Autonomous Robots manuscript No.
(will be inserted by the editor)

Robot Navigation In Very Cluttered Environments By Preference-Based Fuzzy Behaviors

Majura F. Selekwa[1], Damion D. Dunlap[2], Emmanuel G. Collins, Jr.[2] ⋆

[1] Department of Mechanical Engineering and Applied Mechanics, North Dakota State University, Fargo, ND, 58105

[2] Center for Intelligent Systems, Controls and Robotics, Department of Mechanical Engineering, Florida A&M University - Florida State University, Tallahassee, FL,32310

Received: date / Revised version: date

Abstract One of the key challenges in application of Autonomous Ground Vehicles (AGVs) is navigation in environments that are densely cluttered with obstacles. The control task becomes more complex when the configuration of obstacles is not known a priori. The most popular control methods for such systems are based on reactive local navigation schemes that tightly couple the robot actions to the sensor information. Because of the environmental uncertainties, fuzzy behavior systems have been proposed. The most difficult problem in applying fuzzy reactive behavior based navigation control systems is that of arbitrating or fusing the reactions of the individual behaviors, which is addressed here by the use of preference logic. This paper presents the design of a preference-based fuzzy behavior system for navigation control of robotic vehicles using the multivalued logic framework. This design allows the robot to thoroughly use the available sensor information when choosing the control action to be taken. Simulation and experimental results show that the proposed method can smoothly and effectively guide a robot through cluttered environments such as dense forests.

⋆ Manuscript 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.

1 Introduction

Safe maneuvering of Autonomous Ground Vehicles (AGVs) in unstructured complex environments, densely cluttered with obstacles is still a major challenge in goal directed robotic vehicle applications. Navigation through a forest, which attracts special interest from the military community due to the lack of stealth and concealment found in open environments, is typical of such challenges. 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, local navigation strategies that tightly couple the sensor information to the control actions must be used for the robot to successfully achieve its mission [Sgorbissa & Arkin, 2003]. The control complexity is overcome by decomposing the navigation control problem into more simple 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 [Arkin, 1998]. It has attracted the interests of many roboticists and has even been used in industrial process control applications [Linder, 1998].

Since its introduction in [Brooks, 1986] 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 [Aguirre & Gonzalez, 2000, Goodridge & Luo, 1997, Ishikawa, 1991, Li, 1994, Pin & Watanabe, 1994, Saffiotti, 1997, Tunstel, Jr., 1996]. Fuzzy logic also allows a continuum of control variables such as 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. Probably, the greatest strength of behavior-based fuzzy approaches is that they operate on and reason with uncertain perception-based information, which makes them suitable even for difficult environments such as unknown terrains as shown in [Seraji & Howard, 2002].

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 what we will call the standard behavior paradigm, 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 the performance of the robot becomes very poor if the behavior switching frequency becomes high, which can lead the robot to be indecisive [Siegwart & Nourbakhsh, 2004]. 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 [Aguirre & Gonzalez, 2000, Pin & Watanabe, 1994, Tunstel, Jr., 1996, Saffiotti et al., 1995, Hodge & Trabia, 1999]. However, most of these efforts have been based on developing fuzzy versions of the standard behavior structure in which each behavior chooses one action out of the possible actions, in this case a fuzzy action. These structures were found to have performance problems [Payton et al., 1990] especially since they treat behaviors as fully autonomous, which 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 what we will collectively refer to as preference-based behavior systems. The first of these were voting architectures like the DAMN architecture [Payton et al., 1990, Rosenblatt, 1997], where behaviors expressed their preference for or against a possible action through a voting system. The method of [Payton et al., 1990, Rosenblatt, 1997] was implemented using fuzzy logic in [Yen & Pfluger, 1995]; two fuzzy preferences were used: *allowed* and *disallowed*. Independently Saffiotti et al. implemented a similar system using desirability functions and preference logic [Saffiotti et al., 1993]. This early work was later generalized into the multivalued logic approach [Saffiotti et al., 1995], establishing the framework that this paper applies to the dense obstacle navigation problem. Our initial work was proposed in [Selekwa & Collins, 2003] without any experimental validation. Experimental results with the system of [Selekwa & Collins, 2003] demonstrated the need to redesign the behaviors proposed therein to make them behave more realistically and have greater computational efficiency. These observations are now included in the system presented in this paper.

The primary contribution of this paper is the presentation of a system for navigation control of a robotic vehicle in very cluttered environments using preference-based 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 2 discusses the general structures of standard and preference-based systems and points out their differences. Section 3 presents the detailed structure of the proposed system. Simulation results that compare the performance of the proposed system with a navigation control system that uses standard behaviors are presented in Section 4. The experimental results that show the performance of the proposed algorithm on a laboratory robot are described in Section 5. Concluding remarks are given in Section 6.

2 Standard and preference-based fuzzy behavior control structures

This section describes the general structures of standard and preference-based fuzzy behavior control systems. These structures can apply to a variety of control applications.

2.1 Standard structure for fuzzy behavior systems

A standard structure consists of a finite set of distributed independent fuzzy behaviors and a system of arbitration or command fusion. Each behavior is a fuzzy logic control system that responds to its stimuli by issuing a single command that is transmitted for command fusion. Fig. 1 shows the basic structure of a these systems, where each of the behaviors use the environmental information to determine the control command that satisfies its particular objective, e.g., obstacle avoidance, path following, goal seeking, etc. The behaviors determine the appropriate control commands through rules of the form

IF (Stimulus has a particular value) **THEN** $\cdots$

(A Control command is assigned a particular value). (1)

The behaviors are referred to as standard because each behavior responds by triggering only one command signal. The command signal for one behavior does not take into consideration the command of another behavior. If the system has several behaviors with conflicting responses that do not intersect, these behaviors compete for the control of the robot, i.e., each behavior seeks to satisfy its own interests. The behavior conflicts in these systems has been cited as one of the leading causes for robot navigation failures [Payton et al., 1990,Rosenblatt, 1997, Selekwa & Collins, 2003].

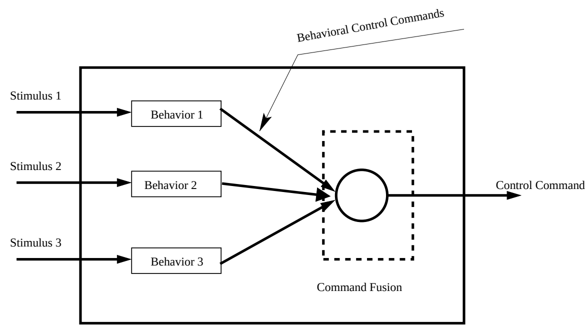

Fig. 1 The General Structure of a Standard Behavior Control System

2.2 Preference-based structure for reactive fuzzy behavior systems

A preference-based structure also consists of a finite set of parallel running behaviors, and a centralized control command unit. A set of possible control commands is kept by this command unit and is also known by each behavior. The behaviors respond to their respective stimuli by expressing their preference levels α_i to each of the available command alternatives i through rules of the form

IF (Stimulus has particular values) $\cdots$

THEN (Command 1 has preference α_1) $\cdots$

AND(Command 2 has preference α_2) $\cdots$

$\cdots$, $\cdots$ **AND**(Command n has preference α_n). (2)

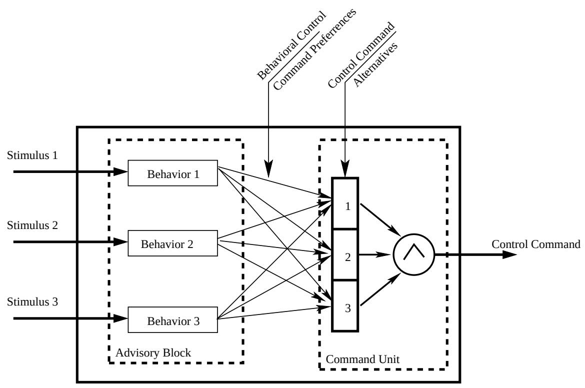

Fig. 2 The General Structure of a Preference-based Behavior Control System.

The central control command unit augments the preferences that each command receives from the behaviors and selects the command that gets the highest score. The behaviors are treated as advisors to the central command unit; they together form an advisory block. Since each behavior has to express its relative preference to each of the available command alternatives, it responds by firing multiple signals, one corresponding to each of the available command alternatives. Fig. 2 shows the basic structure of a preference-based fuzzy behavior system. In a preference-based fuzzy behavior system each behavior is a fuzzy system and the preference levels are expressed as fuzzy sets.

The preference-based fuzzy behavior system presented in the next section was inspired by the results of [Saffiotti et al., 1993,Saffiotti et al., 1995] and [Payton et al., 1990, Rosenblatt, 1997], which were some of first approaches to use preference-based structures for robot navigation. The work of [Saffiotti et al., 1995] served to firmly establish a rigorous generalized framework for the development of similar methods to their navigational algorithm first demonstrated in [Saffiotti et al., 1993]. The Distributed Architecture for Mobile Navigation (DAMN) of [Rosenblatt, 1997] and its predecessor, the fine grained behavior system of [Payton et al., 1990], used crisp behaviors that express their preference through voting (positive,negative, or neutral) for each command. The preference levels or votes from all behaviors to each command were weighted and summed up, and the command that had the highest sum was selected as the system command.

A natural drawback of all crisp behavior systems is the lack of continuity in the commands, where each command is allowed to only have discrete values and intermediate values are not executed. Fuzzy behaviors inherently eliminate this problem, which helps provide smooth motion.

2.3 Qualitative differences

Although both the standard and the preference-based fuzzy behavior systems are behavior based systems that use self contained behaviors, these systems have distinct differences. First, the design of the behaviors in the standard system allows them to directly influence the final control command, i.e., the response of the individual behavior is itself a control command that can drive the robot. On the other hand, the preference-based behaviors do not have this capability. The control commands are issued by the command block. Because of this structure, it is seen that the decision process passes through two steps for a standard system and three steps for a preference-based system.

Second, the behavior fusion process under a standard system tends to favor the high priority behaviors such that the information communicated by the low priority behaviors tend to be ignored. Hence, there is an inherent loss of information under these systems. On the other hand, since the fusion process for the preference-based system picks the command that best fits the interests of all behaviors, this system tends to preserve most of the information required for command generation [Saffiotti et al., 1993].

The performances of representative standard and preference-based behavior control systems in a hypothetical office apartment have been compared [Combey, 2003, Selekwa et al., 2004]. As expected, the preference-based systems consistently outperformed the standard system in terms of success rate, path length, and bending energy.

3 The proposed control system by preference-based fuzzy behaviors

This section describes the design of the proposed preference-based fuzzy behavior system. The overall system is based on the point kinematics of the robot. Since in the x-y plane

$$\begin{aligned} \tfrac{dx}{dt} &= v\cos(\theta), \\ \tfrac{dy}{dt} &= v\sin(\theta), \end{aligned} \tag{3}$$

where v is the robot speed, θ is heading direction relative to the x-axis, the x-y position of the robot at instant k can be computed as

$$\begin{aligned} x(k+1) &= x(k) + \tau_s v_k \cos(\theta_k), \\ y(k+1) &= y(k) + \tau_s v_k \sin(\theta_k), \end{aligned} \tag{4}$$

where τ_s is the sample interval. Hence, the navigation problem can be broken down into two control actions: heading control for determining θ_k, and speed control for determining v_k. The heading control is achieved using four behaviors: 1) goal-seeking, 2) front obstacle avoidance, 3) left obstacle avoidance, and 4) right obstacle avoidance. 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 behaviors use range finding sensors to determine distances to the nearest obstacle; 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 are shown in short hand form, where for example, rule (2) would be expressed as

$$\textbf{IF} \text{ (Stimulus) } \textbf{THEN}(\alpha_1 \text{ and } \alpha_2 \text{ and } \cdots \text{ and } \alpha_n). \quad (5)$$

3.1 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 that depend on the intended control action. The five fuzzy sets shown in Fig. 3 are illustrations of such a set. The fuzzy labels

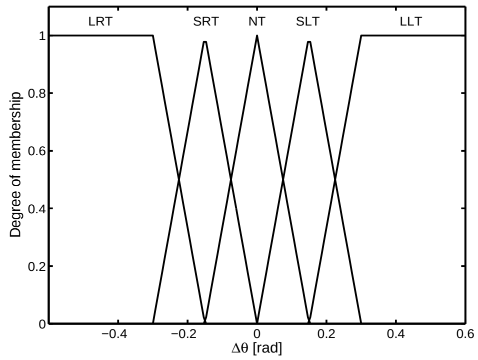

Fig. 3 Fuzzy Sets for the Heading Control Command

in this figure are: Large Right Turn (LRT), Slight Right Turn (SRT), No Turn (NT), Slight Left Turn (SLT), and Large Left Turn (LLT).

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; three fuzzy sets as illustrated in Fig. 4 were found to sufficiently represent the preference of the command alternatives with the linguistic symbols: Not Acceptable (NA), Acceptable (A), Favored (F), and Highly Favored (HF). The structure of the preference-based control system for heading control is illustrated in Fig. 5. A brief description of each behavior is given below.

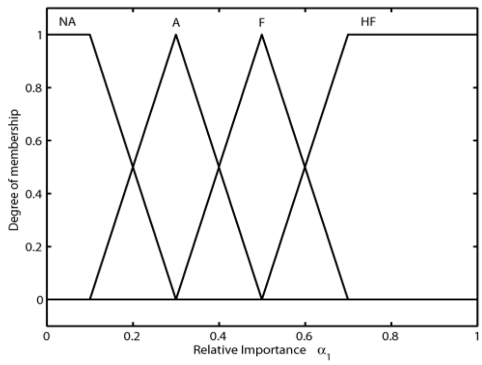

Fig. 4 Fuzzy Sets for the Preference Level

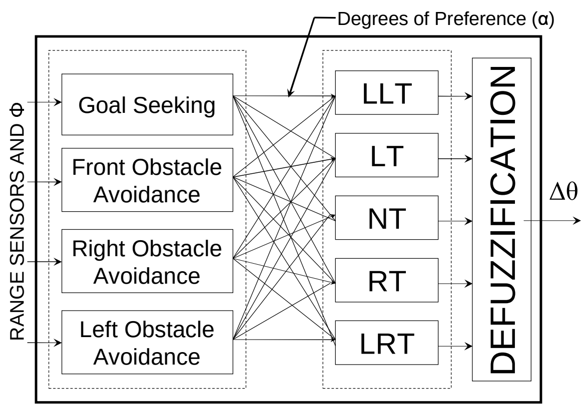

Fig. 5 The Preference-based Behavior Control System for the Heading Control Activity

3.1.1 The obstacle avoidance behaviors The front, left, and right obstacle avoidance behaviors use range sensor measurements to determine the preference for the possible movements. Its design is such that this behavior becomes effective when an obstacle is observed in some neighborhood of the robot. The fuzzy rules for this behavior are expressed as

$$\textbf{IF}(\text{Range Sensor Readings })\textbf{THEN} \quad (\alpha_{i,1} \text{ and } \alpha_{i,2} \text{ and } \alpha_{i,3} \text{ and } \alpha_{i,4} \text{ and } \alpha_{i,5}), \quad (6)$$

where i corresponds to the behavior with $i = 1$, $i = 2$, and $i = 3$ representing the front, right, and left obstacle avoidance behaviors. The range sensor readings express the distances to obstacles in their respective directions; these distances are fuzzified into a reasonable number of fuzzy sets such as the three fuzzy sets shown in Fig. 6. All range sensor readings in this paper will be assumed to be fuzzified as in this figure.

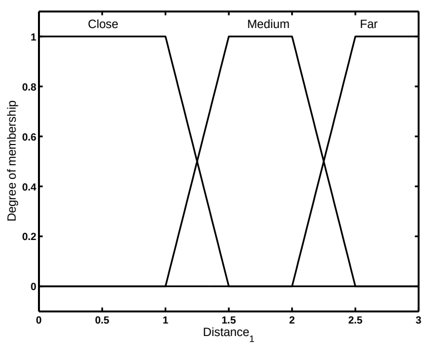

Fig. 6 Fuzzy Groups for the Measured Distance

3.1.2 The goal seeking behavior The goal seeking behavior directs the robot to a specific predefined target.

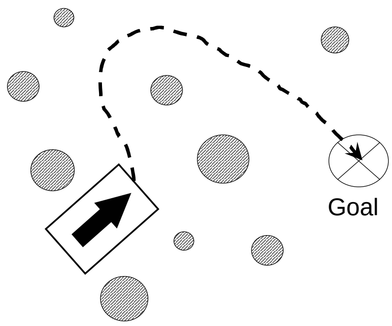

Fig. 7 Turning Away From Goal Required to Avoid Obstacles

The behavior is implemented by fuzzy rules of the form:

$$\textbf{IF}(\Phi)\textbf{THEN}(\alpha_{4,1} \text{ and } \alpha_{4,2} \text{ and } \alpha_{4,3} \text{ and } \alpha_{4,4} \text{ and } \alpha_{4,5}). \quad (7)$$

Since this behavior is specifically concerned only with the goal, it encourages the system to follow a straight path towards the goal. Hence, the presence of any obstacles in the path towards the goal is not considered by this behavior, but rather by the obstacle avoidance behaviors. As a result of behavior fusion, the presence of obstacles may necessitate the robot to follow a path that is not directly towards the goal such as that shown in Fig. 7, which naturally would be undesirable for the goal seeking behavior alone. Because of this possibility, 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.

3.1.3 The command block For all of the above behaviors, the fuzzy outputs $\alpha_{i,j}$ for $(i = 1, 2, \cdots, 4)$ and $(j = 1, 2, \cdots, 5)$ are behavior preferences to the available command alternatives represented by fuzzy sets as in Fig. 5. 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

$$\alpha_j = \bigcap_i \alpha_{i,j}. \tag{8}$$

This type of fusion is known as *conjunctive combination* [Saffiotti et al., 1995]. Each resulting fuzzy α_j for each command alternative j is then defuzzified using the standard center of area method into a real number $\overline{\alpha}_j \in [0, 1] \bigcap \mathbb{R}$, which is the measure of the importance of each command alternative j. The defuzzified values $\overline{\alpha}_j$ are used to determine the appropriate control command $\Delta\theta$. 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

$$\textbf{IF}(\overline{\alpha}_1 \text{ and } \overline{\alpha}_2 \text{ and } \overline{\alpha}_3 \text{ and } \overline{\alpha}_4 \text{ and } \overline{\alpha}_5)\textbf{THEN}(\Delta\theta), \tag{9}$$

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

Alternatively, the defuzzified values of $\overline{\alpha}_j$ can be used as the highest level α-cuts used in the resolution of the fuzzy sets corresponding to the control commands, forming a compound fuzzy set such as the one illustrated in Fig. 8. This latter approach is computationally more attractive and tends to produce better results than the former approach, especially since there are certain α_j combinations for which rules (9) will always be biased in one direction and may cause the robot to exhibit erratic turns. For both methods, the command value $\Delta\theta$ is obtained by defuzzifying the resulting compound fuzzy set of $\Delta\theta$. However, since it is possible to form this compound fuzzy set to have two or more distinct regions such as those shown in Fig. 8, a special defuzzification that avoids the control command from being in the undesired region has to be used. The center of maximum area approach of [Yen & Pfluger, 1995] was found to offer very good results in this case. This method divides the fuzzy output terms into separate groups whenever an intermediate term falls below a certain threshold. This prevents

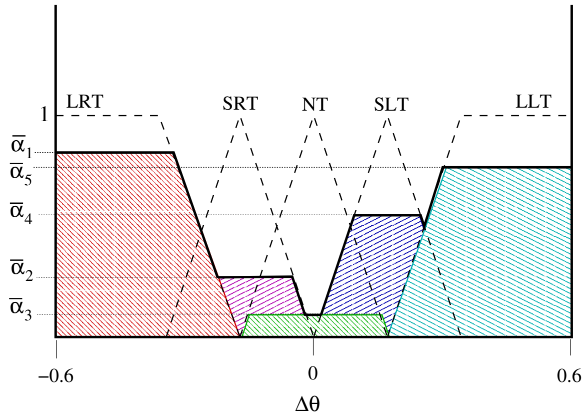

Fig. 8 A Typical Compound Fuzzy for $\Delta\theta$

the output from falling in a region that represents an unsuitable control action. The group with the largest degree of preference 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 equal distinct maximum areas, the area that is closest to the origin (i.e., closest to NT) will be defuzzified to yield the desired control command $\Delta\theta$.

3.2 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 Δv. There are only two behaviors in this activity: an obstacle avoidance behavior and an overturning avoidance behavior. As in the case of the heading control activity, each behavior i assigns a relative importance $\mu_{i,j} \in [0, 1]$ to each of the command alternatives j. The fuzzy sets for $\mu_{i,j}$ are the same as those for $\alpha_{i,j}$ illustrated in Fig. 4. If the maximum acceleration of the vehicle is $a_{\max}$ and the maximum braking decceleration is $a_{\min}$, then the instantaneous speed change Δv is in the region $[a_{\min} \cdot T, a_{\max} \cdot T]$ where T is the controller time step. The speed change Δv 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 such as those shown in Fig. 9, where $a_{\max} \cdot T$ and $a_{\min} \cdot T$ are assumed to be normalized to 0.5 and -0.5 respectively. The corresponding linguistic labels in this figure are Decrease Significantly (DS), Decrease (D), No Change (NC), Increase (I), and Increase Significantly (IS). The schematic structure for

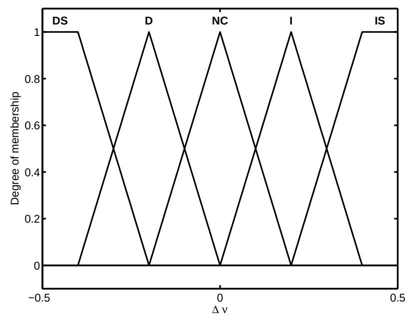

Fig. 9 Fuzzy Groups for Δv

the resulting preference-based control system is shown in Fig. 10. A more detailed description for each behavior is given below.

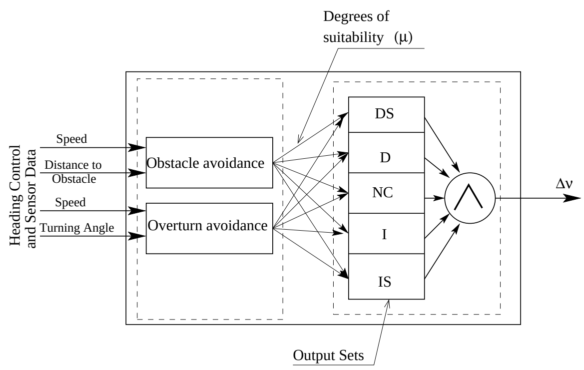

Fig. 10 The Preference-based Behavior Control System for Speed Control Activity

3.2.1 The obstacle avoidance behavior The obstacle avoidance behavior for the speed control activity uses the minimum distance to the nearest obstacle D_m and the current speed v to determine the amount by which the speed should be changed Δv. If the acceptable braking distance that a vehicle can stop from the current speed

v is d_s, then the distance D_m can be fuzzified over a range $[0, d_s]$; any number of fuzzy groups can be used, although it suffices to use the same fuzzy sets as those in Fig. 6, where d_s is normalized to 3.

The fuzzification for the speed v is also done by using any number of fuzzy groups in the range $[0, v_{\max}]$, where $v_{\max}$ is the maximum allowable speed of the robot. Three fuzzy sets such as those shown in Fig. 11 may be sufficient. The fuzzy implementation of the behavior becomes

$$\textbf{IF}(D_m \text{ and } v) \textbf{ THEN}$$
$$(\mu_{1,1} \text{ and } \mu_{1,2} \text{ and } \mu_{1,3} \text{ and } \mu_{1,4} \text{ and } \mu_{1,5}). \quad (10)$$

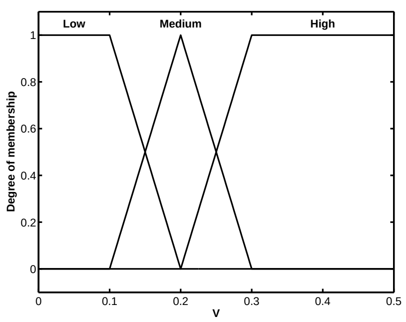

Fig. 11 Fuzzy Groups for v

3.2.2 The overturning avoidance behavior The purpose of the overturning avoidance behavior is to prevent the robot from making turns at higher speeds, which might cause the vehicle to roll over. This behavior does not need any sensor measurements. All it needs are the current speed v and the magnitude of the desired change $|\Delta\theta|$ in heading angle from the heading control activity. The fuzzy implementation is

$$\textbf{IF}(v \text{ and } |\Delta\theta|) \textbf{ THEN}$$
$$(\mu_{2,1} \text{ and } \mu_{2,2} \text{ and } \mu_{2,3} \text{ and } \mu_{2,4} \text{ and } \mu_{2,5}). \quad (11)$$

3.2.3 The command block The command block for the speed control activity is similar to that for the heading control activity described in the previous section. It fuses the behavior preferences $\mu_{i,j}$ from all behaviors i for each command alternative j by the intersection operation

$$\mu_j = \bigcap_i \mu_{i,j}. \quad (12)$$

The resulting fuzzy μ_j for each command alternative j is defuzzified into a real number $\overline{\mu}_j \in [0, 1] \bigcap \mathbb{R}$, and the control command Δv is determined either by using rules of the form

$$\textbf{IF}(\overline{\mu}_1 \text{ and } \overline{\mu}_2 \text{ and } \overline{\mu}_3 \text{ and } \overline{\mu}_4 \text{ and } \overline{\mu}_5) \textbf{ THEN}(\Delta v), \quad (13)$$

or through defuzzification of the compound fuzzy set of Δv formed by using $\overline{\mu}_j$ values as α-cuts for each respective fuzzy set of the alternative control commands of Fig. 11 in a similar fashion as for the heading control discussed earlier.

4 Simulation Results

The original control developments for navigating in cluttered environments actually focused on the development of a standard fuzzy behavior system. Simulation results in a cluttered cave demonstrated the inadequacy of the standard approach. As an illustration, Fig. 12 shows the failure of the standard algorithm to reach the goal. This failure is because the goal seeking behavior is sometimes effectively ignored due to the higher priority of the obstacle avoidance behavior. Although the standard algorithm could be tuned to work for a particular scenario, it would fail for an alternative scenario.

Fortunately, the proposed preference-based fuzzy behavior system was found to be effective in the cluttered cave environments. As illustrated by Fig. 13, the preference-based system reached the goal for a wide variety of cluttered cave scenarios. Further, as illustrated by Fig. 14, the preference-based system reached the goal in a wide variety of scenarios representing dense forests. The standard algorithm failed for all of the forest scenarios simulated.

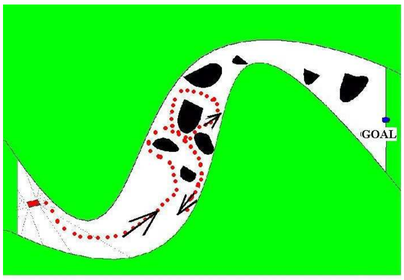

Fig. 12 Simulated Performance of a Standard Control System Showing Failure in a Cave with Cluttered Obstacles

5 Experimental Results

After satisfactory simulation performance, the proposed navigation control system has been implemented and tested in a laboratory environment on a Pioneer 2 robot

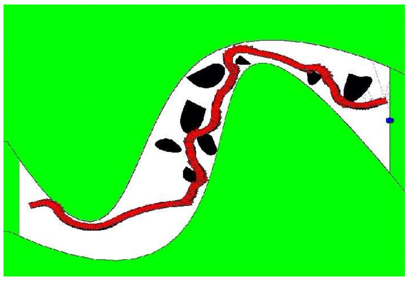

Fig. 13 Simulated Performance of a Preference-based Control System in a Cave with Cluttered Obstacles

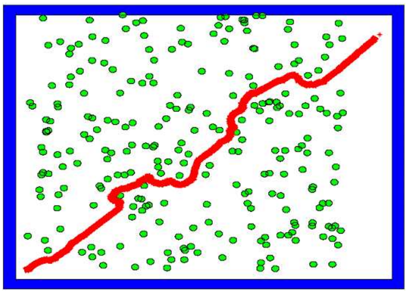

Fig. 14 Simulated Performance of a Preference-based Systems in a Forest

equipped with a SICK laser range finder (Fig. 15) [Dunlap, 2004]. This robot, which is manufactured by Activ-

Fig. 15 Pioneer 2 Robot With SICK Laser Range Finder

Media 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 [Robotics, 2001]. The proposed system was prepared using *fuzzyTECH* software, which generated C++ codes that were implemented on the Pioneer 2.

5.1 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.

5.1.1 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$15mm, a 180° scanning angle, and 10m typical measured distance range [Division, 2003]. Measurements can be made for scan angles as small as 0.25° that can be composed into rectangular and cone shaped regions [Division, 2003].

5.1.2 Localization Sensors Typical localization sensors include a Global Positioning System (GPS), an Inertial Navigation System (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 the wheel encoders. Each motor on the mobile robotic platform is equipped with a 500 tick encoder [Robotics, 2001]. 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.

5.2 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 [Dunlap, 2004], which can be consulted for further details. In the laboratory environment the robot was limited to low speeds, hence this implementation did not cover the speed control.

5.2.1 Range Sensor Configuration Since the laser range finder can scan over a 180° 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° 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 lead 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 the adjacent regions; these sensor regions are illustrated in Fig. 16. The necessary included angle of the cone θ_s for these side sensor regions was calculated as

$$\theta_s = 2\arcsin(\frac{w_t}{2d_s}), \tag{14}$$

which yielded $\theta_s = 35°$ 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 of tuning the algorithm.

5.2.2 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 sometimes described in terms of the *context of applicability* [Saffiotti et al., 1995] of behaviors. They were 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, which is less formal but more efficient.

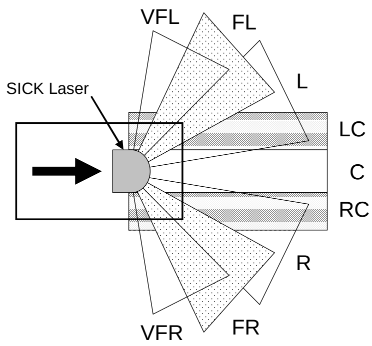

Fig. 16 Configuration of Range Sensor Regions

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 preference of each proposed control action. Since the behavior outputs are combined using an intersection operation, it only takes one behavior to reduce the preference of a proposed control action. Based on this information, a behavior should not take any action that can reduce the preference of a proposed control action that it does not have information about. For example, the right obstacle avoidance 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. 16. Therefore the right obstacle avoidance rule block should not return a preference other than 1.0 for any proposed control action concerned with turning to the left. Similar conditions prevail for the left and front 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, 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.

5.3 Rule Base Determination

The sensor configuration shown in Fig. 16 was divided into three circular zones as shown in Fig. 17. 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 conscience 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). The

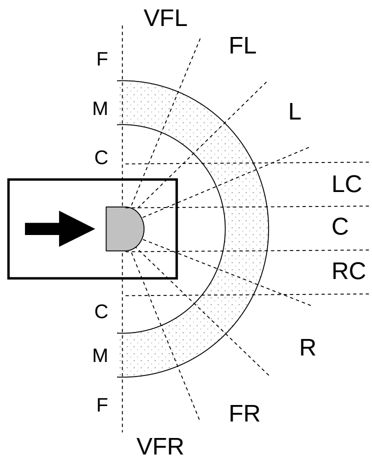

Fig. 17 Zoning the Range Measurements

front, right, and left obstacle avoidance behaviors each had 27 rules while the goal seeking behavior had 5 rules. The fuzzy control command was determined using the α-cuts method and deffuzzified using the center of maximum area as discussed in Subsection 3.1.3.

5.4 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.

5.4.1 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 obstacles but not close to a front obstacle, range measurements had different membership functions as shown in Figs. 18-19. The shapes

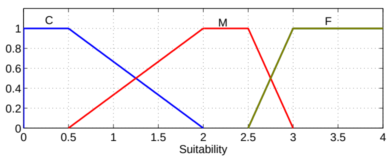

Fig. 18 Membership Functions for FL, L, LC , C, RC, R, and FR.

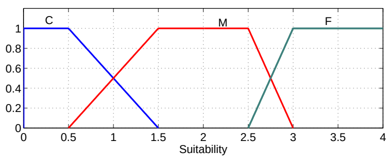

Fig. 19 Membership Functions for VFL and VFR.

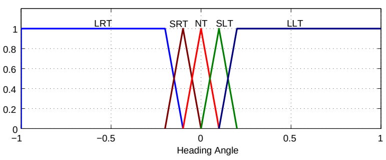

Fig. 20 Membership Functions for Φ.

of the membership 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. 18 and 19. The universe of discourse for the membership functions in Fig. 19 are normalized to ± 1 by a factor of π.The heading direction input terms Φ are closely grouped as shown in Fig. 20 to provide more fine control for smaller changes in angle.

5.4.2 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 discussed before, membership functions for the outputs $\alpha_{i,j}$ of each behavior i were different. Figs. 21 through 24 show the different membership functions for the $\alpha_{i,j}$'s, where $j = 1, 2 \cdots, 5$ with $\alpha_{i,1}$ expressing favorability to LLT and $\alpha_{i,5}$ expressing preference to LRT in sequential order. The outputs represented by the member-

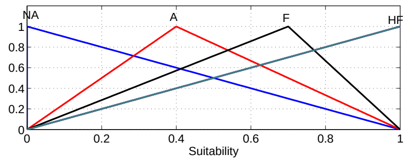

Fig. 21 Membership Functions for $\alpha_{1,3}$, $\alpha_{2,4}$, $\alpha_{2,5}$, $\alpha_{3,1}$, and $\alpha_{3,2}$.

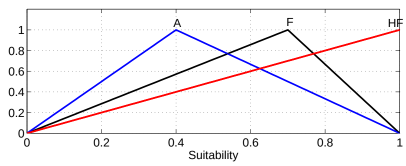

Fig. 22 Membership Functions for $\alpha_{1,2}$,$\alpha_{1,4}$, and $\alpha_{4,j}$ $j = 1, 2 \ldots 5$.

ship functions in Fig. 21 have more fuzzy groups than others. The membership functions for the right and left obstacle avoidance in Fig. 23 are shifted more to the right; this gives the navigation a slight preference to not

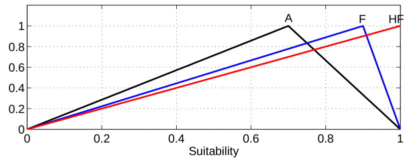

Fig. 23 Membership Functions for $\alpha_{2,3}$ and $\alpha_{3,3}$.

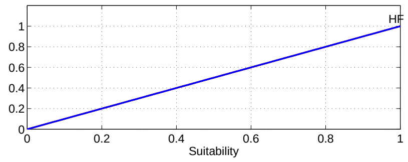

Fig. 24 Membership Functions for $\alpha_{1,1}$, $\alpha_{1,5}$, $\alpha_{2,1}$, $\alpha_{2,2}$, $\alpha_{3,4}$, and $\alpha_{3,5}$.

turn, which reduces the control effort. Due to the realm of influence limitations discussed in Subsection 5.2, the output terms illustrated in Fig. 24 only have a HF term.

The membership functions for $\Delta\theta$ were similar to those shown in Fig. 3, where the universe of discourse was appropriately normalized to limits the maximum turn angle. In the experimental results presented here the maximum turn angle was 35°.

5.5 Command Action Execution in Continuous Time

In simulation, execution of the command action is relatively simple mostly due to the discrete nature of the problem. It uses a simple discrete time kinematic model of the robot with a fixed time step, which makes the determination of the next robot position and orientation a simple calculation. In implementation the control action, which is in terms of a change in angle $\Delta\theta$, could either be sent directly to the robot or translated into a turning rate $\dot{\theta}$ (or ω) for a particular time period Δt where

$$\omega = \dot{\theta} = \frac{\Delta\theta}{\Delta t}. \tag{15}$$

This could possibly produce favorable results, but it is highly dependent on being able to accurately control ω and Δt. If the time step Δt is forced to remain constant it causes ω to be proportional to $\Delta\theta$, which becomes a problem since faster rotational velocities induce more slip that drastically decreases the accuracy of robot localization. Slower rotations also seem to waste time. A piecewise continuous combination of fixed and variable time rate also was considered but not found worth the calculation time in comparison to the direct execution approach. Direct heading control is a much simpler approach, especially since the robot interface contains a function that determines whether or not the heading command is completed. 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 ω_{max}. This physical limitation is attributed to the selected robotic platform and does not significantly slow the algorithm.

5.6 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.

5.6.1 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. First, 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. 25 shows the robot navigating in an obstacle field.

Fig. 25 Robot Navigating Through a Dense Obstacle Field

5.6.2 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 of physical implementation 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, θ) of the robot were recorded and plotted in the same axes as shown in Figs. 26-33. They 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. 26) 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 the feasibilities of both directions are almost identical. Scenario 2 (Fig. 27) 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.

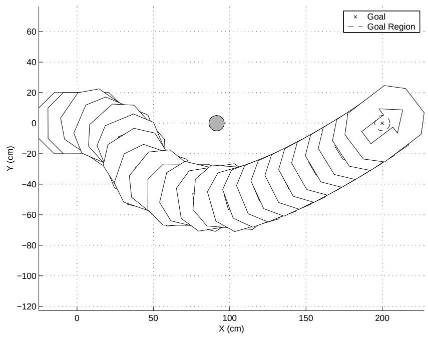

Fig. 26 Experimental Results for Scenario 1

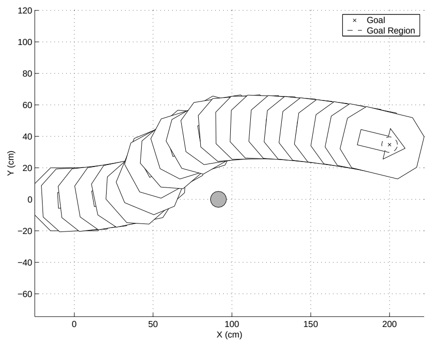

Fig. 27 Experimental Results for Scenario 2

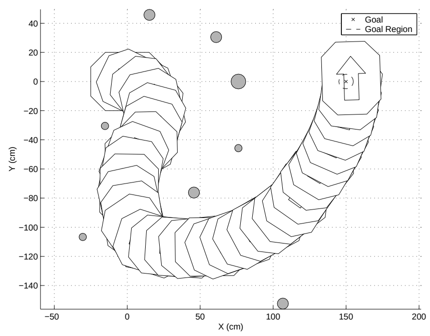

Fig. 28 Experimental Results for Scenario 3

As illustrated by the simulation results, this control system tends to seek the shortest distance to the goal. Scenarios 3 and 4 (Fig. 28 and Fig. 29) represent almost

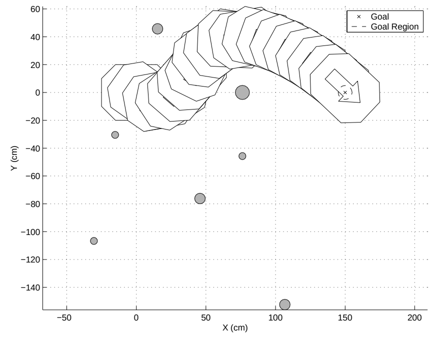

Fig. 29 Experimental Results for Scenario 4

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

Scenarios 5 and 6 (Fig. 30 and Fig. 31) 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. The robot can even move away from the goal after getting very close to it as long as it finds no traversable path direct to the goal.

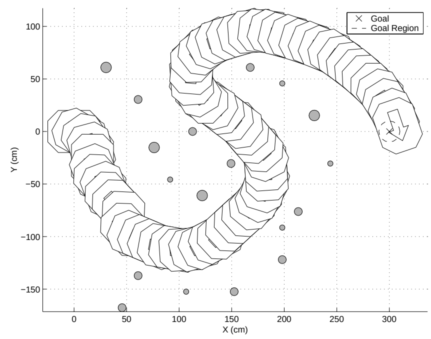

Fig. 30 Experimental Results for Scenario 5

Scenarios 7 and 8 (Fig. 32 and Fig.33) represent a more sparse obstacle configuration that is more typical to an actual forest. There are various traversable paths

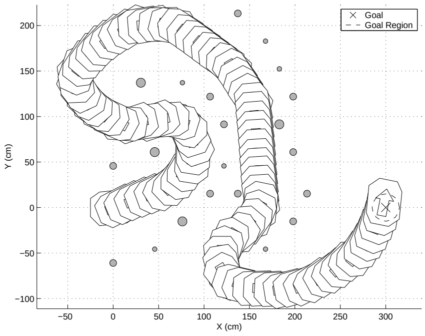

Fig. 31 Experimental Results for Scenario 6

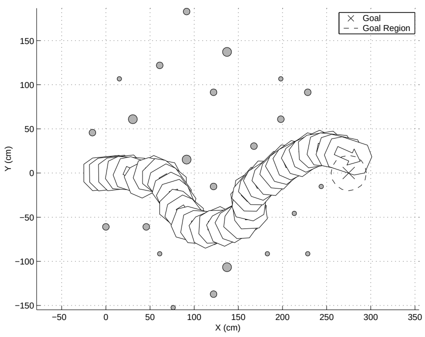

Fig. 32 Experimental Results for Scenario 7

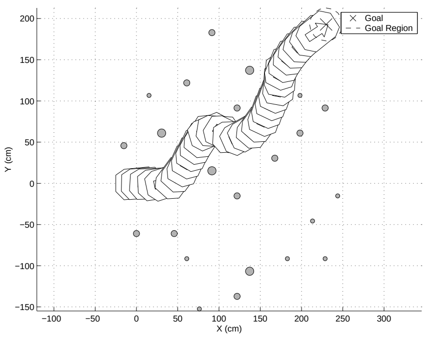

Fig. 33 Experimental Results for Scenario 8

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. 33) 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 make a left turn.

As in all goal-directed reactive methods, this method can have undesired results. The first undesired result happens in situations in which the robot has effectively entered an enclosure, in which case it must effectively retrace its steps to reach the goal. In such cases, the robot can become trapped in a limit cycle and never reach the goal. To prevent this from happening, a separate planner called 'The Deadlock Detection, Retraction and Avoidance' unit is recommended to be added to the proposed system. This planner runs in the background keeping track of places where the robot has visited; whenever it realizes the the robot is revisiting the same places repeatedly, it disables the navigation system described in this paper and starts the retraction and avoidance steps before this navigation system is re-enabled. Further details of such a system are in [Ordonez et al., 2005].

The second undesired results is an effect of the sensor configuration. When making turns, there are instances where the robot comes closer to an obstacle than the desired safe distance. In these circumstances, it is possible for the robot to perceive the inter-obstacle distance as traversable while the gap is actually too narrow. This problem can be addressed by integrating the proposed algorithm with a velocity space method in a similar fashion to the Beam-Curvature Method (BCM) of [Fernández et al., 2004]. This combined method will be able to produce smooth collision-free motion while also taking the vehicle dynamics into account.

6 Conclusion

A preference-based fuzzy behavior control system for robot navigation in dense environments has been presented. We have used the multivalued logic framework [Saffiotti et al., 1995] to develop a control system that enables the robot to navigate smoothly even in very cluttered environments. The increased demands of such environments can cause other behavior control paradigms to be less effective. For example, approaches like the subsumption methods would require switching behaviors at a higher rate, making them less decisive and efficient. In the preference-based system of this paper by not ignoring any behavior, the robot is able to focus towards the goal without being distracted by the presence of multiple obstacles. Experimental results using the proposed method to navigate in a forest with different configurations of multiple obstacles have shown that it can achieve this desired performance. The proposed algorithm represents a significant step in the direction of fully autonomous navigation of unknown dense environments. This work represents a successful application of preference logic to the cluttered environment navigation problem.

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

[Aguirre & Gonzalez, 2000] Aguirre, E., & A. Gonzalez 2000. Fuzzy Behaviors For Mobile Robot Navigation. Design, Coordination and Fusion. International Journal of Approximate Reasoning, 25:225–289.

[Arkin, 1998] Arkin, R. C. 1998. Behavior-Based Robotics. MIT Press, Cambridge Massachusetts.

[Brooks, 1986] Brooks, R. April, 1986. A Robust Layered Control System for a Mobile Robot. IEEE Journal of Robotics and Automation, pages 14–23.

[Combey, 2003] Combey, J. Q. 2003. Comparison of Univalued and Multivalued Fuzzy Behavior Systems For Autonomous Robot Navigation Control. Master's thesis, Florida A & M University, Tallahassee, Florida.

[Division, 2003] Division, Auto Identification June, 2003. Technical Description: LMS 200 Series Laser Measurement Systems. SICK, Inc., 6900 West 110th St., Bloomington, MN 55438, 8 008 970 edition.

[Dunlap, 2004] Dunlap, D. D. 2004. Implementation Of Multi-Valued Fuzzy Behavior Control For Navigation In Cluttered Environments. Master's thesis, Florida A & M University, Tallahassee, Florida.

[Fernández et al., 2004] Fernández, J. L., R. Sanz, J. A. Benays, & A. R. Diéguez 2004. Improving collision avoidance for mobile robots in partially known environments: the beam curvature method. Robotics and Autonomous Systems, 46:205–219.

[Goodridge & Luo, 1997] Goodridge, S. C., & R. C. Luo July, 1997. 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, pages 1622–1627.

[Hodge & Trabia, 1999] Hodge, N. E., & M. B. Trabia 1999. Steering Fuzzy Logic Controller for an Autonomous Vehicle. In Proceedings of the of IEEE Intl. Conference on Robotics and Automation, Detroit Michigan, volume 3, pages 2482–2488.

[Ishikawa, 1991] Ishikawa, S. Nov., 1991. 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, pages 1013–1018.

[Li, 1994] Li, W. June, 1994. 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, pages 1626–1631.

[Linder, 1998] Linder, S. P. September, 1998. Behavior-Based Robust Fuzzy Control. In Proceedings of the 1998 IEEE ISIC/CIRA/ISAS Joint Conference, Gaithersburg, Maryland, USA, pages 233–238.

[Ordonez et al., 2005] Ordonez, C., M. F. Selekwa, E. G. Collins, Jr., & D. Dunlap May 2005. Detection and Retraction from Limit Cycles for Navigation of Robotic Vehicles in Cluttered Environments. In Proc. Florida Conference on Recent Advances in Robotics, Gainesville, Florida.

[Payton et al., 1990] Payton, D. W., J. Rosenblatt, & D. M. Keirsey 1990. Plan Guided Reaction. IEEE Transactions on Systems Man and Cybernetics, 20(6):1370–1382.

[Pin & Watanabe, 1994] Pin, F. G., & Y. Watanabe 1994. Navigation of Mobile Robots Using Fuzzy Behaviorist Approach and Custom Designed Fuzzy Inference Boards. Robotica, 12:491–503.

[Robotics, 2001] Robotics, ActivMedia October, 2001. Pioneer 2 / PeopleBot Operations Manual. ActivMedia Robotics, 44 Concord St., Peterborough NH, 03458, 9 edition.

[Rosenblatt, 1997] Rosenblatt, J. 1997. DAMN: Distributed Architecture for Mobile Navigation. Ph.D. Thesis, Carnegie Mellon University Robotics Institute, Pittsburgh, PA.

[Saffiotti, 1997] Saffiotti, A. July, 1997. Fuzzy Logic in Autonomous Robotics: behavior coordination. In Proceedings of the IEEE International Conference on Fuzzy Systems, Bercelona, Spain, pages 573–578.

[Saffiotti et al., 1995] Saffiotti, A., K. Konolige, & E. H. Ruspini 1995. A multivalued logic approach to integrating planning and control. Artificial Intelligence, 76:481–526.

[Saffiotti et al., 1993] Saffiotti, A., E. H. Ruspini, & K. Konolige June, 1993. Blending Reactivity and Goal Directedness in a Fuzzy Controller. In Proceedings of the 2nd IEEE Conference on Fuzzy Systems, Vol. 1, pages 134–139.

[Selekwa & Collins, 2003] Selekwa, M. F., & E. G. Collins October 2003. A Centralized Fuzzy Behavior Control for Robot Navigation. In Proc. IEEE International Symposium on Intelligent Control, Houston Texas, pages 602–607.

[Selekwa et al., 2004] Selekwa, M. F., E. G. Collins, & J. Q. Combey March, 2004. On Reactive Fuzzy Behavior Systems for Navigation Control of Mobile Robots. In Proceedings of the 10th International Conference of the American Nuclear Society (ANS) on Robotics and Remote Systems for Hazardous Environments, Gainesville, Florida, pages

419–426.

[Seraji & Howard, 2002] Seraji, H., & A. Howard 2002. Behavior-Based Robot Navigation on Challenging Terrain: A Fuzzy Logic Approach. IEEE Trans. Robot. Automat., 18:308–321.

[Sgorbissa & Arkin, 2003] Sgorbissa, A., & R. C. Arkin 2003. Local navigation strategies for a team of robots. Robotica, 21:461–473.

[Siegwart & Nourbakhsh, 2004] Siegwart, R., & I. R. Nourbakhsh 2004. Introduction to Autonomous Mobile Robots. The MIT Press.

[Tunstel, Jr., 1996] Tunstel, Jr., E. W. May, 1996. Mobile Robot Autonomy Via Hierarchical Fuzzy Behavior Control. In Proceedings of the 6th International Symposium on Robotics and Manufacturing (ISRAM'96), Montpellier, France, pages 837–842.

[Yen & Pfluger, 1995] Yen, J., & N. Pfluger 1995. Fuzzy Logic Extension to Payton and Rosenblatt's Command Fusion Method for Mobile Robot Navigation. IEEE Transactions on Systems, Man, and Cybernetics, 25(6):971–978.