IF} \text{ (Stimulus) THEN}(\alpha_{i,1} \text{ and } \alpha_{i,2} \text{ and } \dots \text{ and } \alpha_{i,n}), \quad (1)$$

where  $n$  is the number of possible command alternatives.

### A. The heading control and related behaviors

The control command for the heading control activity is the heading angular change  $\Delta\theta$ . This has to be defuzzified into an odd number of symmetric fuzzy sets to represent possible command alternatives depending on the intended control action. The five fuzzy sets are named: Large Right Turn (LRT), Slight Right Turn (SRT), No Turn (NT), Slight Left Turn (SLT), and Large Left Turn (LLT) as shown by the dashed lines of Fig. 3.

Each behavior  $i$  assigns a relative importance to each command alternative  $j$  by some parameter  $\alpha_{i,j} \in [0, 1]$ ; the larger values correspond to higher importance. This parameter is also expressed by fuzzy sets on the interval  $[0, 1]$ . Any reasonable number of fuzzy sets can be used; four fuzzy sets were found to sufficiently represent the relative importance of the command alternatives with the linguistic symbols Not Acceptable (NA), Acceptable (A), Favored (F), and Highly Favored (HF). The goal seeking structure of the multivalued control system for heading control is illustrated in Fig. 1.

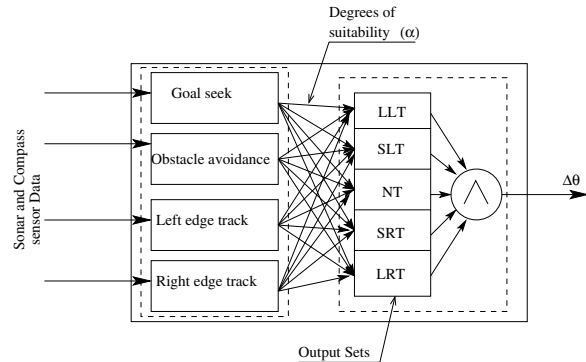


Fig. 1. The Multivalued Behavior Control System for the Heading Control Activity

### B. The speed control activity and related behaviors

The speed control activity determines whether the speed should be increased or decreased; its control command is the speed change  $\Delta v$ . There are only two behaviors in this activity: the obstacle avoidance behavior and the overturning avoidance behavior. As in the case of the heading control activity, each behavior  $i$  assigns a relative importance  $\alpha_{i,j} \in [0, 1]$  to each of the command alternatives  $j$ . If the maximum acceleration of the vehicle is  $a_{\max}$  and the maximum braking deceleration is  $a_{\min}$ , then the instantaneous speed change  $\Delta v$  is in the region  $[a_{\min}, a_{\max}]$ . The speed change  $\Delta v$  command also has to be fuzzified into any reasonable odd number of symmetric fuzzy sets; for many applications it is sufficient to use five fuzzy groups where  $a_{\max}$  and  $a_{\min}$  are assumed to

be normalized to 0.5 and  $-0.5$  respectively. The corresponding linguistic terms are Decrease Significantly (DS), Decrease (D), No Change (NC), Increase (I), and Increase Significantly (IS). The schematic structure for the resulting multivalued control system is shown in Fig. 2.

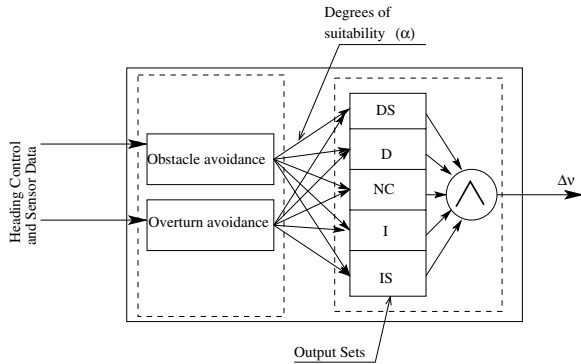


Fig. 2. The Multivalued Behavior Control System for Speed Control Activity

### C. The command block and defuzzification

For all of the above behaviors, the fuzzy outputs  $\alpha_{i,j}$  for ( $i = 1, 2, \dots, 4$ ) and ( $j = 1, 2, \dots, 5$ ) are behavior preferences to the available command alternatives represented by fuzzy sets as in Figs. 1 and 2. The undefuzzified values of  $\alpha_{i,j}$  from the individual behaviors  $i$  for each control command alternative  $j$  are fused by the command block by using the intersection operation

$$\alpha_j = \bigcap_i \alpha_{i,j}. \quad (2)$$

Each resulting fuzzy  $\alpha_j$  for each command alternative  $j$  is then defuzzified using the standard center of area method into a real number  $\bar{\alpha}_j \in [0, 1] \cap \mathbb{R}$ , which is the measure of the importance of each command alternative  $j$ . The defuzzified values  $\bar{\alpha}_j$  are used to determine the appropriate control command  $\Delta\theta$  or  $\Delta v$ . There are two possible approaches for determining  $\Delta\theta$ . The first approach takes  $\bar{\alpha}_j$  as inputs to the fuzzy rules of the form

$$\text{IF}(\bar{\alpha}_1 \text{ and } \bar{\alpha}_2 \text{ and } \bar{\alpha}_3 \text{ and } \bar{\alpha}_4 \text{ and } \bar{\alpha}_5)\text{THEN}(\Delta\theta), \quad (3)$$

where the input fuzzy sets are same as those of  $\alpha_{i,j}$ 's. This approach is computationally intensive, especially since it requires many rules to be defined. For the case when  $\bar{\alpha}_j$  is defined using three fuzzy sets and there are five command alternatives, this approach would require  $3^5 = 243$  rules, which can be prohibitive.

Alternatively, these defuzzified values of  $\bar{\alpha}_j$  are treated as the highest levels of  $\alpha$ -cuts, which are used in resolution of the fuzzy sets corresponding to the control commands as illustrated in Fig. 3. This latter approach is computationally more attractive and tends to produce better results than the former approach especially since there are certain  $\alpha_j$  combinations for which rules (3) will tend to be biased in one direction and may cause the

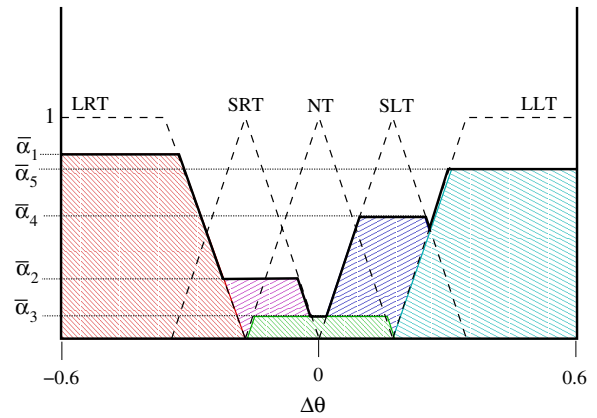


Fig. 3. A Typical Compound Fuzzy for  $\Delta\theta$

robot to exhibit erratic turns when the environment is such that left and right turns have near equal preference. For both methods, the command values  $\Delta\theta$  and  $\Delta v$  are obtained by defuzzifying the resulting compound fuzzy set of  $\Delta\theta$  and  $\Delta v$  respectively. Note that it is possible for this compound fuzzy set to have two or more distinct regions, as those shown in Fig. 3; therefore, any defuzzification approach based on averaging would yield a result that falls between these regions, which is undesirable. To prevent this from happening, the center of maximum area which was proposed in [17] was adopted in the results reported here. The method divides the fuzzy output terms into separate groups whenever an intermediate term falls below a certain threshold. This prevents the output from falling in a region that represents an unsuitable control action. The group with the largest degree of suitability is then defuzzified using the standard center of area, which makes the best compromise between the control alternatives within that group. In the event that the compound fuzzy set has two or more distinct but equal maximum areas, the area that is closest to the origin (i.e., closest to NT or NC) will be defuzzified to yield the desired control command.

### III. ALGORITHM IMPLEMENTATION

After satisfactory simulation performance [19], the proposed navigation control system has been implemented and tested in a laboratory environment on a Pioneer 2 robot equipped with a SICK laser range finder (Fig. 4) [20]. This



Fig. 4. Pioneer 2 Robot With SICK Laser Range Finder

robot, which is manufactured by ActivMedia Robotics, is

a differentially driven platform configured with two drive wheels and one swivel caster for balance. Each wheel is driven independently by a motor with 19.5:1 gear ratio which enables the robot to drive at a maximum speed of 1.2 m/s and climb a 25% grade [21]. The proposed system was prepared using *fuzzyTECH* software, which generated C++ code that was implemented on the Pioneer 2.

#### A. Pioneer 2 sensors

The Pioneer 2 is equipped with several types of sensors. The navigation control system requires range sensors and localization sensors. This subsection will give a brief description of the range and localization sensors used in this implementation.

1) *Range sensors*: Two types of range sensors were available: a bank of sonar sensors as well as a laser range finder. Because of its better accuracy and resolution the laser range finder was used in this control system. The laser range finder used is a SICK LMS 200; it has a resolution of 10mm, a typical measurement accuracy of  $\pm 15$ mm, a  $180^\circ$  scanning angle, and 10m typical measured distance range [22]. Measurements can be made for scan angles as small as  $0.25^\circ$  that can be composed into rectangular and cone shaped regions [22].

2) *Localization Sensors*: Typical localization sensors include Global Positioning Systems (GPS), Inertial Navigation Systems (INS) and an electronic compass. The Pioneer 2 robot used in these experiments had none of these localization sensors. Instead, localization information was achieved computationally by using wheel encoders. Each motor on the mobile robotic platform is equipped with a 500 tick encoder [21]. These measure the change in orientation of the motors in increments of  $1/500$  of a rotation. This information along with the drive gear ratio provide the change in orientation of each wheel, which is differentiated to provide wheel velocities. The obtained wheel velocities are used in the calculation of the position of the vehicle relative to its initial position, and hence localization is achieved.

#### B. Implementation of Control System on Pioneer 2

This subsection gives a concise description of how the control system was implemented on the Pioneer 2 robot. This implementation is based on the work of [20], which can be consulted for further details. In the laboratory environment the robot was limited to a low speed of 50 cm/s, hence this implementation did not cover the speed control.

1) *Range Sensor Configuration*: Since the laser range finder can scan over a  $180^\circ$  angle, it provides many measurements that cannot be effectively processed for control purposes. To make the range finder readings manageable, the sensor inputs were grouped into a total of nine regions. Initially all nine of these regions were represented by equal sized cones of  $20^\circ$  each. This orientation had to satisfy two conditions:

- 1) At a minimum safe distance  $d_s$  between the robot and the side obstacles the width of the six side regions

must represent a width  $w_t$  that is traversable by the robot.

- 2) The combined three front sensor regions must represent a width traversable by the robot.

The latter condition was simplified further, which led to the front regions being rectangular with an overall width that is slightly wider than the actual robot in order to provide for some safety margin. Similarly, the conic side sensor regions were widened to overlap; these sensor regions are illustrated in Fig. 5. The necessary, included angle of the cone  $\theta_s$  for these side sensor regions was calculated as

$$\theta_s = 2 \arcsin\left(\frac{w_t}{2d_s}\right), \quad (4)$$

which yielded  $\theta_s = 35^\circ$  for a  $w_t/d_s$  ratio of 0.6.

It was observed later that the larger these regions are, the more likely it is that the robot will not be able to discern a traversable path even when one actually exists. The sensor inputs for larger regions tend to indicate that an excessively wide space is needed in order for the proposed direction to be traversable. The opposite is also true, i.e., smaller regions make the algorithm more likely to attempt a direction that is not in fact traversable. Therefore the sensor region size becomes a major factor in the process

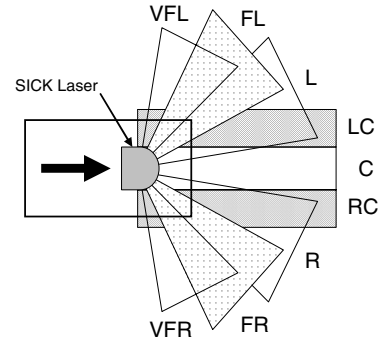


Fig. 5. Configuration of Range Sensor Regions

2) *Behavior influence*: Since the proposed control system requires the behaviors to cooperate, it is important to limit the influence of each behavior such that the behaviors are bound to actions that are directly related to their goals only. These limitations are characterized according to the control objective of each behavior and are enforced through determination of the input and output terms available to each behavior rule block.

The realm of influence of each behavior must be established based upon an understanding of how the behavior results will be combined to determine the suitability of each proposed control action. Since the behavior outputs are combined using an intersection operation, it only takes one behavior to reduce the suitability of a proposed control action. Based on this information, a behavior should not take any action that can reduce the suitability of a proposed control action that it does not have information about. For example, the right edge tracking behavior is only given

sensor information about the obstacles on the right side, which does include the front to some extent based on the sensor configuration, as illustrated in Fig. 5. Therefore the right edge tracking rule block should not return a suitability other than 1.0 for any proposed control action concerning turning to the left. Similar conditions prevail for the left edge tracking and obstacle avoidance behaviors.

It is important that each behavior has relevant and necessary information for it to make proper decisions. This information should be sufficient to satisfy its control objective, avoiding any unnecessary or redundant information. For example, since the purpose of the goal seeking algorithm is to reach the goal then this behavior in and of itself should totally disregard irrelevant information such as sensor inputs regarding obstacles. The only essential information is a relative angle to the goal position.

### C. Rule Base Determination

The sensor configuration shown in Fig. 5 was divided into three circular zones. These zones represent the range distances close (C), medium (M) and far (F), and were used as bases for fuzzification of the sensor measurements. The intended control action for each set of possible input terms was determined in terms of behavior preferences ( $\alpha_{i,j}$ ). This was done systematically for each behavior while making a conscious attempt to maintain smoothness, by avoiding rule sets that allow the output to change drastically from among behaviors and input terms. For example, a rule should not say that one control action is HF (highly favorable) while the action directly next to it is NA (not acceptable). Since the goal seeking behavior directs the robot to a specific predefined target using the shortest route, it is possible that the direct path to the goal has obstacles. Because of the possibility that the direct path may have obstacles as dictated by the obstacle avoidance behavior, the goal seeking behavior should not assign the linguistic value ‘Not Acceptable’ (NA) to any of the possible command alternatives. Its function is then relegated to determining the degree to which the command alternatives are acceptable instead of deciding whether or not those command alternatives are in fact acceptable. This even allows the robot to turn away from the obstacle if there is no traversable path as will be shown later in the experimental results.

The obstacle avoidance, right edge tracking, and left edge tracking behaviors each had 27 rules while the goal seeking behavior had 5 rules. The fuzzy control command was determined using the  $\alpha$ -cuts method and defuzzified using the center of maximum area as discussed in Subsection II-C.

### D. Membership Function Shaping

After having fully determined the input and output variables and their terms as well as establishing the behavior rule blocks it was then necessary to shape their fuzzy membership functions. These membership functions must be shaped according to the purpose of the term that it represents.

1) *Input Variable Terms:* In general, trapezoidal membership functions were chosen for all the obstacle avoidance input terms while the orientation input also uses triangular shapes. The actual shapes of these membership functions depend on the objective of the individual behaviors. For example, since it is acceptable for the robot to be close to the side path edges but not close to a front obstacle, range measurements had different membership functions as shown in Figs. 6 through 7. The shapes of the membership

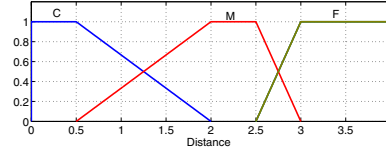


Fig. 6. Membership Functions for FL, L, LC, C, RC, R, and FR.

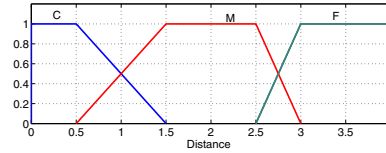


Fig. 7. Membership Functions for VFL and VFR.

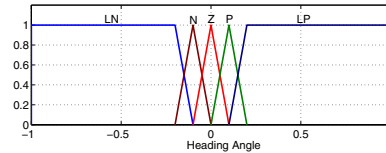


Fig. 8. Membership Functions for  $\Phi$ .

functions were tuned primarily by experimental trial-and-error. Increasing the amount of overlap between input membership functions helped to allow the activation of multiple rules more often, which resulted in more smooth control. This overlap was increased between the C and M terms for all sensor ranges except VFR and VRL, which have a wider M region, as illustrated in Figs. 6 and 7. The heading direction input terms  $\Phi$  are closely grouped as shown in Fig. 8 to provide more fine control for smaller changes in angle. The universe of discourse for the membership functions in Fig. 8 are normalized to  $\pm 1$  by a factor of  $\pi$ .

2) *Output Variable Terms:* The output terms ( $\alpha_{i,j}$ ) used triangular membership functions although *fuzzyTECH* converts them into singletons during code compilation. Because of the limitation of behavior influence as previously discussed, membership functions for the outputs  $\alpha_{i,j}$  of each behavior  $i$  were different. Figs. 9 through 12 show the different membership functions for the  $\alpha_{i,j}$ 's, where  $j = 1, 2, \dots, 5$  with  $\alpha_{i,1}$  expressing favorability to LLT and  $\alpha_{i,5}$  expressing favorability to LRT in sequential order. The outputs represented by membership functions in Fig. 9 have more fuzzy groups than others. The membership

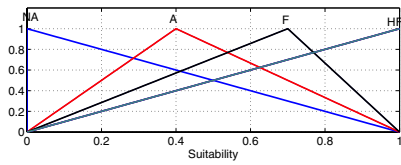


Fig. 9. Membership Functions for  $\alpha_{1,3}$ ,  $\alpha_{2,4}$ ,  $\alpha_{2,5}$ ,  $\alpha_{3,1}$ , and  $\alpha_{3,2}$ .

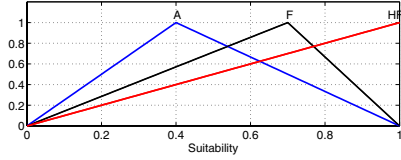


Fig. 10. Membership Functions for  $\alpha_{1,2}$ ,  $\alpha_{1,4}$ , and  $\alpha_{4,j}$   $j = 1, 2 \dots 5$ .

functions for the edge tracking in Fig. 11 are shifted more to the right; this gives the navigation a slight preference to not turn, which reduces the control effort. Due to the realm of influence limitations discussed in Subsection III-B, the output terms illustrated in Fig. 12 only have a HF term.

The universe of discourse for  $\Delta\theta$  membership functions was appropriately normalized to the maximum turn angle. In the experimental results presented here the maximum turn angle was  $180^\circ$ .

#### E. Command Action Execution in Continuous Time

In implementing the control action, which is the change in angle  $\Delta\theta$ , there are two possible approaches: either turn at a constant predetermined turning rate  $\omega$  or complete the turning in a fixed predetermined turning interval  $\Delta t$  such that the turning rate  $\dot{\theta}$  (or  $\omega$ ) becomes

$$\omega = \dot{\theta} = \frac{\Delta\theta}{\Delta t}. \quad (5)$$

The later approach has serious limitations when large turning angles  $\Delta\theta$  are involved since faster rotational velocities induce more slip that drastically decreases the accuracy of robot localization. On the other hand, when small turning angles are involved the robot turns at very low rates that seem to waste time. It was decided to use fixed turning

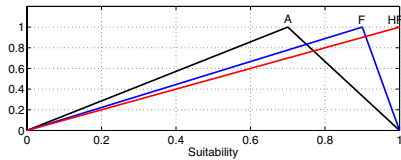


Fig. 11. Membership Functions for  $\alpha_{2,3}$  and  $\alpha_{3,3}$ .

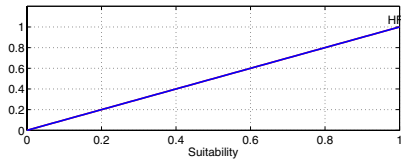


Fig. 12. Membership Functions for  $\alpha_{1,1}$ ,  $\alpha_{1,5}$ ,  $\alpha_{2,1}$ ,  $\alpha_{2,2}$ ,  $\alpha_{3,4}$ , and  $\alpha_{3,5}$ .



Fig. 13. Robot Navigating Through a Dense Obstacle Field

rate and a procedure was included to hold the computation until one control output has been completed before the calculation of another command. The only concern with this constraint is that the resulting execution speed of the navigation is limited by the robot's maximum turning rate  $\omega_{max}$ . This physical limitation is attributed to the selected robotic platform and does not significantly slow the algorithm.

#### IV. OBSERVED PERFORMANCE

This section presents a sample of the experimental results that show the performance of the proposed control system. It starts by describing the configuration of obstacles that the robot was to avoid.

##### A. Obstacle description and configuration

A dense forest in which trees become obstacles to robot motion was chosen as the experimental environment. Such obstacles are very difficult to navigate through because they are relatively small with irregular spacing. Trees were simulated by 2' long by 2'', 3'', and 4'' diameter PVC pipe sections. These pipe diameters scale appropriately to the vehicle size and accurately depict the trunks of trees.

The configuration of these obstacles must be chosen carefully. Firstly it is important for each obstacle configuration to have at least one traversable path. There may be more than one traversable path; however, an obstacle configuration with only one traversable path is the most difficult because the robot must be able to identify and navigate that one path. The existence of multiple paths can serve to illustrate the decision making of the algorithm by forcing the robot to choose a more straight path. The path is considered traversable if it is wide enough for the robot to negotiate and make appropriate turns. Fig. 13 shows the robot navigating in an obstacle field.

##### B. Results

For each experimental scenario the robot is set at a particular start point and the goal is defined in either Cartesian or polar coordinates with respect to the start position. Depiction of experimental results is best represented in video format; however, since that medium is not available for print, the obstacle configurations were mapped and the localization data  $(X, Y, \theta)$  of the robot were recorded and plotted in the same axes as shown in Figs. 14-21. They

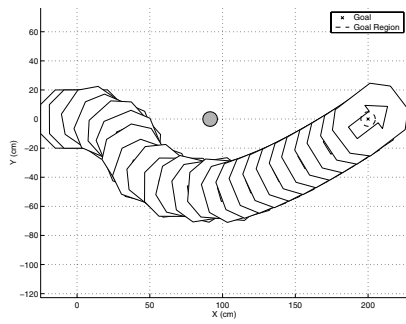


Fig. 14. Experimental Results for Scenario 1

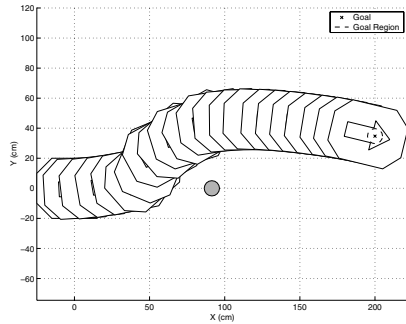


Fig. 15. Experimental Results for Scenario 2

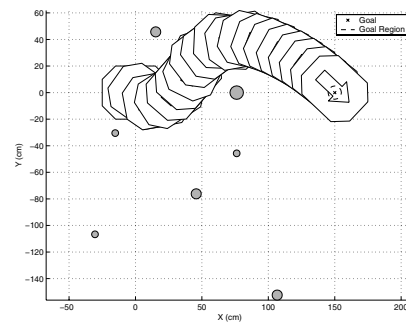


Fig. 17. Experimental Results for Scenario 4

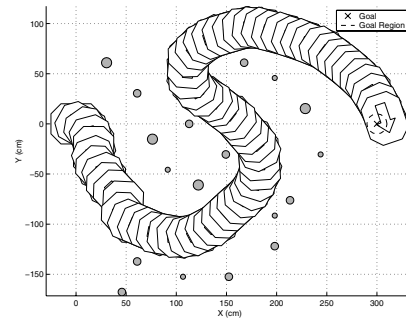


Fig. 18. Experimental Results for Scenario 5

are not simulation results but a depiction of the physical position of the Pioneer 2 robot relative to the obstacles in the  $x - y$  plane of experimentation space. The scenarios presented here progress from simple test cases to more complex obstacle configurations.

Scenario 1 (Fig. 14) has only one obstacle directly in front of the robot along the line that joins the robot start position and the goal. The robot chose to avoid the obstacle by going to the right even though its almost identically feasible to turn in either direction. Scenario 2 (Fig. 15) represents the same obstacle configuration except with the goal position moved slightly to the left which results in the robot avoiding the obstacle in that direction. This shows that the control system does not prefer a particular direction except when the alternatives have equal feasibility; it also demonstrates that the system takes into consideration the goal seeking behavior while avoiding obstacles.

Scenarios 3 and 4 (Fig. 16 and Fig. 17) represent almost

identical obstacle configuration except that Scenario 4 has one less obstacle. The missing obstacle in this scenario creates a shortest path to the goal, which is more favored by this system. As such the path taken in Scenario 4 is shorter than that in Scenario 3.

Scenarios 5 and 6 (Fig. 18 and Fig. 19) represent more complex situations that illustrate the ability to navigate very small gaps and even turn away from the goal when necessary to avoid obstacles. Although the goal seeking behavior steers the robot to the goal, it also yields to the demands of other behaviors especially the obstacle avoidance behavior when there is no traversable direct path to the goal. The robot can even move away from the goal after getting very close to it as long as it finds no traversable path that leads more directly to the goal.

Scenarios 7 and 8 (Fig. 20 and Fig. 21) represent a more sparse obstacle configuration that is more typical of an actual forest. There are various traversable paths

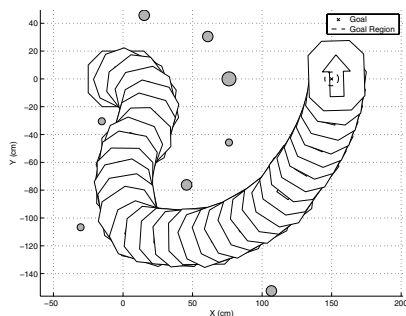


Fig. 16. Experimental Results for Scenario 3

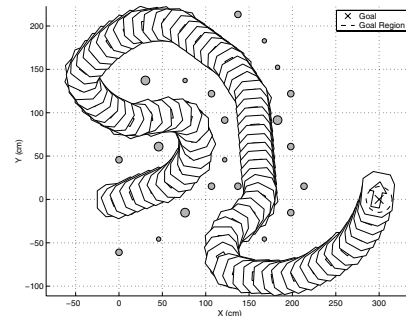


Fig. 19. Experimental Results for Scenario 6

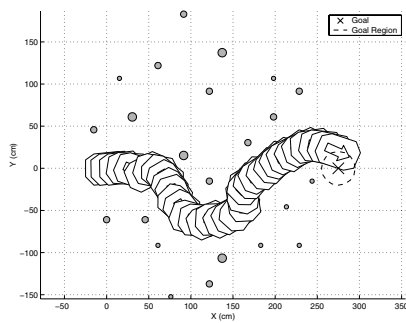


Fig. 20. Experimental Results for Scenario 7

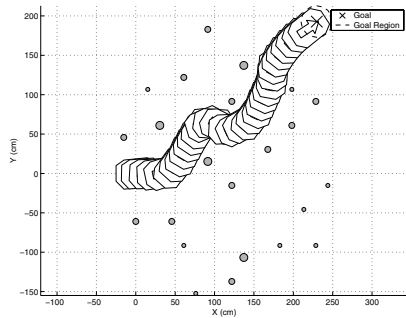


Fig. 21. Experimental Results for Scenario 8

but the robot navigates the most direct path to the goal. These scenarios have the same obstacle configuration with different goal locations. In Scenario 8 ( Fig. 21) turning to the right is more favorable than turning to the left from an obstacle avoidance perspective but the contribution of the goal seeking behavior makes it more appropriate to take a left turn.

## V. CONCLUSION

This paper has discussed the major improvements done to the previously proposed multivalued behavior system for robot navigation control system, which provides a reliable method of behavior fusion. The improved algorithm was implemented on Pioneer 2 robot in cluttered environments showing very good performance in avoiding obstacles and reaching a goal in a variety of obstacle configurations. The success may be attributed to its structure, which allows all behaviors to express their interest in the available commands and fuse them by a fuzzy intersection operation. This method ensures that no behavior is ignored during the process, which results in smooth navigation even in cluttered environments where the subsumption methods would require switching behaviors at a higher rate, which can lead to indecisiveness and failure. Additionally, by not ignoring any behavior, the robot is always able to focus towards the goal without being distracted by the presence of multiple obstacles.

## DISCLAIMER

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing

the official policies, either expressed or implied, of the Army Research Laboratory or the U. S. Government.

## REFERENCES

- [1] A. Sgorbissa and R. C. Arkin, "Local navigation strategies for a team of robots," *Robotica*, vol. 21, pp. 461–473, 2003.
- [2] R. C. Arkin, *Behavior-Based Robotics*. Cambridge Massachusetts: MIT Press, 1998.
- [3] S. P. Linder, "Behavior-based robust fuzzy control," in *Proceedings of the 1998 IEEE ISIC/CIRA/ISAS Joint Conference, Gaithersburg, Maryland, USA*, September, 1998, pp. 233–238.
- [4] R. Brooks, "A Robust Layered Control System for a Mobile Robot," *IEEE Journal of Robotics and Automation*, pp. 14–23, April, 1986.
- [5] E. Aguirre and A. Gonzalez, "Fuzzy Behaviors For Mobile Robot Navigation. Design, Coordination and Fusion," *International Journal of Approximate Reasoning*, vol. 25, pp. 225–289, 2000.
- [6] S. C. Goodridge and R. C. Luo, "Fuzzy behavior fusion for reactive control of an autonomous mobile robot: MARGE," in *Proceedings of the IEEE International Conference on Robotics and Automation, San Diego, CA*, July, 1997, pp. 1622–1627.
- [7] S. Ishikawa, "A method of indoor mobile robot navigation by using fuzzy control," in *Proceedings of the IEEE/RSJ International Workshop on Intelligent Robots and Systems IROS '91, Osaka, Japan*, Nov., 1991, pp. 1013–1018.
- [8] W. Li, "Fuzzy Logic-based 'Perception-Action' Behavior Control of a mobile robot in Uncertain Environments," in *Proceedings of the 3rd IEEE Conference on Fuzzy Systems, Vol. 3*, June, 1994, pp. 1626–1631.
- [9] F. G. Pin and Y. Watanabe, "Navigation of Mobile Robots Using Fuzzy Behaviorist Approach and Custom Designed Fuzzy Inference Boards," *Robotica*, vol. 12, pp. 491–503, 1994.
- [10] A. Saffiotti, "Fuzzy logic in autonomous robotics: behavior coordination," in *Proceedings of the IEEE International Conference on Fuzzy Systems, Barcelona, Spain*, July, 1997, pp. 573–578.
- [11] A. Saffiotti, K. Konolige, and E. H. Ruspini, "A multivalued logic approach to integrating planning and control," *Artificial Intelligence*, vol. 76, pp. 481–526, 1995.
- [12] E. W. Tunstel, Jr., "Mobile Robot Autonomy Via Hierarchical Fuzzy Behavior Control," in *Proceedings of the 6th International Symposium on Robotics and Manufacturing (ISRAM'96), Montpellier, France*, May, 1996, pp. 837–842.
- [13] R. Siegwart and I. R. Nourbakhsh, *Introduction to Autonomous Mobile Robots*. The MIT Press, 2004.
- [14] A. Saffiotti, E. H. Ruspini, and K. Konolige, "Blending Reactivity and Goal Directedness in a Fuzzy Controller," in *Proceedings of the 2nd IEEE Conference on Fuzzy Systems, Vol. 1*, June, 1993, pp. 134–139.
- [15] N. E. Hodge and M. B. Trabia, "Steering Fuzzy Logic Controller for an Autonomous Vehicle," in *Proceedings of the IEEE Intl. Conference on Robotics and Automation, Detroit Michigan*, vol. 3, May 1999, pp. 2482–2488.
- [16] D. W. Payton, J. Rosenblatt, and D. M. Keirse, "Plan Guided Reaction," *IEEE Transactions on Systems Man and Cybernetics*, vol. 20, no. 6, pp. 1370–1382, 1990.
- [17] J. Yen and N. Pfluger, "Fuzzy Logic Extension to Payton and Rosenblatt's Command Fusion Method for Mobile Robot Navigation," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 25, no. 6, pp. 971–978, 1995.
- [18] J. Rosenblatt, "DAMN: Distributed Architecture for Mobile Navigation," Ph.D. Thesis, Carnegie Mellon University Robotics Institute, Pittsburgh, PA, 1997.
- [19] M. F. Seleka and E. G. Collins, "A Centralized Fuzzy Behavior Control for Robot Navigation," in *Proc. IEEE International Symposium on Intelligent Control, Houston Texas*, October 2003, pp. 602–607.
- [20] D. D. Dunlap, "Implementation Of Multi-Valued Fuzzy Behavior Control For Navigation In Cluttered Environments," Master's thesis, Florida A & M University, Tallahassee, Florida, 2004.
- [21] *Pioneer 2 / PeopleBot Operations Manual*, 9th ed., ActivMedia Robotics, 44 Concord St., Peterborough NH, 03458, October, 2001.
- [22] A. I. Division, *Technical Description: LMS 200 Series Laser Measurement Systems*, 8th ed., SICK, Inc., 6900 West 110th St., Bloomington, MN 55438, June, 2003.