A Fuzzy Logic Controller for Autonomous Wheeled Vehicles

Mohamed B. Trabia
University of Nevada, Las Vegas
U.S.A.
Linda Z. Shi
University of California, San Diego
U.S.A.
Neil E. Hodge
University of California, Berkeley
U.S.A.

1. Introduction

Autonomous vehicles have potential applications in many fields, such as replacing humans in hazardous environments, conducting military missions, and performing routine tasks for industry. Driving ground vehicles is an area where human performance has proven to be reliable. Drivers typically respond quickly to sudden changes in their environment. While other control techniques may be used to control a vehicle, fuzzy logic has certain advantages in this area; one of them is its ability to incorporate human knowledge and experience, via language, into relationships among the given quantities. Fuzzy logic controllers for autonomous vehicles have been successfully applied to address various (and sometimes simultaneous) navigational issues,including:

- reaching a static target (Baturone, et al., 2004, Boada, et al., 2005, El Hajjaji & Bentalba, 2003, Maeda et al., 1991, Chen & Ozguner, 2005),
- tracking moving targets (Ollero, et al., 2001),
- maintaining stable vehicular velocity (Holzmann et al., 1998, Nobe & Wang, 2001),
- following a lane or a wall (Rosa & Garcia-Alegre, 1990, Hessburg & Tomizuka, 1994, Peng & Tomizuka, 1993) and,
- avoiding collision with static and dynamic obstacles (Baturone, et al., 2004, Murphy, 2001, Godjevac, et al., 2001, Seraji, 2005, Lee & Wang, 1994, Wheeler & Shoureshi, 1994, Ye & Wang, 2001).

Several researchers combined fuzzy logic controllers with various learning techniques, such as:

- supervised learning method (Godjevac, 2001),
- evolutionary method (Hoffman, 2001, Kim, et al., 2001),
- neural network (Pasquier, et al., 2001, Cang, et al., 2003),
- reinforcement learning (Dai, et al., 2005) and,
- optimization methods (Hong, 1997, Sanchez, et al., 1999).

To accomplish most tasks of a mobile robot, the controller must be able to adjust the steering angle and the velocity of the vehicle simultaneously. Most of the presented work below considers one or two aspects of the problem. For example, (Rosa & Garcia-Alegre, 1990) proposed a fuzzy logic controller that can modify the speed and the steering of a mobile robot to steer it at a fixed distance along a wall. Using a kinematic model and a variety of wall shapes, including cubic splines and line segments with ninety-degree intersections, they designed a fuzzy logic controller with a minimum number of rules to accomplish the task. (Maeda, et al., 1991) implemented fuzzy logic for the steering and the speed control of a two-wheel drive robot. The environment in which the control scheme was tested was limited to straight-line paths with a minimum number of ninety-degree turns. Fuzzy logic was incorporated in lateral vehicle control to solve the lane following problem (Hessburg & Tomizuka, 1994). They tested their design on a dynamic model first, and then implemented it in an actual vehicle to evaluate it. The experiment was to drive the vehicle on a road with multiple smooth curves interspersed by straight segments. (Peng & Tomizuka, 1993) described an engine throttle fuzzy logic controller for accelerating and decelerating the vehicle during typical driving conditions. The proposed controller had no provisions for obstacle avoidance. (Lee & Wang, 1994) proposed the use of fuzzy logic to assist in obstacle avoidance. Their simulation included both static and moving obstacles. (Wheeler & Shoureshi, 1994) controlled a vehicle on handling track using fuzzy logic. Their model included the vehicle's dynamic behavior. The handling track consisted of a set of pylons (known static obstacles) that the vehicle must navigate at high speeds (i.e., around 50 miles per hour). (Holzmann et al., 1998) combined a conventional vehicle acceleration controller with a fuzzy logic controller for vehicle velocity and inter-vehicle distance. (Sanchez et al., 1999) presented an off-line two-level fuzzy logic controller to track a path previously recorded or computed by means of a path planning program. The number of fuzzy rules was optimized to improve performance. (Ye & Wang, 2001) presented a novel navigation method for an autonomous vehicle in unknown environments. Their navigator consisted of an Obstacle Avoider (OA), a Goal Seeker (GS), a Navigation Supervisor (NS) and an Environment Evaluator (EE). The fuzzy actions inferred by the OA and the GS were weighted by the NS using the local and global environmental information and fused through fuzzy set operation to produce a command action. (Nobe and Wang, 2001) reviewed recent developments in advanced vehicle control systems (AVCS) including lateral steering, longitudinal throttle control, and integration of these controls for vehicles. Autonomous intelligent cruise control (AICC) and cooperative intelligent cruise control (CICC) were considered for a platoon of two or more closely spaced vehicles travelling with same velocity in a same lane. Look-down and look-ahead systems for automated steering were presented for the lateral vehicle control. (Kodagoda et al., 2002) developed and implemented fuzzy proportional derivative-proportional integral (PD-PI) controllers for the steering and the speed control of an autonomous guided vehicle. (Ohara & Murakami, 2004) proposed a model for a compact tractor-trailer using an electrical vehicle that estimated the cornering force and friction and changed speed and steering angle based on the inputs. The proposed PD controller can avoid the jackknife phenomenon of the vehicle. (Chen and Ozguner, 2005) presented a real-time fuzzy navigation algorithm for off-road autonomous ground vehicles. The controller's goal was to direct the vehicle safely, continuously and smoothly across natural terrain to reach a goal. The proposed navigator consisted of two fuzzy controllers, the steering

controller and the speed controller. These two collaborative controllers were designed separately by mimicking human performance.

The objective of this chapter is to describe a fuzzy logic controller for the steering and the velocity of an automobile. Observation indicates that drivers tend to separate the various driving tasks. For example, most drivers conceptualize velocity and direction separately. This separation of objectives is the basis of the proposed distributed fuzzy logic controller for the vehicle. A controller for the steering of an autonomous vehicle needs to achieve these objectives:

- steering the vehicle toward the target,
- steering the vehicle around any obstacle to avoid a collision,
- avoid being trapped in maze or cluttered environment,
- steering the vehicle to stop at a desired orientation.

The velocity controller should emulate the following human behaviors:

- starting the vehicle from a complete stop, and stopping it when it reaches the target,
- slowing down the vehicle when it approaches an obstacle and speeding it up as it moves beyond the obstacle,
- slowing down the vehicle when its turning radius decreases (i.e., the tighter the turn, the lower the velocity).

A fuzzy logic controller module, of the Mamdani-type, is created for each of these objectives.

The following is a brief summary for the remainder of this chapter. The second section briefly describes the nonlinear model of the vehicle dynamics. The third section details the first two modules of the steering controller, which meet the fundamental driving requirements: the Target Steering Fuzzy module, and the Collision Avoidance Steering Fuzzy module. The fourth section describes the third and the forth modules of the steering controller, which deal with some special driving configurations and requirements: the Modified Bug Steering Fuzzy module, and the Final Orientation module. The fifth section introduces the three modules of the velocity controller: the Target Throttle Fuzzy module, the Cornering Throttle Fuzzy module, and the Collision Avoidance Throttle Fuzzy module. The sixth section proposes a tuning of the Target Throttle Fuzzy module to maintain a smooth velocity profile when the vehicle reaches its target position. The seventh section depicts two examples to show the operation of the two proposed fuzzy controllers, and comparison with the results of (Lee and Wang, 1994) are also included. The final section presents conclusions and recommendations for further work.

2. Vehicle Model

Since an automobile is a complex dynamic system, the details of its model are crucial to the accuracy of the simulation. The principal work in this field is (Wong, 1993) who developed the equations of motion for various types of vehicles. The model that is used in this paper depicts a two-axle, four-wheel vehicle, which is commonly known as the "bicycle" model since it uses only two wheels to represent a four-wheel vehicle; it neglects the lateral variations in the tire-road interface forces. It is further modified according to the suggestions of (Wheeler & Shoureshi, 1994) whose vehicle model was made more realistic by limiting the total tire force vector. They also derived expressions for calculating the longitudinal braking and accelerating forces. Figure 1 shows free body diagram of the vehicle. The equations of motion are

$$\begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix} \begin{Bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\theta} \end{Bmatrix} + \begin{Bmatrix} -m\dot{y}\dot{\theta} \\ m\dot{x}\dot{\theta} \\ 0 \end{Bmatrix} = \begin{Bmatrix} F_{xf}\cos\delta_f + F_{xr} - F_{yf}\sin\delta_f \\ F_{yr} + F_{yf}\cos\delta_f + F_{xf}\sin\delta_f \\ L_1 F_{yf}\cos\delta_f - L_2 F_{yr} + L_1 F_{xf}\sin\delta_f \end{Bmatrix} \quad (1)$$
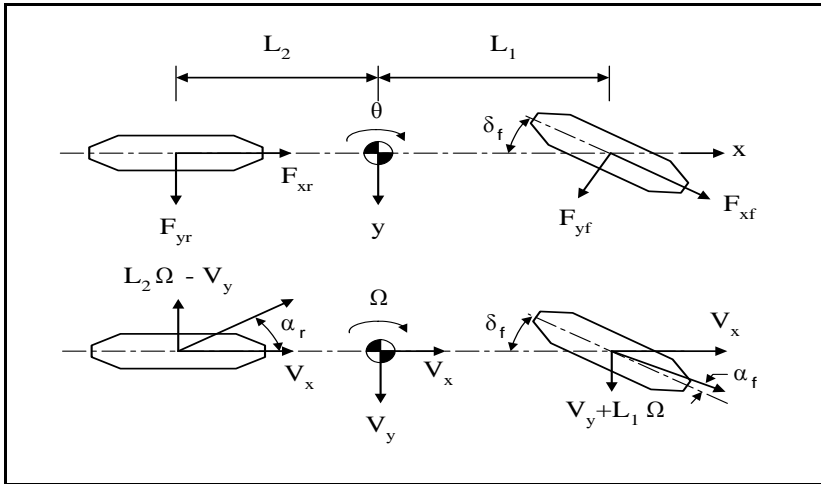

Fig. 1. Free Body Diagram of Vehicle Model.

The tires, which are modeled as nonlinear springs, are described as

$$\overline{F}_f = \frac{\tilde{F}_f}{\left\| \tilde{F}_f \right\|} \min\left(\left\| \tilde{F}_f \right\|, F_{f\max} \right), \quad (2)$$

$$\overline{F}_r = \frac{\tilde{F}_r}{\left\| \tilde{F}_r \right\|} \min\left(\left\| \tilde{F}_r \right\|, F_{r\max} \right). \quad (3)$$

The above forces can be resolved into x and y components to describe the motion. The lateral tire forces, F_{yf} and F_{yr}, are functions of each tire slip angle α and the cornering stiffness C_α. The lateral tire forces can be calculated using the following expressions:

$$\tilde{F}_{yf} = 2C_{\alpha f}\left(\delta_f - \tan^{-1}\left(\frac{L_1\dot{\theta} + \dot{y}}{\dot{x}} \right) \right), \quad (4)$$

$$\tilde{F}_{yr} = 2C_{\alpha r}\left(\tan^{-1}\left(\frac{L_2\dot{\theta} - \dot{y}}{\dot{x}} \right) \right) \quad (5)$$

The axial tire forces, F_{xf} and F_{xr}, are dependent on the angle of the gas pedal, δ_{gb}. This model uses the convention that a positive gas pedal angle represents the driver pushing on the gas pedal, and a negative gas pedal angle represents the driver pressing on the brake pedal. As such, there are two sets of axial tire force equations. For $\delta_{gb}<0$,

$$\tilde{F}_{xf} = 0.7K_b\delta_{gb}, \quad (6)$$

$$\tilde{F}_{xr} = 0.3 K_b \delta_{gb} \cdot \qquad (7)$$

The power train is assumed to correspond to a rear wheel drive. Thus, for $\delta_{gb}>0$,

$$\tilde{F}_{xf} = 0 , \qquad (8)$$

$$\tau_g \dot{\tilde{F}}_{xr} = K_g \delta_{gb} - \tilde{F}_{xr} \qquad (9)$$

Many of these parameters, which are particular to each vehicle, are usually determined experimentally. For this model, the parameters for a typical sports utility vehicle are used. Some of these parameters were found in (Byrne & Abdallah, 1995). The parameters K_g, K_b, and τ_g, are determined by varying their values and comparing the performance of the model during starting from rest, various peak velocities, and braking with experimental data.

It is assumed that the target has a beacon to guide the vehicle. It is also assumed that the vehicle is equipped with a proximity sensing system to determine the distance and direction of obstacles. This sensing system may be a single sensor mounted on a rotating platform on the front of the vehicle or a battery of sensors arrayed around the front of the vehicle and pointed along regular intervals. The configuration of the buffer zone created by the sensors is shown in Figure 2. To simplify obstacle detection, the sensor outputs the minimum measured distance to the nearest obstacle d_o and the direction of d_o, $\Delta\phi_o$. While d_o is not guaranteed to be the minimum distance to obstacle, sampling many points at a high frequency will increase the accuracy of the proposed technique.
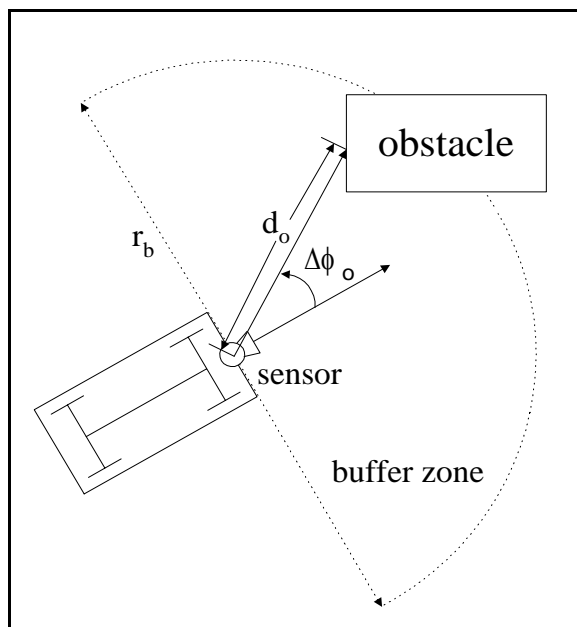

Fig. 2. Sensing Field Configuration.

3. Basic Steering Fuzzy Controller

In an effort to incorporate human knowledge and experience most efficiently in the design of the controller, the driving is divided into several tasks and a fuzzy controller was designed for each task. The basic driving tasks are to drive the vehicle toward the target and to avoid collision with obstacles. Two fuzzy modules, Target Steering Fuzzy module and Collision Avoidance Steering Fuzzy module, are designed to fulfill those two tasks. Two additional modules are discussed in the next section that meet special configurations or requirements.

Figure 3 shows a schematic of the inputs and outputs of Target Steering Fuzzy module and Collision Avoidance Steering Fuzzy module. The total steering angle is a summation of the outputs of two fuzzy modules. When the vehicle is near an obstacle, the output of Collision Avoidance Steering Fuzzy module is given a higher weight than that of Target Steering Fuzzy module, so that Collision Avoidance Steering Fuzzy module will be able to significantly affect the behavior of the vehicle in a short period of time.
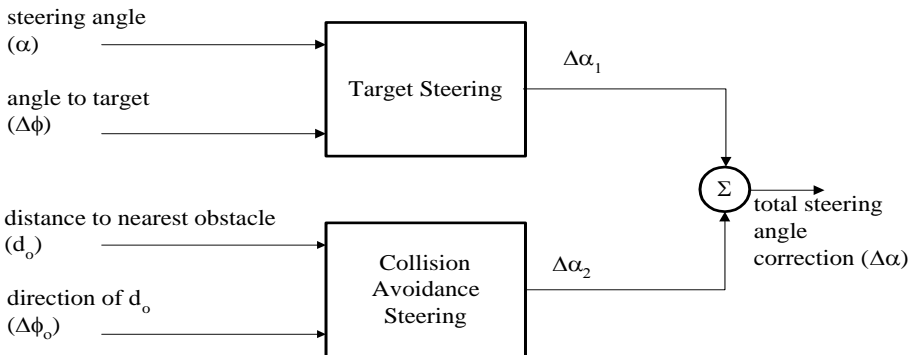

Fig. 3. Target Steering Fuzzy and Collision Avoidance Steering Fuzzy modules.

The membership functions in all of the fuzzy modules in this paper are either *sigmoid* or *product of sigmoid* types. The *sigmoid* membership function is defined as

$$f(x) = \frac{1}{1 + e^{-a(x-c)}} \ . \tag{10}$$

The *product of sigmoid* membership function is the result of multiplying two sigmoid membership functions together,

$$\mu(x) = \frac{1}{1 + e^{-a_L(x-c_L)}} \ \frac{1}{1 + e^{a_R(x-c_R)}} \tag{11}$$

Membership functions and the rules of the controller's modules that are used throughout the paper are based on common sense and observation of drivers' behavior and description of the variables used. These functions and rules are the result of tuning by running the simulation through various environments and by altering the initial vehicle position and orientation, the target vehicle position and orientation, the number and configuration of static obstacles, and the path and speed of dynamic obstacles.

3.1 Target Steering Fuzzy Module

The objective of this module is to steer the vehicle toward a target from its current location. The inputs to this module are the steering angle α and the angle to the target $\Delta\phi$, Figure 4. The output of this module is the change of steering angle $\Delta\alpha_1$. This module has the following goals:

i. If the target is on the right side of the vehicle, it should turn to this direction. The bigger the angle to the target, the sharper the change of steering angle should be.

ii. The change of steering angle depends on the current value of the steering angle.
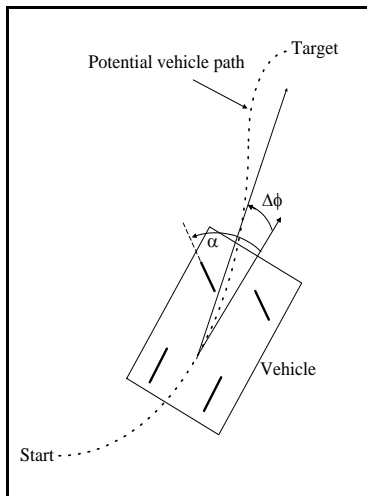
Fig. 4. Input Variables of the Target Steering Module.

Five membership functions: Negative Big (NB), Negative Medium (NM), Zero (Z), Positive Medium (PM) and Positive Big (PB), as shown in Figures 5 and 6, are used to describe the inputs and output of this fuzzy module. The same membership functions were used for α and $\Delta\phi$ for simplicity. The inner bounds of the PM and NM are all at zero due to the gradual tuning that was performed on them. This tuning also accounts for the lack of a "small" membership set, which was eliminated during the fine tuning phase.
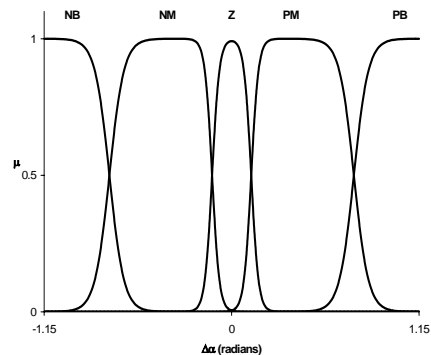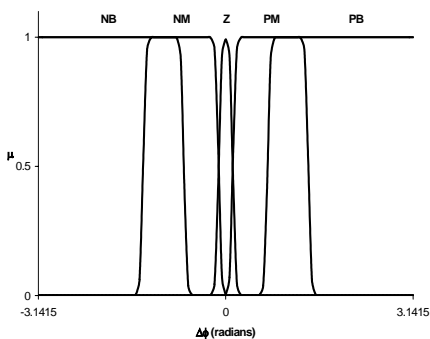
Fig. 5. Membership Functions for α and $\Delta\phi$.   Fig. 6. Membership Functions for $\Delta\alpha_1$.

The rationale behind several of the rules of this module is presented here. Larger correction is usually not used to allow for the time delay in the system.

- If (α is Z) and ($\Delta\phi$ is NB) then ($\Delta\alpha_1$ is NM): If the vehicle is moving along a straight path and the target is to the right of the vehicle, then the correction to the steering angle should be of medium magnitude and to the right.
- If (α is PB) and ($\Delta\phi$ is Z) then ($\Delta\alpha_1$ is NM): If the vehicle is turning to the left and the target is straight ahead, then the correction to the steering angle should be of medium magnitude and to the right.
- If (α is Z) and ($\Delta\phi$ is Z) then ($\Delta\alpha_1$ is Z): If the current steering angle is zero and the target is straight in front of vehicle, then no correction to the steering angle is necessary.

The full rule base of this module is given in Table 1.

$\alpha\Rightarrow$ $\Delta\phi\Downarrow$	NB	NM	Z	PM	PB
NB	Z	Z	NM	NB	NB
NM	Z	Z	NM	NM	NM
Z	PM	Z	Z	Z	NM
PM	PB	PB	PM	Z	Z
PB	PB	PB	PM	Z	Z

Table 1. Rules for Target Steering Fuzzy Module

3.2 Collision Avoidance Steering Fuzzy Module

The objective of this module is to steer the vehicle away from obstacles, both static and dynamic. It is assumed that the vehicle is equipped with a sensing system that can determine the distance and direction of obstacles. The sensing system may be a single sensor mounted on a rotating platform on the front of the vehicle or may be a battery of sensors arrayed around the vehicle's front section along regular intervals. The buffer zone radius will be denoted by r_b, and is assigned a value of twenty meters. The configuration of the buffer zone is shown in Figure 2. Sensors input only two bits of information to the controller to simplify the analysis. The inputs are the distance and angle to the nearest obstacle, d_o and $\Delta\phi_o$, respectively. As Figure 2 illustrates, the sensor may not return information about the closest point and the corresponding angle since the proposed sensing system used is not continuous. The sensor takes n_s samples over its 180-degree sweep. The output of the module is the steering correction $\Delta\alpha_2$. Since the locations of the obstacles are not known before the vehicle starts traversing the environment, the module uses a right-hand turning rule whenever it confronts an obstacle. The two basic goals of this module are as follows:

i. The closer the vehicle is to the obstacle, the more extreme the evasive maneuver is, i.e., the larger the steering correction.
ii. The direction of the obstacle with respect to the front of the vehicle determines the magnitude and direction of the steering correction.

Three membership functions, Big (B), Small (S) and Zero (Z) are used to describe the first input variable, d_0 as shown in Figure 7. Five membership functions, Negative Medium (NM), Negative Small (NS), Zero (Z), Positive Small (PS) and Positive Medium (PM) are used to describe the second input variable, $\Delta\phi_0$ and the output variable, $\Delta\alpha_2$ as shown in Figure 8. The membership functions of $\Delta\alpha_2$ in this module have the same shape and relative

size as those of $\varDelta\alpha_1$ in the previous module, Figure 9. The only difference is that all of the membership functions have been scaled down.
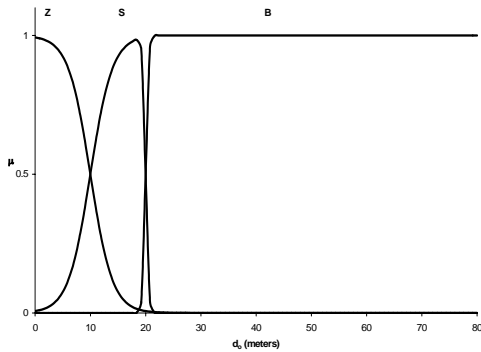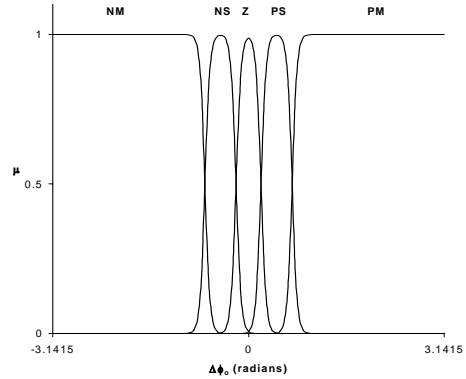

Fig. 7. Membership Functions for d_0.

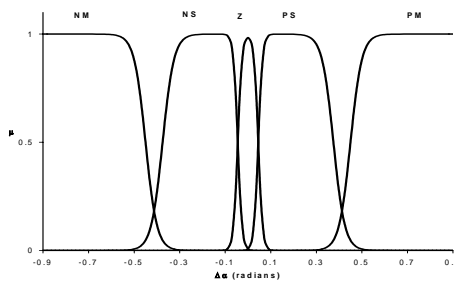

Fig. 8. Membership Functions for $\varDelta\phi_0$.

Fig. 9. Membership Functions for Output Variable $\varDelta\alpha_2$.

At this point, it would be useful to look at a few of the rules to observe the relation between the rules and the goals of this fuzzy module:

- If (d_o is S) and ($\varDelta\phi_0$ is PS), then ($\varDelta\alpha_2$ is NM): If the obstacle is between ten and twenty meters away, and is between zero to sixty degrees off to the left of the longitudinal axis of the vehicle, a sharp steering correction to the right will be needed.
- If (d_o is B) then ($\varDelta\alpha_2$ is Z): If the obstacle is more than twenty meters away from the vehicle, then no steering correction is made, regardless of its direction.

The rules of this module are shown in Table 2.

$d_0 \Rightarrow$ $\varDelta\phi_0 \Downarrow$	Z	S	B
NM	PM	PM	Z
NS	PM	PS	Z
Z	NM	NM	Z
PS	NM	NM	Z
PM	NM	NM	Z

Table 2. Rules for Collision Avoidance Steering Fuzzy Module.

4. Extended Steering Fuzzy Controller

The basic steering fuzzy controller described in the previous section works well in most driving conditions. However, it may get stuck in a loop and never reach the target under certain special configurations. It may also fail to determine that the target is unreachable. In some applications, the vehicle is required to park at the target position with a specific orientation. Thus, two modules are added in the steering fuzzy controller: Modified Bug Steering Fuzzy module and Final Orientation module. The first module steers the vehicle within a maze and the other helps steer the vehicle toward its target position at a desired orientation.

4.1 Modified Bug Steering Fuzzy Module

As a backup to the primary steering fuzzy controller, which usually regulates obstacle avoidance steering, the Bug module is implemented. This module is not designed to emulate human behavior, but rather to reach the target when the steering fuzzy module is not able to.

The Bug module is based on the Bug2 algorithm proposed by (Lumelsky & Stepanov ,1987, Lumelsky & Skewis, 1988, Lumelsky, 1991) whose research focused on the application of maze theory to the path planning of a "point automaton". His research concluded that Bug2 had "unbounded worst-case performance", i.e. in very rare cases, Bug2 can still drive the vehicle in an unending loop. (Sauerberger & Trabia, 1996) proposed a modified form of this algorithm for autonomous omnidirectional vehicles. Their algorithm, which produces shorter paths in many cases, triggered the Bug algorithm when the orientation of the vehicle was more than 360 degrees away, in either direction, from the angle of the ST line, Figure 10. This can be expressed as the following:

$$\left| \theta_v - \theta_{ST} \right| > 2\pi$$

(12)

For example, if the ST line was at *45* degrees, the vehicle would have to traverse a closed loop, and have an orientation of *405* degrees, before the Bug2 algorithm was activated.

The controller presented here presents another modification of the Bug2 algorithm. The modified algorithm is activated once the trigger condition is met. At this stage, the algorithm performs the following tasks:

1) Initially, the algorithm guides the vehicle straight toward the target. If the finishing criteria (discussed below) are satisfied, the vehicle turns off the Bug2 algorithm and goes straight to the target. If the vehicle gets to within r_o meters of an obstacle, the vehicle will turn to the right, putting the obstacle on the left of the vehicle.

2) The vehicle proceeds to follow the boundary of the obstacle, always keeping the obstacle on its left until one of the following conditions is met:
    a) If the finishing criteria are satisfied, the vehicle goes straight to the target.
    b) When the vehicle approaches the ST line, it treats it as an obstacle, turning to the right and keeping the line on its left if it moves closer to the target. Therefore, the algorithm causes the vehicle to stay on one side of the ST line, which will aid the vehicle in reaching the target quickly.
    c) If the vehicle approaches the ST line and following it will drive the vehicle away from the target, it behaves as if the line is not an obstacle and crosses it.

The following finishing criteria must be simultaneously satisfied for the controller to allow the vehicle to go to the target:

•   The vehicle must be within a distance r_b to the target. Larger distances may indicate that an obstacle lies between the vehicle and the target and may be undetected by the vehicle's sensors.

- The vehicle must have a clear "line of sight" to the target. This condition is implemented by comparing the magnitude of the angle to the target to the magnitude of the angle to the nearest obstacle.

Schematics of the paths generated by the Bug module are shown in the two examples of Figure 10 and Figure 11.
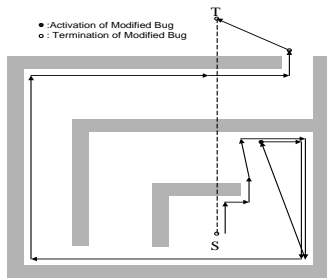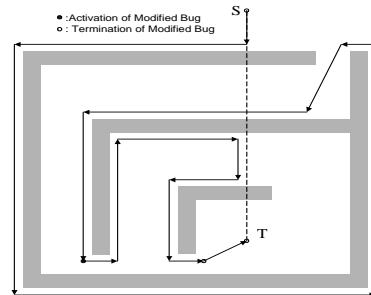

Fig. 10. Path 1 by Bug Steering Module.

Fig. 11. Path 2 by Bug Steering Module.

The Bug Steering fuzzy module is designed to accommodate four-wheel vehicles, which cannot make the zero-radius turns that omnidirectional vehicles can. Thus, the module begins the appropriate steering adjustments ahead of time to allow the vehicle to make the necessary turns without colliding with obstacles or crossing the ST line. The inputs to the Bug module are the distance to the obstacle d_o, and the angle to the obstacle $\Delta\phi_0$. The output of this module is the correction in the steering angle $\Delta\alpha_3$.

Four membership functions, Big (B), Medium (M), Small (S) and Zero (Z) are used to describe the first input variable, d_0 as shown in Figure 12. Seven membership functions, Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (Z), Positive Small (PS), Positive Medium (PM) and Positive Big (PB) are used to describe the second input variable, $\Delta\phi_0$, as shown in Figure 13. Five membership functions, Negative Medium (NM), Negative Small (NS), Zero (Z), Positive Small (PS) and Positive Medium (PM) are used to describe the output variable, $\Delta\alpha_3$, as shown in Figure 14. Note that the membership functions in this module are different from those of Figure 7 through Figure 9. This modification was necessary to improve the module response.
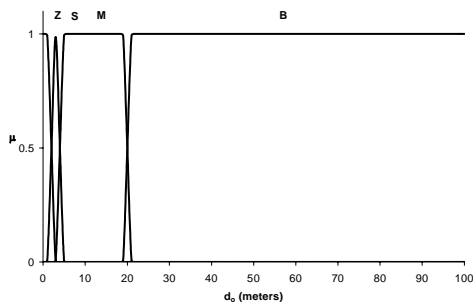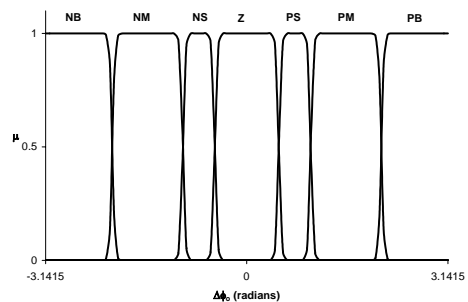

Fig. 12. Membership Functions for d_o.

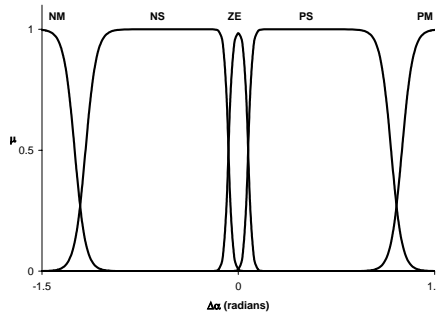Fig. 13. Membership Functions for $\Delta\phi_0$.

Fig. 14. Membership Functions for Output Variable $\Delta\alpha_3$.

As the rules in this module allow the vehicle to track obstacles, they are not very different from those of the Collision Avoidance Steering Fuzzy module. The rules of this module are shown in Table 3.

$d_0 \Rightarrow$ $\Delta\phi_0 \Downarrow$	Z	S	M	B
NB	PM	PM	PM	Z
NM	PM	PM	PM	Z
NS	PM	PM	PM	Z
Z	NM	NM	NS	Z
PS	NM	NS	NS	Z
PM	NS	Z	PS	Z
PB	PS	PS	PS	Z

Table 3. Rules for Modified Bug Steering Fuzzy Module.

4.2 Final Orientation Module

Orienting the vehicle in a particular direction at the target point (similar to parking a car in a parking lot) presents a challenging problem. This paper presents a simple solution for it by adding the fourth module in the steering controller. The final orientation consists of setting up "virtual" targets that are based on the desired final orientation. The controller guides the vehicle to the correct orientation by passing the vehicle through the virtual targets, which gradually orients the vehicle in the right direction. The coordinates of these target points are shown in Table 4. X_t and Y_t are the actual target coordinates and θ_t is the desired final orientation. The final orientation module is independent of the other three fuzzy steering modules. The steering fuzzy modules continue to control the vehicle as they otherwise would while the final orientation algorithm is operational.

Target	X coordinate (m)	Y coordinate (m)
T_{v1}	$X_t + 20\cos(\theta_t - \pi)$	$Y_t + 20\sin(\theta_t - \pi)$
T_{v2}	$X_t + 10\cos(\theta_t - \pi)$	$Y_t + 10\sin(\theta_t - \pi)$
T	X_t	Y_t

Table 4. Target Coordinates and Tracking Order.

Figure 15 shows a vehicle approaching the target. The final orientation algorithm first creates T_{v1} and drives the vehicle toward it. The algorithm then creates T_{v2}, and again drives the vehicle toward it. Finally, the algorithm allows the vehicle to detect the actual target, and the vehicle goes toward it.
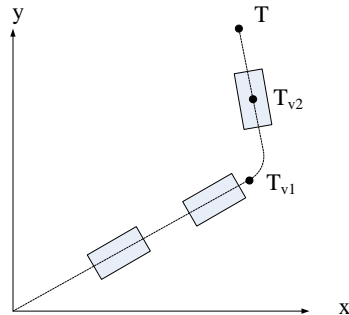
Fig. 15. Vehicle Approaching Target from a Specified Direction.

5. Velocity Fuzzy Control

To use human knowledge and experience efficiently in controlling the velocity of the vehicle, the problem is separated into several tasks. A fuzzy controller module is designed for each task. These tasks are target throttle, cornering throttle, and collision avoidance throttle. Figure 16 shows a schematic of the inputs and outputs of the three fuzzy modules to achieve these tasks. The total throttle/brake angle is the summation of the outputs of these modules.
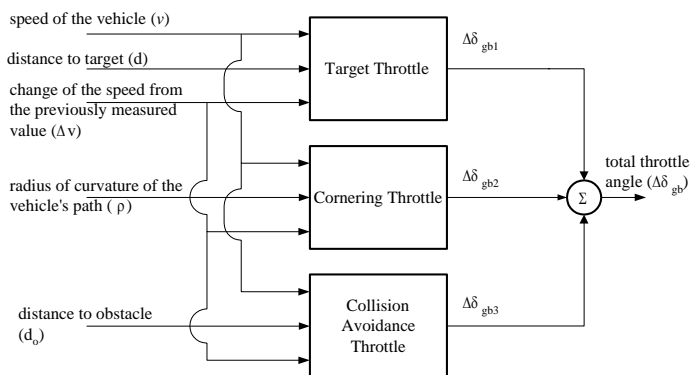
Fig. 16. Schematic Diagram of the Velocity Fuzzy Controller Modules.

5.1 Target Throttle Fuzzy Module

The objective of this module is to speed up the vehicle to reach the target, to slow it down when approaching the target and to stop it at the target position. The inputs to this module are the velocity of the vehicle v, distance to target d, and the change of velocity from the previously measured value Δv. The module has one output, which is the change in the gas pedal/brake angle $\Delta\delta_{gb1}$.

Four membership functions, Big (B), Medium (M), Small (S) and Zero (Z) are used to describe the first and the second input variables, v and d as shown in Figure 17 and 18 respectively. Note that the upper bound on the B membership set in Figure 17 is *twenty-five* meters per second, which indicates that the maximum velocity of the vehicle can be up to *ninety* kilometers per hour. Since the controller is designed for a totally autonomous vehicle, it is conservative; thus, the reduction of velocity due to the distance to the target, d, begins early to avoid the potential for collision. Three membership functions, Negative (N), Zero (Z), Positive (P) are used to describe the third input variable, Δv, as shown in Figure 19. These membership functions are selected to only indicate whether the velocity is decreasing, constant, or increasing. Finally, six membership functions, Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (Z), Positive Small (PS), and Positive Big (PB) are used to describe the output variable, $\Delta\delta_{gb1}$, as shown in Figure 20. Note that the membership functions for $\Delta\delta_{gb1}$ are not symmetric around zero since the vehicle should stop in less time than that it takes to accelerate the vehicle. The figure also shows that there are more negative functions than positive ones, which allows finer control over braking. The following is a sample of the rules of this module,

- If (v is B), (d is Z) and (Δv is P), then ($\Delta\delta_{gb1}$ is NB): The rule states that if the velocity is *big*, the distance to target is *zero,* and the change of velocity is *positive*, the controller should supply a *big* brake angle.

- If (v is Z), (d is B) and (Δv is N), then ($\Delta\delta_{gb1}$ is PB): This rule states that if the velocity is *zero*, the distance to target is *big,* and the change of velocity is *negative*, the controller should supply a *big* gas pedal angle.

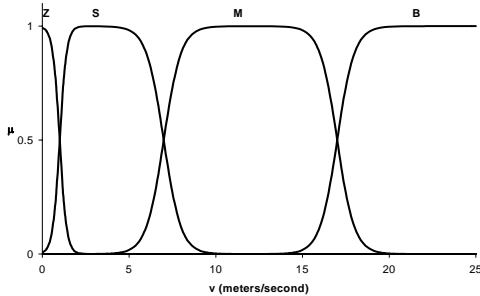The rules of this module are listed in Table 5.
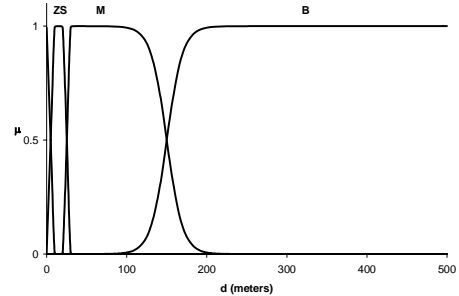

Fig. 17. Membership Functions for v.

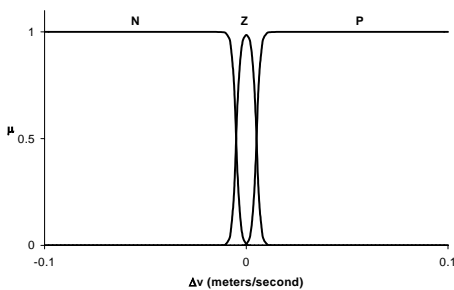Fig. 18. Membership Functions for d.

Fig. 19. Membership Functions for Δv.
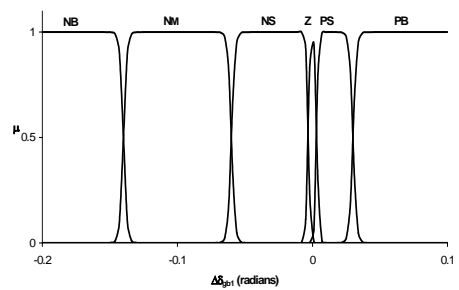
Fig. 20. Membership Functions for $\Delta\delta_{gb1.}$

$v \Downarrow$	$d \Rightarrow$ $\Delta v \Downarrow$	Z	S	M	B
Z	N	PS	PS	PB	PB
	Z	PS	PS	PB	PB
	P	NS	Z	PB	PB
S	N	NS	Z	PS	PB
	Z	NS	NS	Z	PB
	P	NS	NM	NS	PB
M	N	NB	NS	PS	PS
	Z	NB	NM	Z	PS
	P	NB	NB	NM	Z
B	N	NS	NS	Z	PS
	Z	NB	NS	NS	Z
	P	NB	NB	NS	NB

Table 5. Rules for Target Throttle Fuzzy Module.

5.2. Cornering Throttle Fuzzy Module

The objective of this module is to slow down the vehicle when it is turning to ensure its stability. The inputs to this module are the velocity of the vehicle v, the radius of curvature of the vehicle's path ρ (Figure 21), and the change of velocity from the previous measured value Δv. The module has one output, which is the change in the gas pedal/brake angle $\Delta\delta_{gb2}$.

The membership functions of v, Δv, and $\Delta\delta_{gb2}$ are the same as those used in the target throttle module. However, $\Delta\delta_{gb2}$ uses only NS and Z membership functions since cornering module primarily reduces the velocity of the vehicle at sharp corners. Three membership functions, Z, S, B are used to describe the radius of curvature ρ, as shown in Figure 22. Note that these membership functions are selected to activate this module at sharp corners only.
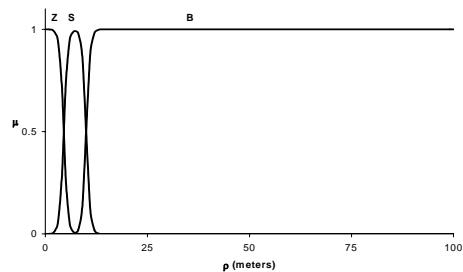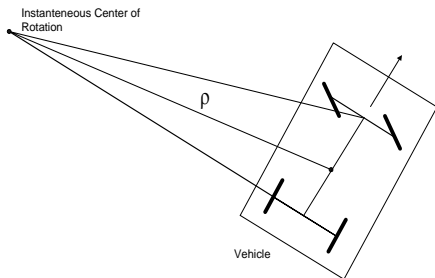

Fig. 21. Instantaneous Radius of Curvature. Fig. 22. Membership Functions for ρ.

The following is a sample of the rules of this module,

- If $(v\ is\ B)$, $(\rho\ is\ Z)$, $and\ (\Delta v\ is\ P)$, $then\ (\Delta\delta_{gb2}\ is\ NS)$: The first rule states that if the velocity is big, the radius of curvature is zero, and the change of velocity is positive, the controller should supply a small brake angle.
- If $(v\ is\ S)$, $(\rho\ is\ B)$, $and\ (\Delta v\ is\ Z)$, $then\ (\Delta\delta_{gb2}\ is\ Z)$: This rule states that if the velocity is small, the radius of curvature is big and the change of velocity is zero, the controller should do nothing.

The rules of this module are shown in Table 6.

$v \Downarrow$	$\rho \Rightarrow$ $\Delta v \Downarrow$	Z	S	B
Z	N	Z	Z	Z
	Z	Z	Z	Z
	P	Z	Z	Z
S	N	Z	Z	Z
	Z	Z	Z	Z
	P	NS	Z	Z
M	N	Z	Z	Z
	Z	NS	Z	Z
	P	NS	Z	Z
B	N	NS	Z	Z
	Z	NS	Z	Z
	P	NS	NS	Z

Table 6. Rules for Cornering Throttle Fuzzy Module.

5.3. Collision Avoidance Throttle Fuzzy Module

The goal of this module is to slow down the vehicle as it moves near obstacles. The inputs to this module are the velocity of the vehicle v, minimum measured distance to the nearest obstacle d_o, and the change of velocity from the previous measured value Δv. The output of the module is the change in the gas pedal/brake angle $\Delta \delta_{gb3}$.

The membership functions of v and Δv are the same as those used in the target throttle module. Figure 23 and Figure 24 show membership functions of d_o and $\Delta \delta_{gb3}$ respectively. Figure 23 shows that this modules is activated when the distance from the vehicle to the obstacle becomes less than twenty meters.

The membership functions for v and Δv in the previous module are used in this module. Three membership functions, Z, S, B (Figure 23) are used to describe d_o. This module uses five membership functions, NB, NS, Z, PS, PB (Figure 24) to describe $\Delta \delta_{gb.3}$ The minimum value of range of $\Delta \delta_{gb3}$ is significantly less than the minimum value of range of $\Delta \delta_{gb1}$ to indicate the need to brake the vehicle faster near obstacles.
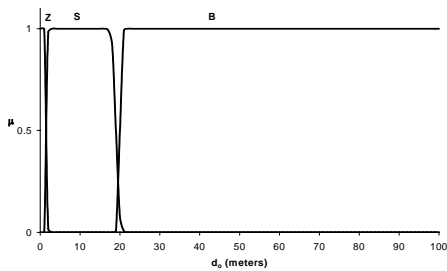

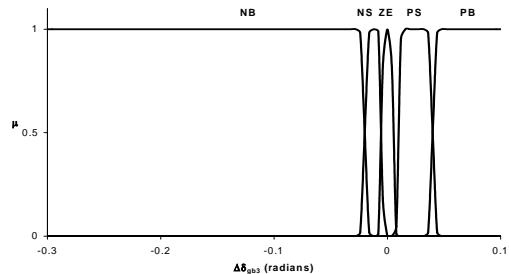Fig. 23. Membership Functions for Input Variable, d_o.

Fig. 24. Membership Functions for Output Variable, $\Delta \delta_{gb3}$.

The following is a sample of the rules of this module,

- If $(v\ is\ B)$, $(d_0\ is\ Z)$, $and\ (\Delta v\ is\ P)$, $then\ (\Delta\delta_{gb3}\ is\ NB)$: The rule states that if the velocity is *big*, the minimum measured distance to the nearest obstacle is *zero*, and the change of velocity is *positive*, the controller should supply a *big* brake angle.

- If $(v\ is\ S)$, $(d_0\ is\ S)$, $and\ (\Delta v\ is\ N)$, $then\ (\Delta\delta_{gb3}\ is\ Z)$: This rule states that if the velocity is *small*, the minimum measured distance to the nearest obstacle is *small*, and the change of velocity is *negative*, the controller should *do nothing*.

The full rule set of this module is shown in Table 7.

$v \Downarrow$	$d_o \Rightarrow$ $\Delta v \Downarrow$	Z	S	B
Z	N	PS	PS	Z
	Z	PS	PS	Z
	P	NS	Z	Z
S	N	PS	Z	Z
	Z	PS	Z	Z
	P	NS	NS	Z
M	N	NS	Z	Z
	Z	NB	NS	Z
	P	NB	NB	Z
B	N	NS	NS	Z
	Z	NB	NS	Z
	P	NB	NB	Z

Table 7. Rules for Cornering Throttle Fuzzy Module.

6. Tuning Target Throttle Fuzzy Module

Tests show that cornering throttle and obstacle avoidance modules performed adequately after little manual tuning. However, the Target Throttle Fuzzy module consistently produces an oscillatory vehicle velocity profile as the vehicle approaches the target. Manual tuning of the membership functions and rules fails to improve the performance of this module. While the rules of this module generally fit within the driving patterns of most drivers, this oscillatory behavior can be easily explained by considering that two objectives of this module compete with each other near the target:

    i.   Slowing down the vehicle as the vehicle approaches the target.
    ii.   Maintaining a non-zero velocity to ensure that the vehicle does not stop before reaching the target.

Usually drivers solve this problem by pressing the gas or brake pedals lightly when they approach their final target. Narrowing the ranges of the membership functions of the output variable of a fuzzy controller may simulate this behavior.

The design presented here proposes an adaptive tuning method to smooth the velocity of the vehicle as it approaches the target. Since the tuning is to be performed in real time, it is important to avoid intensive computations. Tuning the controller is based on the following decisions:

i.   The previous fuzzy rules of Section 5.1 are left unchanged.
ii. The membership functions of the input variables are considered reasonable.
iii. The membership functions of the output variable $\Delta\delta_{gb1}$, Figure 20, are used as an initial guess.
iv. The Target Throttle Fuzzy module remains unchanged if the vehicle is far away from the target. This controller will be from now on labeled the *cruising throttle module*. Tuning becomes active only when the vehicle approaches the target. The tuned controller will be labeled the *slowing-down throttle module*.

The philosophy of tuning the membership functions can be still described using linguistic terms. The algorithm takes one of these actions:

Case 1: If Δv is *positive* (accelerating), and $\Delta\delta_{gb}$ at the previous sample is also *positive*, attempt to decrease the velocity by moving the membership function of *positive small (PS)* of $\Delta\delta_{gb1}$ closer to zero.

Case 2: If Δv is *negative* (decelerating), and $\Delta\delta_{gb}$ at the previous sample is also *negative*, attempt to increase the velocity by moving the membership function of *negative small (NS)* of $\Delta\delta_{gb1}$ closer to zero.

Case 3: If Δv and $\Delta\delta_{gb}$ at the previous sample have different signs, it indicates that the controller is trying to stabilize the velocity. In this case, all the membership functions of $\Delta\delta_{gb1}$ should remain unchanged.

The product of sigmoid membership functions, given by Equation (11), offers the advantage of changing only one side of the membership function while keeping the other unchanged.

To maintain smooth output of the controller, moving the right side of PS membership function requires a corresponding movement of the left side of PB membership function. Similarly, moving the left side of NS membership function requires a corresponding movement of the right side of NM membership function. That is, *boundaries* of membership functions must be moved such that there are no gaps created between membership functions. As such, the proposed scheme maintains the initial amount of overlap (or gap) between any adjacent membership functions. The controller incorporates the tuning system, Figure 25, which has these components:

- *Trigger*: If the distance is small, the tuning scheme is triggered. In this paper, a crisp trigger of comparing the distance to the target with D_{small} is used.
- *Cruising throttle module*: The module is the controller of Section 6.1.
- *Tuner*: The trainer tunes the membership functions of the output of target throttle module if the trigger is fired. Its decision is based on the three cases discussed above. The new membership functions are

$$\mu_{PB}(i) = \frac{1}{1 + e^{-a_{PBL}(x-\phi(i)\,c_{PBL}(i-1))}} \quad , \tag{13}$$

$$\mu_{PS}(i) = \frac{1}{1 + e^{-a_{PSL}(x-c_{PSL})}} \frac{1}{1 + e^{a_{PSR}(x-\phi(i)\,c_{PSR}(i-1))}} \quad , \tag{14}$$

$$\mu_{NS}(i) = \frac{1}{1 + e^{-a_{NSL}(x-\phi(i)\,c_{NSL}(i-1))}} \frac{1}{1 + e^{a_{NSR}(x-c_{NSR})}} \quad , \tag{15}$$

$$\mu_{NM}(i) = \frac{1}{1 + e^{-a_{NML}(x-c_{NML})}} \frac{1}{1 + e^{a_{NMR}(x-\phi(i)c_{NMR}(i-1))}} \tag{16}$$

When $|\Delta v|$ is significantly close to zero, the original membership functions should be used. Therefore, the value of the scaling factor $\phi(i)$ is equal to one. As the value of $|\Delta v|$ increases, the original membership functions are modified by reducing the value of $\phi(i)$. $\phi(i)$ is described using the following equation:

$$\begin{aligned} \varphi(i) &= 0.25 - 300\left(\left|\Delta v(i)\right| - 0.0025\right) \quad &\text{if } \left|\Delta v(i)\right| < 0.0025 \\ \varphi(i) &= 0.25 \quad &\text{if } \left|\Delta v(i)\right| \geq 0.0025 \end{aligned} \tag{17}$$

Product of sigmoid membership functions require that the center of the right side be of higher value than that of the left side. To maintain this characteristic, the minimum value of $\phi(i)$ is limited to 0.25. The *Slowing-down module* is similar to the *Cruising throttle module*. The only exception is that it uses the tuned membership functions produced by the *Tuner* for the output variable.

The parameter PI (i.e., the *Performance Index*) is used to assess the performance of the controller with and without the tuning scheme:

$$PI = \sum_{i=k} \Delta v(i)^2 \qquad t(k) = t(D_{small}) \tag{18}$$
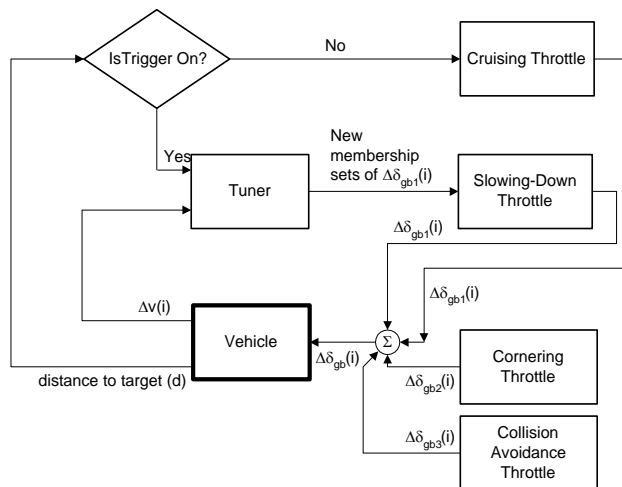

Fig. 25. Schematic of Target Throttle Tuning System.

7. Examples

The seven modules of the two proposed controllers are used to guide the vehicle around obstacles and reach the target in several examples to check the validity of the ideas proposed in this paper. In each example, the vehicle and the obstacle configurations are chosen to display certain characteristics of the controller or to contrast various modules of it.

7.1 Velocity Fuzzy Control Example

The following example, shown in Figure 26, demonstrates the abilities of the velocity fuzzy controller. The starting position and orientation are (0.0, 0.0) meters and zero degree respectively. The target position and orientation are (100.0, 60.0) meters and ninety degrees respectively. The buffer zone radius will be denoted by r_b, and is generally assigned a value of 20 meters. The sensors are sampled every 0.1 seconds while the time step for the simulation is 0.01 second. The example has both static obstacles and a dynamic obstacle that moves at a constant velocity. D_{small} is equal to 40 meters. The markers of Figure 26 correspond to $t_0 = 0$ seconds, $t_1 = 10$ seconds, $t_2 = 15$ seconds, $t_3 = 20$ seconds, and $t_4 = 45$ seconds. t_5 is equal to 99.5 seconds using the original target throttle module and 91.9 seconds using the target throttle tuning system. Figure 27 shows the velocity versus time of the vehicle at both cases. If no tuning is used, the velocity of the vehicle experiences an oscillatory pattern until it reaches the target. If the proposed tuning scheme is used, the velocity profile becomes smooth. The total path traversal time is also reduced. PI is equal to 0.0068 when the tuned controller is used, while PI is equal to 0.0566 when using the original cruising controller.

Figures 28 through 30 show the output of the modules, $\Delta\delta_{gb1}$ (with tuning), $\Delta\delta_{gb2}$, and $\Delta\delta_{gb3}$, respectively, when tuning is used. The target throttle controller (cruising controller) initially accelerates the vehicle. This controller later reduces its input to keep the velocity of the vehicle bounded. In the meantime, the cornering throttle controller attempts to slow the vehicle as it turns around the static obstacle. The obstacle avoidance controller has similar output at this stage. The effect of these controllers at t_1 becomes apparent as the vehicle starts to slow down. This trend continues until t_4 as the vehicle turns and tries to avoid both obstacles. The contribution of $\Delta\delta_{gb1}$ diminishes at this stage as this module slows the vehicle down while it moves closer to the target. The value of $\Delta\delta_{gb2}$ also approaches zero at this stage since the turning radii of the vehicle are reasonably larger. However, the obstacle avoidance controller is also active at this stage. This phase continues until the vehicle leaves the obstacles behind. The tuning trigger is activated directly after that point at around 46 seconds. The effect is clear in Figure 28. The reduction of $\Delta\delta_{gb1}$ is counteracted by the increase in $\Delta\delta_{gb3}$, which increases as the effects of the obstacles diminish. Eventually, the slowing-down module, which is the only active one at the end of the motion, moves the vehicle toward the target.
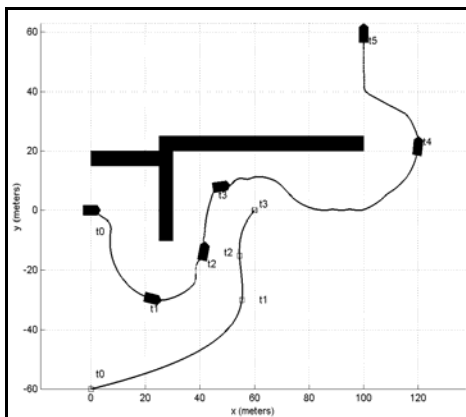

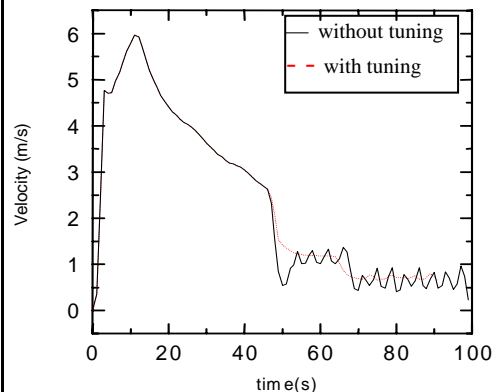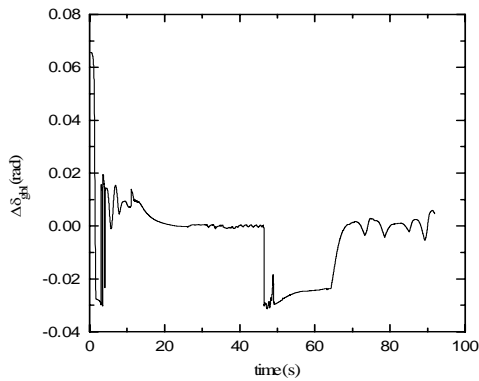Fig. 26. Trajectory of Velocity Fuzzy Controller.   Fig. 27. Velocity Profiles.

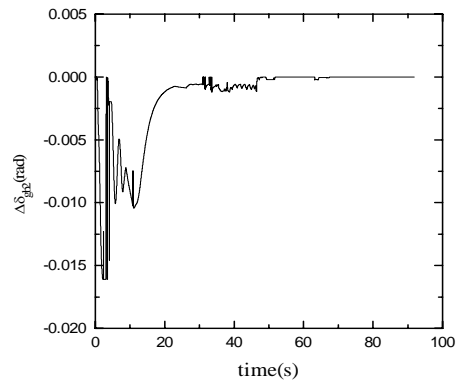Fig. 28. Output of $\Delta\delta_{gb1}$.

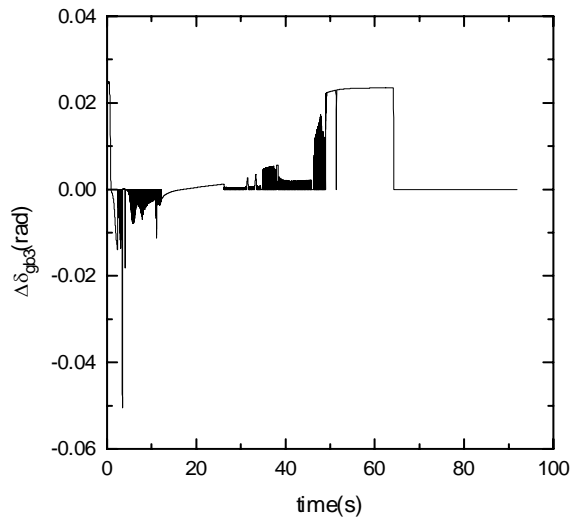Fig. 29. Output of $\Delta\delta_{gb2}$.

Fig. 30. Output of the Collision Avoidance Throttle Module, $\Delta\delta_{gb3}$.

7.2. Bug Steering Module Example

The conditions for the simulation are an initial position of (0.0, 0.0) meters, an initial orientation of zero degrees, a final position of (100.0, 100.0) meters and a final orientation of zero degrees. The results for this example are shown in Figure 31. The times depicted in the figures are t_0 = 0 seconds, t_1 = 15 seconds, t_2 = 25 seconds, t_3 = 75 seconds, t_4 = 140 seconds, and t_5 = 145 seconds, t_6 = 170 seconds, t_7 = 198 seconds, and t_8 = 217.8 seconds.

Initially, when there are no obstacles in its vicinity, the vehicle attempts to go straight to the target. Once the vehicle gets closer to the obstacle, it goes to the right, attempting to avoid it. When the vehicle senses an opening in the obstacle, it attempts to enter; however, as the opening is only 10 meters wide; thus, the controller will not allow the vehicle to enter since r_b is set to twenty meters. The vehicle then continues moving around the obstacles, attempting to find a clear path to the target. Once its orientation has deviated from the ST

angle by mire that 360 degrees, the Modified Bug Fuzzy module is triggered. This occurs between times t_4 and t_5. At that point, the vehicle attempts again to go straight to the target. Once it senses the obstacle, it turns right, keeping the obstacle at a distance of about three meters to the left of the vehicle. It follows the wall of the maze until about t_7. At this point, the vehicle is within twenty meters of the target and the vehicle has an unobstructed line of sight to the target. The final orientation controller is switched on here. Thus, the vehicle goes straight to the target while maintaining a course that results in a final orientation of zero degrees. The controller of (Lee and Wang, 1994)starts in the same direction as the proposed controller but fails to enter the maze as it is does not have a module for such a task, as shown in Figure 32.
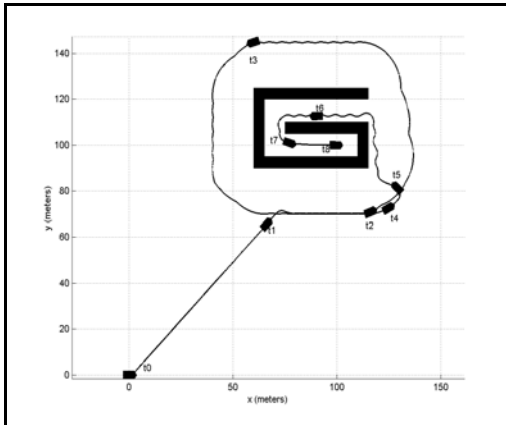

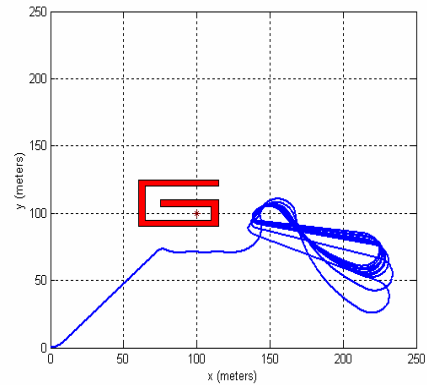Fig. 31. Results of Using the Proposed Controller in Maze Tracking.

Fig. 32. Results of Using the Controller of (Lee and Wang, 1994) in the Maze of Figure 31 .

8. Conclusions

This chapter presents a fuzzy logic control system for steering a two-axle vehicle. Two controllers are presented individually: the steering controller and the velocity controller. Each fuzzy controller is divided into several modules to represent the distributed way in which humans deal with different driving tasks. All of these modules are of the Mamdani-type and use sigmoid or product of sigmoid membership functions. The outputs of the various modules are added together to control the steering angle and the speed of the vehicle, respectively.

Two fuzzy modules in the steering controller are designed to meet the basic driving requirement: Target Steering Fuzzy module and Collision Avoidance Steering Fuzzy module. The first module steers the vehicle toward the target by monitoring the steering angle and the angle to the target. The objective of the second module is to avoid collisions with static and dynamic obstacles. Its inputs are the distance and angle to the nearest obstacle. When the vehicle is near an obstacle, the output of this module is given a higher weight than that of the first module, so that it will be able to significantly affect the behavior of the vehicle.

The Modified Bug Steering Fuzzy module is proposed as a backup for the Target Steering Fuzzy module in the event that it cannot guide the vehicle to the target. The criterion used

to turn it on consists of monitoring the vehicles global orientation and making sure that the vehicle's orientation does not change by more than 360 degrees. This module includes a start to target line as a fictitious obstacle. The vehicle follows this line if it encounters it, unless it drives away from the target, which reduces path length in many cases. The criteria used to turn this module off consist of making sure that i) the vehicle is within 20 meters of the target, and ii) the vehicle has an unobstructed line of sight to the target. Finally, the rules and membership functions of the Modified Bug Fuzzy module account for the physical properties of a two-axle vehicle by allowing it enough time and space to make the necessary turns.

A separate module in the steering controller adjusts the vehicle to the desired final orientation by introducing two intermediate target points. The module drives the vehicle through these two points and adjusts the vehicle's orientation accordingly.

The velocity controller is divided into several modules, each dealing with a separate objective, to mimic human behavior. These modules are:

i.   Target Throttle Fuzzy module: The objectives of the module are to start the vehicle moving from a complete stop, to speed it up, and to stop it when it gets sufficiently close to the target. The inputs to this module are the velocity of the vehicle, distance to target, and the change of velocity from the previously measured value. The module has one output, which is the change in the gas pedal/brake angle ($\Delta\delta_{gb1}$).

ii. Cornering Throttle Fuzzy module: The objective of this module is to reduce the speed of the vehicle when its turning radius decreases (i.e., the tighter the turn, the lower the velocity). The inputs to this module are the velocity of the vehicle, the radius of curvature of the vehicle's path, and the change of velocity from the previous measured value. The module has one output, which is the change in the gas pedal/brake angle ($\Delta\delta_{gb2}$).

iii. Collision Avoidance Throttle Fuzzy module: The objective of this module is the reduce the speed of the vehicle as it approaches an obstacle and increase the speed as the vehicle continues past the obstacle. The inputs to this module are the vehicle velocity, the minimum measured distance to the nearest obstacle, and the change of velocity from the previous measured value. The output of the module is the change in the gas pedal/brake angle ($\Delta\delta_{gb3}$).

The Cornering Throttle and Collision Avoidance Throttle Fuzzy modules performed satisfactorily after some manual tuning. The initial Target Throttle Fuzzy module, which produces an oscillatory velocity pattern when the vehicle approaches the target, may be explained by the fact that two objectives of this controller are in conflict at this stage:

•   Decelerating the vehicle to stop at the target point.
•   Accelerating the vehicle if the velocity approaches zero before the vehicle reaches the target point.

The Target Throttle Fuzzy module is tuned to eliminate these velocity oscillations near the target. This tuned controller is composed of two fuzzy logic controllers: *cruising module* and *slowing-down module*. The *cruising module* is similar to the Target Throttle Fuzzy module. It drives the vehicle while it is away from the target. The *slowing-down module* is triggered if the distance to the target is small. This controller is similar to the cruising controller. However, the ranges of the membership functions of its output are continuously varied based on two inputs: the current change in velocity and the change in the gas/brake angle at

the previous sample. This arrangement succeeds in driving the vehicle to its target without exciting oscillations in the velocity response.

Simulation examples show that the fuzzy controllers successfully guide the vehicle toward the target, while avoiding all obstacles placed in its path. Future work will focus on extending the proposed fuzzy logic controllers to other types of autonomous vehicles, such as autonomous underwater or unmanned aerial vehicles. An intelligent algorithm that is able to generate fuzzy rules and tune the parameters of the fuzzy logic controllers of a vehicle can reduce the implementing time in different systems. Real-time calculation of the throttle angle during cornering could be modeled, but that would require a four wheel model, with calculations of the force at each tire. Criteria to indicate loss of traction on the tires toward the inside of a turn could also be used as inputs for the Cornering Throttle Fuzzy Module.

9. References

Baturone, I. ; Moreno-Velo, F. J., Sanchez-Solano, S. & Ollero, A. (2004). Automatic design of fuzzy controller for car-like autonomous robots, *IEEE Trans. on Fuzzy Systems,* Vol. 12, No. 4, 447-465

Boada, M. J. L.; Boada, B. L. & Diaz, V. (2006). Integrated control of front-wheel steering and front braking forces on the basis of fuzzy logic, *Journal of Automobile Engineering,* 253-267

Byrne R. & Abdallah, C. (1995). Design of a Model Reference Adaptive Controller for Vehicle Road Following, *Mathl. Comput. Modelling*, Vol. 22, No. 4, 343-354

Cang, Y., Yung, N.H.C., Wang, D. W. (2003). A fuzzy controller with supervised learning assisted reinforcement learning algorithm for obstacle avoidance," *IEEE Trans. on Systems, Man and Cybernetics*, Part B, Vol.33, 17 -27

Chen, Q. & Ozguner, U. (2005). Real-time navigation for autonomous vehicles: A fuzzy obstacle avoidance and goal approach algorithm, *Proceedings of the American Control Conference 3*, pp. 2153-2158, Oregon, June 2005, Portland

Dai, X. ; Li, C. K. & Rad, A. B. (2005). An approach to tune fuzzy controller based on reinforcement learning for autonomous vehicle control, *IEEE Trans. on Intelligent Transportation systems*, Vol. 6, No. 3, 285-293

El Hajjaji, A. & Bentalba, S. (2003). Fuzzy path tracking control for automatic steering of vehicles, *Robotics and Autonomous Syst.*, Vol. 43, 203-213

Godjevac J. & Steele, N. (2001). Neuro-fuzzy control for bacis mobile robot behavious, In: *Fuzzy Logic Techniques for Autonomous Vehicle Mavigation,* Driankov D. & Saffotti, A., (Eds.), 97-117, Physica-Verlag, A Springer-Verlag Company, New York

Hessburg T. & Tomizuka, M. (1994). Fuzzy Logic Control for Lateral Vehicle Guidance, *IEEE Trans. On Control Systems*, 55-63

Hoffmann, F. (2001). The role of fuzzy logic control in evolutionary robotics, In: *Fuzzy Logic Techniques for autonomous vehicle mavigation,* Driankov D. & Saffotti, A., (Eds.), 119-147, Physica-Verlag, A Springer-Verlag Company, New York

Hong, C. (1997). Tuning the Fuzzy Control Autopilot Strategy for Driving Pattern Simulation of Road Vehicles, *International Journal of Vehicle Design*, 35-52

Holzmann, H.; Germann, S., Halfmann, C. & Isermann, R. (1998). Intelligent Fuzzy Distance and Cruise Control for Passenger Cars, *Journal of Intelligent and Fuzzy Systems*, 315-327

Kim, S. H.; Park, C. & Harashima, F. (2001). A self-organized fuzzy controller for wheeled mobile robot using an evolutionary algorithm, *IEEE Trans. on Industrial Electronics*, Vol. 48, 467-474

Kodagoda, K. R. S.; Wijesoma, W. S. & Teoh, E. K. (2002). Fuzzy speed and steering control of an AGV, *IEEE Transactions on Control Systems Technology*, Vol. 10, 112 -120

Lee P. & Wang, L. (1994). Collision Avoidance by Fuzzy Logic Control for Automated Guided Vehicle Navigation, *Journal of Robotic Systems*, 743-760

Lumelsky, V. & Stepanov, A. (1987). Path-Planning Strategies for a Point Mobile Automaton Moving Amidst Unknown Obstacles of Arbitrary Shape, *Algoritimica*, Vol. 2, 403-430

Lumelsky, V. & Skewis, T. (1988). A Paradigm for Incorporating Vision in the Robot Navigation Function, *IEEE Conference on Robotics and Automation*, 734-739, Philadelphia, Pennsylvania, April 1988

Lumelsky, V. (1991). A Comparative Study on the Path Length Performance of Maze-Searching and Robot Motion Planning Algorithms, *IEEE Trans. on Robotics and Automation*, Vol. 7, No. 1, 57-66

Maeda, M.; Maeda, Y., & Murakami, S. (1991). Fuzzy Drive Control of an Autonomous Mobile Robot, *Fuzzy Sets and Systems*, 195-204

Murphy, R. R. (2001). Fuzzy Logic for fussion of tractical influences on vehicle speed control, In: *Fuzzy Logic Techniques for autonomous vehicle mavigation*, Driankov D. & Saffotti, A., (Eds.), 73-96, Physica-Verlag, A Springer-Verlag Company, New York

Nobe, S.A. & Wang, F. Y. (2001). An overview of recent developments in automated lateral and longitudinal vehicle controls, *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics 5*, pp. 3447-3452, Nashville, October, 2000

Ohara, H. & Murakami, T. (2004). Tracking control of a compact electrical vehicle trailer based on equivalent dynamics model, *International Workshop on Advanced Motion Control*, AMC, pp. 183-186, Kawasaki, Japan, March 2004

Ollero, A.; Ferrus, J., Sanchez, O. & Heredia, G. (2001). Mobile robot path tracking and visual target tracking using fuzzy logic, In: *Fuzzy Logic Techniques for autonomous vehicle mavigation*, Driankov D. & Saffotti, A., (Eds.), 51-72, Physica-Verlag, A Springer-Verlag Company, New York

Pasquier, M.; Quek, C. & Toh, M. (2001) Fuzzylot: a novel self-organising fuzzy-neural rule-based pilot system for automated vehicles, *Neural networks,* vol. 14, 1099-1112

Peng H. & Tomizuka, M. (1993). Preview Control for Vehicle Lateral Guidance in Highway Automation, *Journal of Dynamic Systems, Measurement, and Control*, 679-686

Rosa, R.; & Garcia-Alegre, M. (1991). Fuzzy Logic Strategies to Control an Autonomous Mobile Robot, *Cybernetics and Systems*, 267-276

Sanchez, O.; Ollero, A. & Heredia, G. (1999). Hierarchical fuzzy path tracking and velocity control of autonomous vehicles, *Integrated Computer-Aided Engineering,* Vol. 6, 289-301

Sauerberger M. & Trabia, M. (1996). Design and Implementation of an Autonomous Control Algorithm for an Omnidirectional Vehicle, *Proceedings of the IASTED International Conference on Robotics and Manufacturing*, 131-134

Seraji, H. (2005). SmartNav: A rule-free fuzzy approach to rover navigation, *Journal of Robotic Systems*, Vol. 22, No. 12, 795-808

Wheeler M. & Shoureshi, R. (1994). A Fuzzy Driver on the Handling Track, *Dynamic Systems and Control*, vol. 1, 431-439

Wong, J. (1993). *Theory of Ground Vehicles*, 2nd ed., John Wiley and Sons, Inc., New York, NY

Ye, C. & Wang, D.W. (2001). A novel navigation method for autonomous mobile vehicles, *Journal of Intelligent & Robotic Systems*, Vol. 32, 361-388

Mobile Robotics, Moving Intelligence

Edited by Jonas Buchli

ISBN 3-86611-284-X
Hard cover, 586 pages
Publisher Pro Literatur Verlag, Germany / ARS, Austria
Published online 01, December, 2006
Published in print edition December, 2006

This book covers many aspects of the exciting research in mobile robotics. It deals with different aspects of the control problem, especially also under uncertainty and faults. Mechanical design issues are discussed along with new sensor and actuator concepts. Games like soccer are a good example which comprise many of the aforementioned challenges in a single comprehensive and in the same time entertaining framework. Thus, the book comprises contributions dealing with aspects of the Robotcup competition. The reader will get a feel how the problems cover virtually all engineering disciplines ranging from theoretical research to very application specific work. In addition interesting problems for physics and mathematics arises out of such research. We hope this book will be an inspiring source of knowledge and ideas, stimulating further research in this exciting field. The promises and possible benefits of such efforts are manifold, they range from new transportation systems, intelligent cars to flexible assistants in factories and construction sites, over service robot which assist and support us in daily live, all the way to the possibility for efficient help for impaired and advances in prosthetics.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Mohamed B. Trabia, Linda Z. Shi and Neil E. Hodge (2006). A Fuzzy Logic Controller for Autonomous Wheeled Vehicles, Mobile Robotics, Moving Intelligence, Jonas Buchli (Ed.), ISBN: 3-86611-284-X, InTech, Available from:
http://www.intechopen.com/books/mobile_robotics_moving_intelligence/a_fuzzy_logic_controller_for_autonomous_wheeled_vehicles

INTECH
open science | open minds