2005 American Control Conference
June 8-10, 2005. Portland, OR, USA

Real-Time Navigation for Autonomous Vehicles: A Fuzzy Obstacle Avoidance and Goal Approach Algorithm

Qi Chen and Ümit Özgüner

Abstract— **In this paper, a novel real-time fuzzy navigation algorithm of the off-road autonomous ground vehicle is presented. The navigator's goal is to direct the AGV safely, continuously and smoothly across nature terrain en route to a goal. The proposed navigator consists of two fuzzy controllers, the steering controller and the speed controller. These two controllers are designed separately by mimicking the human performances, yet they work collaboratively. Both the simulation and the demonstration of our AGV in the Grand Challenge 04 justify the performance of our navigator.**

I. Introduction

THIS work describes a fuzzy approach in developing a navigation system for an off-road autonomous ground vehicle (AGV). The task of the navigation system is to guide the AGV toward the goal without collision with obstacles.

Since early 1980s, huge efforts have been paid to develop the obstacle avoidance algorithms. The researches can be classified into two major areas: the global path planning [1][2] and the local motion planning [3][4]. For an AGV, it is likely to be able to sense only the nearby obstacles, which constraints the applications of global path planning methods. On the other hand, the local motion planning methods dynamically guide the AGV according to the locally sensed obstacles, which requires less prior knowledge about the environment. Therefore, the local motion planning methods are more suitable and practical for the off-road AGVs, since the environment is too complicated to be known precisely and may be time-varying.

Motivated by the fact that human performance is reliable in driving the ground vehicle, fuzzy logic navigation methods have been proposed to substitute the human performance [5]. Moreover, the fuzzy logic has the feature to make it a useful tool to cope with the large amount of uncertainty that is inherent of natural environments, as noted by Saffiotti [6]. Most of the existing fuzzy approaches, as [5], tend to design toward-target mode and avoid-obstacle mode. The navigator switches between the two modes according to the distance to the obstacles. Therefore, the trajectory is not suitable for most nonzero-turn-radius AGVs.

In this paper, we designed two heterogeneous fuzzy controllers for the AGV navigator. The steering controller emphasizes on the goal reaching and plays an important role in obstacle avoidance at the same time, while the speed controller helps on collision avoidance. These two fuzzy controllers collaborate to safely direct the AGV to the goal.

Both authors are with Department of Electrical and Computer Engineering, The Ohio State University, Columbus, Ohio 43210, USA. {chenq, umit}@ece.osu.edu

As an example, we shall explain our application of this navigation algorithm in our AGV in the DARPA's Grand Challenge in 2004 (GC04). At the Ohio State University, we have had extensive experience in developing AGVs. Our vehicles performed excellently in the structured environments supported by multiple sensor fusion [7][8][9][10]. Nevertheless, as far as off-road is concerned, the environment being unstructured, a stronger navigation algorithm is required to deal with more complicated situations. The performance of our AGV in GC04 justified this navigation algorithm.

This paper is organized as following. Section II describes the design for the both steering and velocity controller. Our simulation result and our AGV performance are presented in Section III. Section IV concludes the paper.

II. Navigator design

To design the navigator, there are some general assumptions which are satisfied by most AGV systems. First, the ego states information of the AGV, such as the speed, direction, position, are available to the navigator. Second, the navigator possess the local information around the vehicle, i.e. the boundary of the nearby obstacles are known to the navigator. Third, the navigator's task is to generate a serial of GPS points and a speed set-point. It is assumed that the AGV can be controlled to follow the reference points at the speed set-point with bounded error.

The notations used in this paper are listed in Table I.

A. Steering Fuzzy controller

The steering fuzzy controller focus more on the goal reaching while keeping obstacle avoidance.

Figure 1 shows one simple scenario that explains the basic steering rules. Based on the local sensor ability, only the boundary of the obstacle in the local sensor region is detectable. It is impossible to know the shape and size of the whole obstacle outside of the sensing region.

Practically, we expand the boundary of the obstacle with half size of the vehicle so that we can consider the vehicle as a mass point. This expanded obstacle boundary is then called the "real boundary" henceforth. Furthermore, to ensure the safety of the AGV, the obstacles is further enlarged to the "safe boundary", so that the real boundaries are enveloped with the safe boundaries.

1) Basic Fuzzy Controller: By the two-step obstacle extension, each obstacles have two different boundaries, the "real boundary" and the "safe boundary", as shown in Figure 1. The scan distances from the AGV to both boundaries inside the local sensor region

TABLE I

THE NOTATION LIST

$\theta(k)$	the direction of the vehicle at the time k
$v(k)$	the speed of the vehicle at the time k
$\phi(k)$	the goal direction at the time k
$D_R(\theta)$	the distance from the vehicle to the real-obstacle-boundary in the direction θ
$D_S(\theta)$	the distance from the vehicle to the safe-obstacle-boundary in the direction θ
$\theta_{RD}(k)$	the decision based on the real-obstacle-boundary
$\theta_{SD}(k)$	the decision based on the safe-obstacle-boundary
$\theta_D(k)$	the final decision direction at the time k
d_{max}	the maximum sensing distance
$\theta_{min}(k)$	the start angle of the obstacle scanning, $\theta(k) - \frac{\pi}{2}$ is used in this paper
$\theta_{max}(k)$	the end angle of the obstacle scanning, $\theta(k) + \frac{\pi}{2}$ is used in this paper
$\mathcal{I}$	the candidate set of $\theta_{RD}(k)$
$\mathcal{J}$	the candidate set of $\theta_{SD}(k)$
$d_{CR}(\theta)$	the criterion distance of direction θ, used in $\mathcal{I}$ and $\mathcal{J}$
F_s	the choice strategy in choosing the $\theta_D(k)$
F_D	the fuzzy strategy in deciding the $d_{CR}(\theta, k)$
D_{safe}	the minimum distance that the AGV is required to keep from obstacles in the front
$D1$	the distance to the obstacle in front of the vehicle, which is $D_R(\theta(k))$
$D2$	the nearest distance to the obstacle within the angles from $\theta(k)$ to $\theta_D(k)$
$D3$	the distance to the obstacle in the decision direction, which is $D_R(\theta_D(k))$
V_0	the base speed, a reference value
$V_s(k)$	the speed set point that the speed controller generates
A	the anti-collision strategy for speed control
T	the safe-sharp-turn strategy for speed control

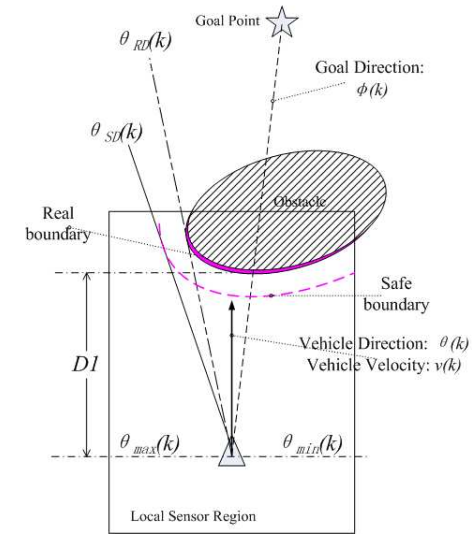

Fig. 1. Local navigation scenario

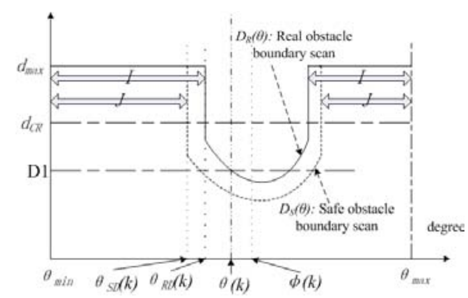

Fig. 2. The scan distance to the obstacle boundaries at time k

is available to the controller. Figure 2 represents the scan distances to the obstacle boundaries shown in Figure 1.

The basic steering strategy is to select the direction that has the smallest angle difference to the goal direction. The formulas are expressed in 1a-d.

$$\mathcal{I} = \{\theta | D_R(\theta) > d_{CR}\} \tag{1a}$$

$$\mathcal{J} = \{\theta | D_S(\theta) > d_{CR}\} \tag{1b}$$

$$\theta_{RD}(k) = \arg\min_{\theta \in \mathcal{I}} \{|\theta - \phi(k)|\} \tag{1c}$$

$$\theta_{SD}(k) = \arg\min_{\theta \in \mathcal{J}} \{|\theta - \phi(k)|\} \tag{1d}$$

where $\mathcal{I}$ and $\mathcal{J}$ are the candidate sets of angles at which the distance to the obstacle is larger than the criterion, d_{CR}.

Equations (1a-d) give the method in finding two directions, $\theta_{RD}(k)$ and $\theta_{SD}(k)$, as decisions based on the real boundary and the safe boundary scan distances respectively.

F_s, in equation (2), represents the rule of selecting $\theta_D(k)$, the final decision direction of the steering controller. Table II shows the selection rule.

$$\theta_D(k) = F_s(\theta_{RD}(k), \theta_{SD}(k), \phi(k)) \tag{2}$$

The "Small", "Medium" and "Large" in the Table II are fuzzy descriptions. The defuzzification function picks either $\theta_{SD}(k)$ or $\theta_{RD}(k)$, as the result of $\theta_D(k)$, according to which one the weight center of the result is closer to. In other words, the result would not be some value between $\theta_{SD}(k)$ and $\theta_{RD}(k)$.

Remark 1: If $\theta_{SD}(k)$ is chosen to be the steering decision, then there is an open space in the direction so that vehicle is allowed to pick up a relatively high speed safely. On the other hand, if $\theta_{RD}(k)$ is chosen, it can be concluded that $|\theta_{SD}(k) - \theta_{RD}(k)|$ is large, i.e. the $\theta_{SD}(k)$, presenting the open free space, is in the other direction, and the chosen direction must be a narrow path. Therefore, a relatively low speed and carefulness are recommended.

The criterion distance, d_{CR}, in the equations (1a, 1b) is

TABLE II

THE SELECTION RULE

		$\lvert\theta_{SD}(k) - \theta_{RD}(k)\rvert$		
	$\theta_D(k) =$	Small	Medium	Large
$\lvert\theta_{SD}(k) - \phi(k)\rvert$	Small	$\theta_{SD}(k)$	$\theta_{SD}(k)$	$\theta_{RD}(k)$
	Medium	$\theta_{SD}(k)$	$\theta_{RD}(k)$	$\theta_{RD}(k)$
	Large	$\theta_{SD}(k)$	$\theta_{RD}(k)$	$\theta_{RD}(k)$

an important parameter in this basic steering controller. If d_{CR} is carefully selected, the basic fuzzy steering controller works well when there are few obstacles in the environment and the obstacles are all convex-shaped.

However, this is not generally the case in off-road environments. When the environment has many obstacles and the obstacles are concave-shaped, there exist four major problems in this basic steering controller. First, if d_{CR} is chosen too large, then both $\mathcal{I}$ and $\mathcal{J}$ could be *empty* sets. On the other hand, if d_{CR} is chosen too small, then both $\mathcal{I}$ and $\mathcal{J}$ equal to the interval $[\theta_{min} \quad \theta_{max}]$, hence $\theta_D(k) = \theta_{SD}(k) = \theta_{RD}(k) = \phi(k)$. In other words, there is no obstacle avoidance. The second problem is that the basic steering controller sometimes makes wide left turn or wide right turn instead of finding a path in the front. When sensors fail to detect the obstacles on both sides or there is no obstacles in either side, the steering controller regards it as a wide free space and select it as decision. In other words, it would rather detour than find path in its front. The third problem is that the AGV could get *stuck* at some concave-shaped obstacle surface or some complicated compositions of obstacles if d_{CR} is not selected good enough. The fourth problem is that the navigator may generate *discontinuous* output of $\theta_D(k)$, i.e. $|\theta_D(k) - \theta_D(k-1)|$ is large. In most cases, the last two problems happen together and cause the AGV to zigzag toward an obstacle and get stuck.

The following rules are proposed to solve all these problems, so that our navigator can deal with most of complicated situations and guide the AGV to the goal safely.

2) First Fuzzy Rule:
To solve the first problem, we define a fuzzy rule to determine the suitable value of $d_{CR}(k)$:

$$d_{CR}(k) = F_D(v(k)) \tag{3}$$

where $v(k)$, the vehicle speed, is fuzzified into (*LOW, MED, HIGH*), representing low, medium and high speed. The triangular membership functions are used. This fuzzy rule maps the low speed into small d_{CR} and vice versa.

In general, we can assume here that the velocity controller works properly so that it slows down the AGV while the distance to the obstacle decreases and stops the AGV before the distance is too small, say D_{safe}.

Remark 2: If $F_D(0) <= D_{safe}$ and proper velocity controller works, then set $\mathcal{I}$ or $\mathcal{J}$ is nonempty. By reducing the d_{CR}, the set $\mathcal{I}$ or $\mathcal{J}$ could not be empty by how they are define in (1a-b).

To deal with the remaining problems, we develop the following fuzzy rules by mimicking the human performance, such as "Focus","Search" and "Persist" properties.

3) Focus Rules:
The second problem that the AGV may make wide left or right turns is caused by the failure of detecting the obstacle on both sides of the AGV. However, human drivers don't have the problem even if their side views are restricted. It is observed that human drivers pay more attention to the front paths than to the side ones. This property is studied and applied in our steering controller as the "Focus" rules. Formulas are represented as below:

$$\mathcal{I}(k) = \{\theta | D_R(\theta) > d_{CR}(\theta, k)\} \tag{4a}$$

$$\mathcal{J}(k) = \{\theta | D_S(\theta) > d_{CR}(\theta, k)\} \tag{4b}$$

$$d_{CR}(\theta, k) = F'_D(v(k), \phi(k), \theta(k)) = F_D(v(k)) - \Delta \tag{5}$$

where $d_{CR}(\theta, k)$, the looking ahead distance at the direction θ at the time k, is enforced by the "Focus" rules. The equations (1a,b) are improved to equations (4a,b). $F'_D(\cdot)$ in (5) is improved from (3) by adding the term "$-\Delta$". The function of $F'_D(\cdot)$ is to reduce the value of $d_{CR}(\theta, k)$ so as to increase the priority in direction θ, when $|\theta - \theta(k)|$ and $|\theta - \phi(k)|$ are small.

Let $\Delta 1$ and $\Delta 2$ represent the $|\theta - \theta(k)|$ and $|\theta - \phi(k)|$ respectively. Let (S, MS, M, ML, L) represent (Small, Medium Small, Medium, Medium Large and Large) respectively. The fuzzy "Focus" rules that work on the Δ in the equation (5) are as following:

- If ($\Delta 1$ is S) and ($\Delta 2$ is S), then (Δ is L): if θ is toward the forward and toward the goal direction, then the direction has the higher priority to be chosen, i.e. the $d_{CR}(\theta, k)$ has a larger deduction.
- If (($\Delta 1$ is S) and ($\Delta 2$ is M)) or (($\Delta 1$ is M) and ($\Delta 2$ is S)) or (($\Delta 1$ is M) and ($\Delta 2$ is M)), then (Δ is ML).
- If (($\Delta 1$ is S) and ($\Delta 2$ is L)) or (($\Delta 1$ is L) and ($\Delta 2$ is S)), then (Δ is M).
- If (($\Delta 1$ is L) and ($\Delta 2$ is M)) or (($\Delta 1$ is M) and ($\Delta 2$ is L)), then (Δ is MS).
- If ($\Delta 1$ is L) and ($\Delta 2$ is L), then (Δ is S): the angles that are neither forward nor the goal direction should have a lower priority, so that they are less likely to be chosen.

The example shown in Figure 3 is a snap shot from our simulator. The left figure is the "real world", in which the obstacles are already expanded to the real boundary. The measurements of the obstacle boundaries, both real and safe boundaries, are displayed in the middle figure. The scan distances to the boundaries are then shown in the right figure, where the vehicles direction $\theta(k)$ and the goal direction $\phi(k)$ are marked in the figure. The $d_{CR}(\theta)$ is a V-shaped curve shown in the figure. Note that the curve can be a W-shaped curve when $\theta(k)$ is separated away from $\phi(k)$. By selecting the direction θ_D, the steering controller generates a curve of path points shown in the middle figure, following which the AGV avoids the obstacle.

In this fuzzy method, triangle membership functions are used to define the fuzzy sets for the $\Delta 1$ and $\Delta 2$, as shown in Figure 4. The values of (a, b, c, d) determine the membership function, consequently determines the shape, especially the width, of the notch in the curve of $d_{CR}(\theta)$. On the other hand, the vales assigned to (S, MS, M, ML, L) for Δ determines the depth of the notch. The larger is the values, the deeper is the notch.

It is important to select the values of (a, b, c, d) carefully. When values are small, fewer angles are classified into S, so

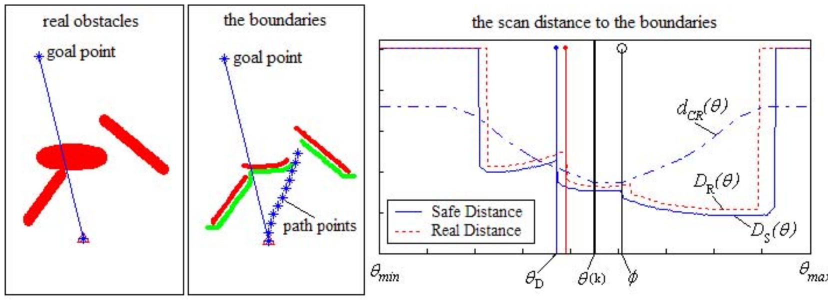

Fig. 3. The example of the focus rules

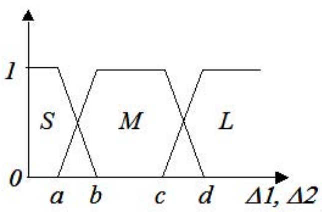

Fig. 4. The membership functions for $\Delta 1$ and $\Delta 2$

that $d_{CR}(\theta)$ is reduced only in narrow range of directions. Thus, the "Focus" effort is enhanced. On the other hand, if the values are large, then the "Focus" effort is neutralized and the navigator has a wide range of direction to select from.

For some θ, if $d_{CR}(\theta) > D_R(\theta)$, then $\theta \notin \mathcal{I} \cup \mathcal{J}$ by the focus rule, so that it is excluded from being selected as θ_D. Therefore, a specific *priority window* is opened for the navigator to choose within. Thus, the navigator does not choose the directions in the left or right side, unless the goal direction is in the wide left or right. The second problem is solved by the focus rules.

4) Focus vs. Search Rules:
The focus rules, equations (4a-b, 5), prevent the distraction by side directions quite well.The focus rules trade the ability in searching the feasible directions for the distraction prevention. It works well when there are few obstacles in the environment. However, when there are many obstacles in the environment, the focus rule impair the search capability of the navigator.

Furthermore, as stated in previous section, the values of (a, b, c, d) in the membership function of the fuzzy sets are important parameters of the focus rules. The smaller values of (a, b, c, d) correspond to stronger "Focus" and vice versa.

Motivated by this, the rule (5) is then improved by considering the fuzzy evaluation of the AGV status. The $v(k)$, the vehicle's speed, is regarded as the index of vehicle's situation: the lower the speed is, the more search capability is required. Therefore, the equations are:

$$S(k) = F_{ST}(v(k), V_0) \quad (6)$$

$$d_{CR}(\theta, k) = F''_D(v(k), \phi(k), \theta(k), S(k)) \quad (7)$$

where $S(k)$ is the situation classification, a membership function value of the fuzzy sets: *exploring* and *travelling*. The membership function is triangular. The rule $F_{ST}(\cdot)$ states: When the $v(k)/V_0$ is small, $S(k)$ is "*exploring*" (EX). On the other hand, when $v(k)/V_0$ is large, $S(k)$ is "*travelling*" (TR). The speed, $v(k)$, is classified by comparing rather to the base speed than to the absolute values. The base speed, V_0, is provided to the AGV according to different types of terrain surfaces.

The difference between (7) and (5) is that the membership function parameters, the (a, b, c, d), are not fix values. The fuzzy rules lie in (7) are as following:

- If $S(k)$ is EX, then the values of (a, b, c, d) increases: When the AGV is nearing obstacles, we assume that the velocity controller slows down the AGV. In this case, the $S(k)$ becomes more "EX", so that $\mathcal{I}$ and $\mathcal{J}$ is enlarged and the focus rules are neutralized.
- If $S(k)$ is TR, then the values of (a, b, c, d) reduces: vice versa, the focus rules are enhanced in this case.

By applying the rules (6,7), our simulation show that the AGV is much less likely to be stuck in front of the concave-shaped obstacles, though it may be stuck in some extreme situations like a very deep-U-shaped obstacle.

When the AGV approaches the obstacle, the speed is reduced so that the "priority window" is open up for search and the "focus" effect is neutralized. The "Search" rules help to prevent the AGV from being stuck. This solves the third problem.

5) Persistence Rules:
The last problem is the *discontinuous* outputs from the steering controller that causes the oscillation of the vehicle's heading. To solve this problem, the selecting rules defined in (2) is modified as following.

$$\theta_D(k) = F'_s(\theta_{RD}(k), \theta_{SD}(k), \phi(k), \theta_D(k-1)) \quad (8)$$

The new choosing strategy, F'_s, is improved from the memoryless strategy F_s in (2). To chose $\theta_{RD}(k)$ or $\theta_{SD}(k)$, the "persistence" rules consider the last decision, $\theta_D(k-1)$. When $|\theta_{SD}(k) - \theta_D(k-1)|$ is small, the controller prefer to chose $\theta_{SD}(k)$ rather than $\theta_{RD}(k)$, and vice versa.

By doing so, the "Persistence" rules put weight on the last decision and give the controller a characteristic of persistence, therefore kills the zigzag performance of the AGV and solves the last problem.

Remark 3: The fuzzification of the $|\theta_{SD}(k) - \theta_D(k-1)|$ can be introduced to (7), instead of using it in (8), so that the searching window is adjusted to raise the priority of the formal decision, so called the persistence.

Finally, the steering fuzzy controller in our system consists of the following equations: 6, 7, 4a-b, 1c-d and 8, together with the corresponding fuzzy rules.

When the decision direction is selected, a sequence of path points in GPS coordination is then generated as the steering output of the navigator.

B. Speed Fuzzy Controller

The velocity controller's main aim is to avoid the collision into the obstacles. Moreover, due to the physical constraint of the vehicle, sharp turning at high speed must be forbidden to prevent the AGV from rolling over. Therefore, the velocity fuzzy controller consists of two sets of rules: the *Anti-collision rules* and the *safe-sharp-turn rules*.

1) Anti-collision Rules:
The main purpose of the collision rules is to prevent the collision. Furthermore, the rules are to guarantee the condition of remark 2 holds. Let A be the anti-collision rules.

$$V_s(k) = V_0 \times A(D1(k), D2(k), D3(k)) \tag{9}$$

The fuzzy rules, A, have three dimensions of inputs. Let (S, MS, M, ML, L) have the same meaning as stated above, and let (DA) resent the fuzzy set of dangerous distance. The rules are stated as following:

- If (one of $(D1, D2, D3)$ is DA), then V_s is $STOP$: the vehicle should stop when the distance to the obstacle is too close and it is dangerous to move on.
- If (two of $(D1, D2, D3)$ are S) and (the other is S or M), then V_s is S: the vehicle speed should be reduced when it is close to the obstacles.
- If (one of $(D1, D2, D3)$ is S) and (the other two are M), then V_s is MS.
- If (two of $(D1, D2, D3)$ are M) and (the other one is M or L), then V_s is M.
- If (two of $(D1, D2, D3)$ are L) and (the other one is M), then V_s is ML.
- If (all of $(D1, D2, D3)$ are L), then V_s is L: no obstacle is nearby, the speed is set to a high value.

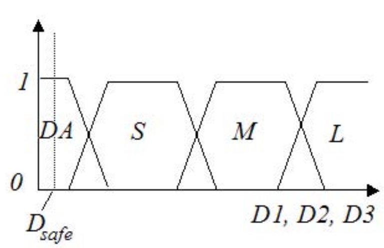

Fig. 5. The membership functions for $D1, D2$ and $D3$

The boundaries of the fuzzy set is selected according to the kinematic performance of the vehicle and how much risk you want to take. The membership functions used in our system are as shown in Figure 5.

The membership functions for $D1, D2$ and $D3$ are the same in our system, although they may be different with the boundaries of the fuzzy sets. Note that the D_{safe}, used in remark 2, is marked in the figure, which is totally within the fuzzy set DA. Therefore, the AGV stops before reaching the distance D_{safe} so that collision is prevented. If the vehicle is stopped by the rules because it is dangerously close to an obstacle, it is not recoverable by the navigator itself. In general, this case does not happen. If so, the situation analysis module of the AGV system changes the mode from navigation to other modes to pull back the vehicle and redirect the vehicles heading, which is beyond the discussion of this paper.

2) Safe-Sharp-Turning Rules:
The other issue is the safety in making a turn. If a high speed set point and the sharp-turning command were sent to the AGV, it might cause the AGV to roll over by the centrifugal effect. To prevent this danger, the safe-sharp-turning rules are defined as following.

$$\Delta\theta = |\theta_D(k) - \theta_D(k-1)|$$
$$V_s(k) = V_0 \times (A(D1, D2, D3) - T(\Delta\theta)) \tag{10}$$

The formula (10) is improved from (9) by adding the rule T, which studies the value of $|\theta_D(k) - \theta_D(k-1)|$, say $\Delta\theta$, the turning rate. Being a SISO rule, the rule T says: If ($\Delta\theta$ is S) then T is S; If ($\Delta\theta$ is M) then T is M; If ($\Delta\theta$ is L) then T is L. Therefore, the speed set point of the AGV is reduced when the sharp turn command is sent. This prevent the rolling-over and guarantees the safety of the AGV.

Finally, the speed fuzzy controller in our system is of equation (10), together with the corresponding fuzzy rules.

Remark 4: As noted in remark 1, if $\theta_{RD}(k)$ is chosen, a relatively low speed are recommended by multiplying a coefficient less than 1.

In the speed controller, the original steering controller result $\theta_D(k)$ is used in the safe-sharp-turning rules. On the other hand, the vehicle speed $v(k)$ is used as an important parameter in the steering controller. Therefore, these two fuzzy controllers works collaboratively, although they are designed separately. Furthermore, this cooperation matches the human driving performance: reduce the speed if you want to make a sharp turn!

III. Performance of the Navigator

The navigator designed in this paper is justified by both the simulation performance and its application in our AGV.

We built a simulator to test the navigator designed in this paper. In the simulator, the obstacles are expanded, therefore, the vehicle is regarded as a point mass. The vehicle model in this simulator is commonly known as the Dubins' car model with a minimum turning radius.

We assume that there are some known obstacles of "map-level" and pre-planned route points are given. This route points is not accurate enough to guide the AGV directly

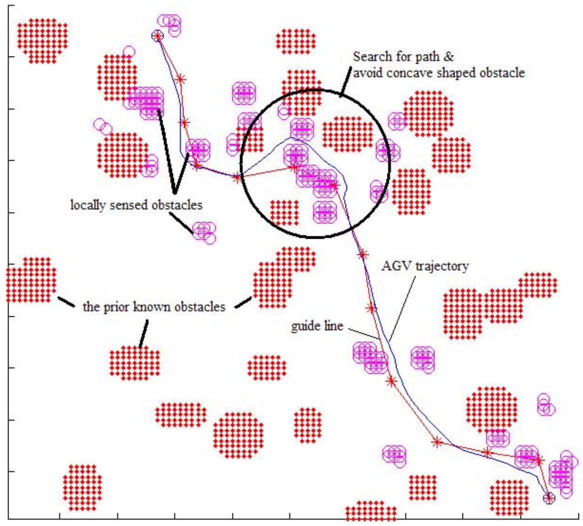

Fig. 6. The AGV trajectory over the obstacle-occupied terrain

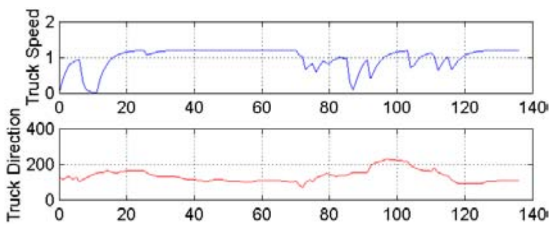

Fig. 7. The AGV speed and heading profile along the run

since there are also unknown obstacles in the field which is not prior known.

Figure 6 shows a simulation result of a AGV over a terrain in which obstacles are randomly generated. The bottom right circle is the starting point and the up left circle is the final goal. The (red) dots represent the prior known obstacles. The (pink) circles along the trajectory are the unknown obstacles that the AGV locally sensed. The (red) star points are the guideline points. The trajectory of the AGV, the smooth (blue) line, indicates that the AGV follows the given guideline points, avoids all obstacles and reaches the goal smoothly and safely. The circled part in the map shows that our algorithm searches the path when surrounded by many obstacles without being stuck by the concave-shaped obstacle.

Figure 7 is the speed and heading profile of the AGV along the simulation run. It clearly shows that the speed was reduced whenever sharp-turning was encountered. The profile also shows that the speed was reduced whenever the AGV was heading close to obstacles.

The performance of the navigator designed in this paper was also demonstrated by our AGV, known as TerraMax, in the DARPA GC04 event [11][12][13][14]. Our AGV was one of the just seven vehicles that completely traversed the QID course [14], which fully exhibited the obstacle-avoidance and goal-approaching capability of this fuzzy navigator and our AGV system. The performance of our AGV in the GC04 also verified the general assumptions that we have made in Section II.

IV. Conclusion

In this paper, a novel local navigation system was designed to guide our AGV, the TerraMax, which was one of the just seven vehicles that completely traversed the QID course in GC04 [14]. The navigator relies only on the locally sensed information and internal sensed ego states to direct the AGV.

By mimicking the human driver's performance, our navigator separates the steering controller and the speed controller, yet they work collaboratively to keep the AGV safe. Derived from a basic obstacle-avoiding algorithm, a series of fuzzy rules are designed and discussed in this paper, including the basic fuzzy controller, the first rule, focus rules, focus vs. searching rules and the persistence rules. Both the snapshot and the simulator results show the success of our navigator.

References

[1] C. Warren, "Global path planning using artificial potential fields," in *Proc. IEEE International Conference on Robotics and Automation*, (Scottsdale, AZ), pp. 316–321, May 1989.

[2] S. Kambhampati and L. Davis, "Multiresolution path planning for mobile robots," *IEEE Journal on Robotics and Automation*, vol. 2, pp. 135–145, Sept. 1986.

[3] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proc. IEEE International Conference on Robotics and Automation*, (St. Louis, MO), pp. 500–505, Mar. 1985.

[4] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 19, pp. 1179–1187, Sept.-Oct. 1989.

[5] N. Hodge and M. Trabia, "Steering fuzzy logic controller for an autonomous vehicle," in *Proc. IEEE International Conference on Robotics and Automation*, (Detroit, MI), pp. 2482–2488, May 1999.

[6] A. Saffiotti, "The uses of fuzzy logic in autonomous robot navigation," *Soft Computing*, vol. 1, pp. 180–197, Dec. 1997.

[7] Ü. Özgüner, C. Hatipoglu, and K. Redmill, "Autonomy in a restricted world," in *IEEE Conference on Intelligent Transportation System ITSC'97*, (Boston, MA), pp. 625–630, Nov. 1997.

[8] K. Redmill and Ü. Özgüner, "The Ohio State University automated highway system demonstration vehicle," in *SAE Paper 980855, 1998SAE International Congress and Exposition*, (Detroit, MI), pp. 117–130, Feb. 1998.

[9] Ü. Özgüner, K. Redmill, Ü. Ogras, O. Dagci, and M. Launsbach, "Autonomous vehicles in structured and semi-structured environments," in *Proc. of the 41st IEEE Conference on Decision and Control*, (Las Vegas, NV), pp. 124–129, Dec. 2002.

[10] C. Hatipoglu, Ü. Özgüner, and K. Redmill, "Automated lane change controller design," *IEEE Transactions on Intelligent Transportation Systems*, vol. 4, pp. 13–22, Mar. 2003.

[11] Ü. Özgüner, K. Redmill, and A. Broggi, "Team terramax and the darpa grand challenge: A general overview," in *Procs. IEEE Intelligent Vehicles Symposium 2004*, (Parma, Italy), pp. 232–237, June 2004.

[12] Q. Chen, Ü. Özgüner, and K. Redmill, "The Ohio State University team at the grand challenge 2004: Developing a completely autonomous vehicle," *IEEE Intelligent Systems*, vol. 19, pp. 8–11, Sept.-Oct. 2004.

[13] H. Yu, Q. Chen, and Ü. Özgüner, "Control system architecture for terramax - the off-road intelligent navigator," in *5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles*, (Lisboa, Portugal), July 2004.

[14] "http://www.darpa.mil/grandchallenge/." DARPA Grand Challenge Website.