Available online at www.sciencedirect.com
SCIENCE DIRECT®
Robotics and Autonomous Systems 51 (2005) 111–127
Robotics and Autonomous Systems
www.elsevier.com/locate/robot

Robust automatic parallel parking in tight spaces via fuzzy logic☆

Yanan Zhao, Emmanuel G. Collins Jr.*

Department of Mechanical Engineering, Florida A&M University–Florida State University, Tallahassee, FL, USA

Received 26 March 2003; accepted 12 January 2005
Available online 17 March 2005

Abstract

This paper develops and experimentally demonstrates a robust automatic parallel parking algorithm for parking in tight spaces. Novel fuzzy logic controllers are designed for each step of the maneuvering process. The controllers are first demonstrated by simulation using the kinematic model of a skid steering autonomous ground vehicle (AGV). They are then demonstrated experimentally on a skid steering AGV and it is shown that the developed algorithm has the ability to parallel park AGVs in tight spaces under both vehicle localization errors and parking space detection errors. This paper also presents a genetic fuzzy system which uses a genetic algorithm's learning ability to determine effective parameters for the developed fuzzy logic controllers. The genetic fuzzy system is used to tune the fuzzy logic algorithm for both a skid steering AGV and a front-wheel steering AGV.
© 2005 Elsevier B.V. All rights reserved.

Keywords: Autonomous ground vehicle; Fuzzy logic; Genetic algorithms; Mobile robot; Parallel parking

1. Introduction

This research is aimed at developing a robust automatic parallel parking controller for autonomous ground vehicles (AGVs). The objective is to efficiently store an AGV when it is not in operation, which is an important task for both industrial and military applications. This technology can also be used to alleviate the driving burden and enhance safety in next-generation passenger vehicles.

The automatic parallel parking problem has previously attracted a great deal of attention among researchers. Current approaches to solving this problem can be classified into two main categories: (1) the path planning approach, where a feasible geometry path is planned in advance, taking into account the environmental model as well as the vehicle's dynamics and constraints, and then control commands are generated to follow the reference path; (2) the skill-based approach, where fuzzy logic or neural networks are used to acquire and transfer an experienced human driver's parking skill to an automatic parking controller. There

☆ Prepared through collaborative participation in the Robotics Consortium sponsored by the U.S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0012. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.
* Corresponding author.
E-mail address: ecollins@eng.fsu.edu (E.G. Collins Jr.).

0921-8890/$ – see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2005.01.002

is no reference path to follow and the control command is generated by considering the orientation and position of the vehicle relative to the parking space.

For path planning Paromtchik and Laugier [13,19,20] proposed a parallel parking approach for a nonholonomic vehicle. A parking space is scanned before the vehicle reverses into the parking bay. The vehicle then follows a sinusoidal path in backward motion; that is the control commands (steering angle and velocity) are generated such that the corresponding (x, y) path is sinusoidal. To keep the vehicle from colliding with the front left corner of the parking bay, a collision-free start position is obtained from an off-line lookup table according to the length of the parking bay and the lateral distance of the vehicle to the front left corner of the parking bay. Jiang and Seneviratne [12] also studied sensor guided autonomous parking, where the process consists of three phases: scanning, positioning and maneuvering. The path in the maneuvering phase is constructed by two circular arcs of minimum radius tangentially linked to each other. A forbidden area inside the parking bay is defined for the maneuvering phase to avoid possible collision. Xu et al. [24] planned a quintic polynomial curve for the reference path, where the steering angle was obtained by the instant turning radius of the vehicle.

For skill-based parking an artificial neural network was trained to directly map the video sensor's CCD-image of the environment to the corresponding steering angle in the direct neural control architecture for parallel parking [6]. In the fuzzy logic approaches of [10,17,18], the control command (i.e., the steering angle) was generated based on the relative longitudinal and lateral distance of the vehicle to the parking space and the orientation of the vehicle. Fuzzy rules were built for each of the parking steps [10] or for different parking positions [17,18].

Path planning is an open-loop approach. The accuracy of the resulting path depends on the accuracy of actuators. However, the effect of control actions is not completely reliable, for example, wheels may slip. The position error caused by actuators cannot be compensated by an open-loop strategy. In the case of open-loop parking, even though position errors may be compensated by subsequent iterative motions (i.e., backward or forward motions), this compensating strategy increases the time and cost of implementation. In contrast, a skill-based fuzzy approach generates control commands at each sampling period based on the current position of the vehicle relative to the parking space, it has the ability to compensate actuator errors promptly. Thus, the fuzzy logic approach is a feedback strategy.

Both the path planning approach and the fuzzy approach rely on the environmental modelling and sensor information which are in general approximate, having uncertainties. Fuzzy systems employ a mode of approximate reasoning which enables them to make robust and meaningful decisions under uncertainty and partial knowledge. The nature of fuzzy logic leads to robust control algorithms in spite of sensor uncertainties and control errors.

A fuzzy control based approach was selected in this research also with the intention of producing a generic design, so that the control algorithms could be developed on a laboratory robot and then easily transferred to other prototypes.[1] In addition, fuzzy logic control provides the flexibility to apply a non-linear control law derived from an experienced human-driver and expressed using words and If–Then rules. Further, fuzzy logic navigation allows various behaviors to be easily combined through a command fusion process [7,21,25]. For cost effective implementation a fuzzy controller can be implemented using lookup tables.

The size of the parking space has significant impact on the degree of difficulty of the parallel parking maneuvering. In the category of fuzzy logic approaches references [10,17,18] do not explicitly discuss the size of the maneuvering space considered. However, it is observed that in [17] and [18], the size of the maneuvering space used was twice the length of the vehicle. In the category of path planning approaches references [13,19,20] also do not discuss the size of the parking space. In this category, the smallest parking space used in [12] is 1.67 times the length of the vehicle. To improve the maneuverability and capability of AGVs, the focus of this research is to develop simple fuzzy controllers that can reverse vehicles into tight spaces and to demonstrate the resulting parking algorithm experimentally using the ATRV-Jr mobile robot shown in Fig. 1. A tight space maneuvering algorithm is obtained

[1] This feature of the proposed fuzzy logic approach is only demonstrated in this paper via simulation since the approach is experimentally applied to only one robot. Ultimately, the hard evidence requires experimental demonstration since it is difficult to incorporate real world effects in a simulation.

Fig. 1. ATRV-Jr mobile robot.

by proper tuning of the fuzzy controllers. Tuning can be conducted by human operators using trial and error, which is generally time-consuming. However, a so-called genetic fuzzy system uses a genetic algorithm's learning ability to automatically tune the parameters of fuzzy controllers.

Genetic algorithms (GAs) are a part of evolutionary computing, which is a rapidly growing area of artificial intelligence. GAs can be used to tune the membership functions, rule bases and scaling factors of fuzzy controllers [1,2,9,11,14,16]. In this paper, GAs are used in the determination of membership functions and scaling factors for the reverse-motion fuzzy logic controller only (which is the most complex of the three fuzzy controllers that comprise the proposed automatic parking algorithm). The prototype vehicle used for fuzzy logic and genetic fuzzy controller design is a skid-steering vehicle. The genetic fuzzy system is later used to tune the fuzzy algorithm for front-wheel steering AGVs. For both types of vehicles, the rule bases are the same, the only changes made are the membership functions and scaling factors. The results demonstrate the suitability of the reverse-motion maneuvering algorithm for a variety of vehicle platforms.

This paper is organized as follows. Section 2 describes the parallel parking problem. Section 3 presents a parallel parking algorithm based on fuzzy logic. Section 4 gives experimental results that demonstrate the capability of the fuzzy algorithm. Section 5 considers two different steering models and tunes the fuzzy algorithm for each system using a genetic algorithm. Finally, Section 6 gives conclusions.

2. The parallel parking problem

Two important aspects for parallel parking are parking space detection and automatic controller design. In this section, the parking space detection method is discussed and the process of parallel parking is described.

2.1. Parking space detection

The map of the vehicle's environment has to be built when the vehicle is passing the maneuvering space. Either image sensors or ultrasonic sensors can be used for the task [6,12,13,24]. For examples, by using one or two digital cameras, the image around the vehicle can be acquired in real time, and then an image processing algorithm can be employed to identify the parking space. The built map can provide a contour constructed by the obstacles on the right side of the vehicle. Then, a sufficiently large rectangle can be fit into the contour [10]. As shown in Fig. 2, this rectangle will be the maneuvering space available for the vehicle. To focus the study on the controller design part, it is assumed below that the parking space is detected and to the right of the vehicle. (It is straightforward to generalize the results to the case in which the parking space is on the left-hand side.)

2.2. The three-step maneuvering process

The detected space can be described as shown in Fig. 3. The size of the rectangular space is defined as $h_p \times l_p$, and BK, FT and SE represent the "back", "front" and "side" of the space, respectively. The origin of the local coordinate system is chosen as the intersection of BK and SE. The reverse-motion maneuvering algorithm developed here corresponds

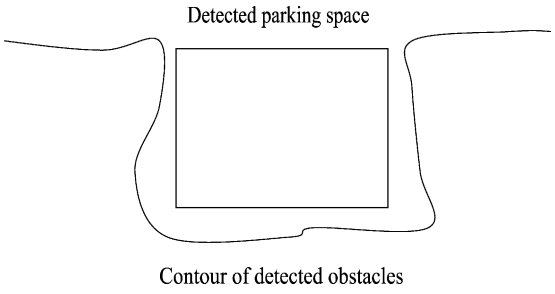

Fig. 2. Maneuvering space detection.

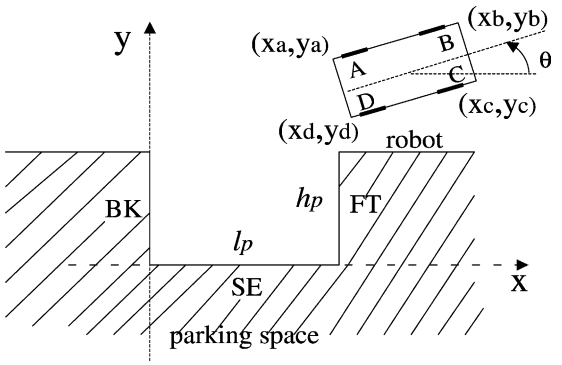

Fig. 3. Maneuvering space and the local coordinate system.

to the parallel parking procedure of an experienced human driver [26] and the whole maneuvering process can be completed with the following three steps as illustrated by Figs. 4–7.

In the first step (see Fig. 4), the vehicle is navigated forward to reach a ready-to-reverse position with the vehicle orientation parallel to the parking space. Here, the ready-to-reverse position for the center of the vehicle is chosen as $(l_p + 0.5l, h_p + 0.65b)$ in the local coordinate system, where l and b represent, respectively, the length and width of the vehicle. To reach both the desired position and orientation simultaneously requires a complex fuzzy system. Hence, to yield a simpler and more easily implementable algorithm, this step is divided into two sub-steps.

The task of the first sub-step is to have the vehicle center reach an intermediate position $(0.9l_p, h_p + 0.65b)$

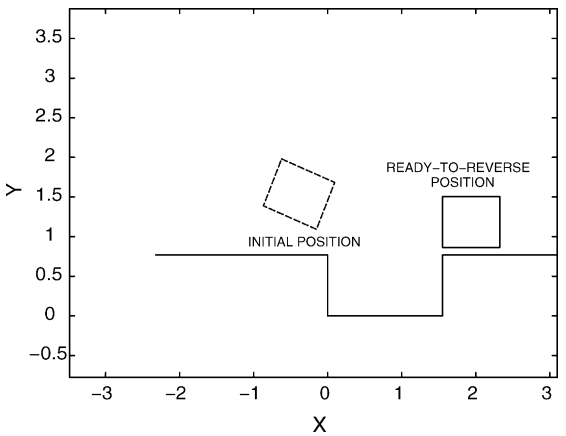

Fig. 4. Reach a ready-to-reverse position.

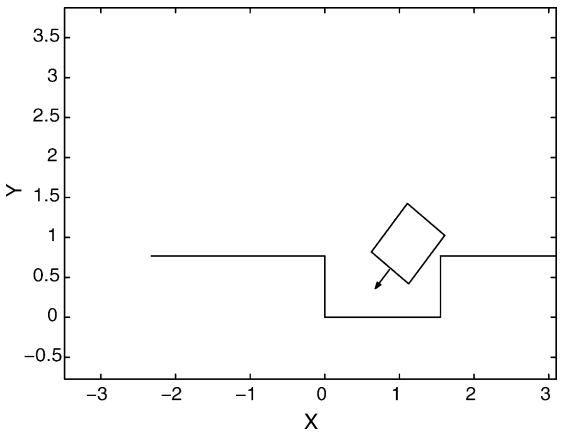

Fig. 5. Reverse with increasing θ.

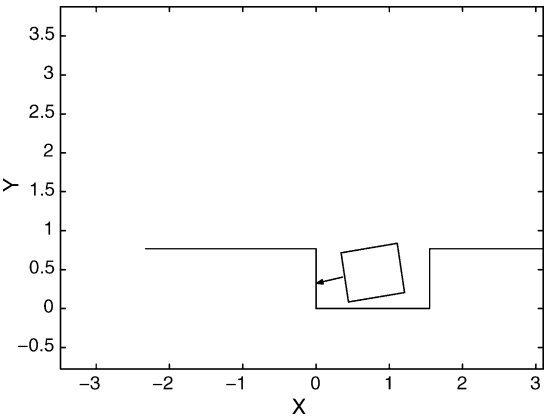

Fig. 6. Reverse with decreasing θ.

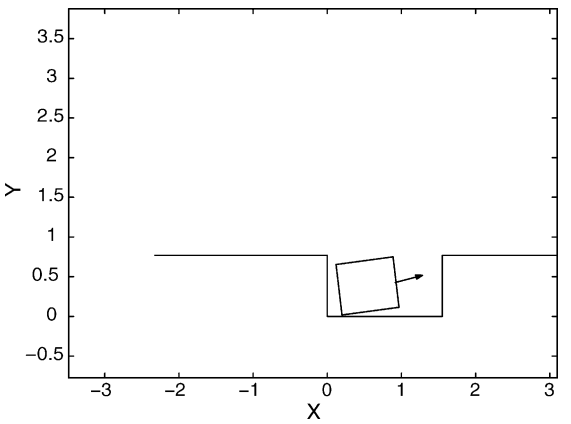

Fig. 7. Adjust forward inside the space.

without considering the orientation angle of the vehicle. The desired y position of the vehicle is reached at this stage. In the second sub-step, the orientation angle is adjusted while the vehicle moves forward to reach the desired x position. Although this step does not ensure that the orientation is precisely parallel to the space, the error in the orientation angle is made very small.

In the second step (see Figs. 5 and 6) the vehicle is first reversed into the maneuvering space with an increasing θ until its rear right wheel is at a certain distance from the boundary SE of the space. Then the vehicle is reversed with decreasing θ until one of the rear wheels is very close to the boundary BK of the space.

In the third step the vehicle is moved forward to adjust its position inside the space (see Fig. 7). The desired final position of the vehicle is that it is parallel to and at the center of the space.

The second and third steps can be repeated several times until the desired final position is reached within some tolerance.

3. Fuzzy logic controller (FLC) design

Each step of the maneuvering process can be implemented with the use of fuzzy logic techniques as described in the following subsections. The iRobot ATRV-Jr is the autonomous vehicle considered here. The wheels on one side of the ARTV-Jr are mechanically coupled and it uses skid steering to maneuver the robot [27]. To easily generalize the research results to vehicles with Ackerman steering, the outputs of all the fuzzy logic control systems proposed here are taken as steering rate. An alternative is to have the fuzzy controller output the left and right wheel velocities as was done in [15] and [23] for fuzzy obstacle avoidance and goal seeking for a skid steering vehicle.

3.1. Kinematic equations of mobile robot with skid steering

The simulation of the parallel parking algorithm is based on the kinematic model of the ARTV-Jr. As illustrated in Fig. 8, let (x, y) denote the coordinate of the center of the vehicle in the local coordinate system defined previously, v the total translational velocity, v_r and v_l the velocity of the right side and left side wheels,

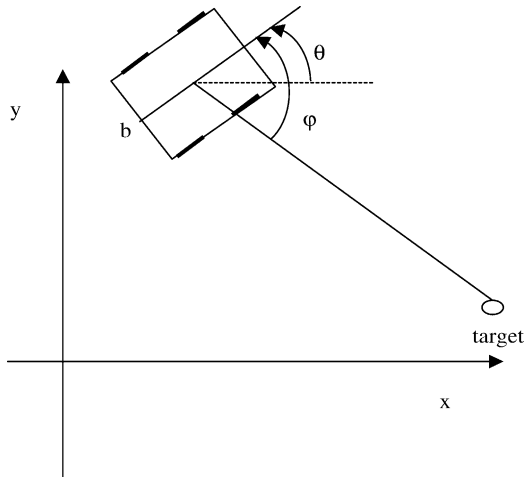

Fig. 8. The vehicle kinematics and heading angle difference.

respectively, θ the angle between the positive x-axis and the main axis of the vehicle, and $\dot{\theta}$ the steering rate. Assuming constant velocity, the kinematic model of a skid steering mobile robot can be described as [8]:

$$\theta(i+1) = \theta(i) + \dot{\theta}(i)\,\mathrm{d}t, \tag{1}$$

$$x(i+1) = x(i) + v(i+1)\cos(\theta(i+1))\,\mathrm{d}t, \tag{2}$$

$$y(i+1) = y(i) + v(i+1)\sin(\theta(i+1))\,\mathrm{d}t, \tag{3}$$

where

$$v(i) = \frac{v_r(i) + v_l(i)}{2}, \tag{4}$$

$$\dot{\theta}(i) = \frac{v_r(i) - v_l(i)}{b}. \tag{5}$$

At each sampling time, the velocities v_r and v_l can be obtained using (4) and (5). The motor coupled with the corresponding wheels can be adjusted to reach the desired speed by lower level controllers. Below, it is assumed that the steering rate is adjusted to maneuver the robot while the translation speed v is taken as a constant. Hence, the process of accelerating the vehicle from rest and decelerating to rest is omitted from the simulation.

3.2. FLC for moving the vehicle to a ready-to-reverse position

As discussed above, this first step is completed by a goal-seeking behavior followed by an orienta-

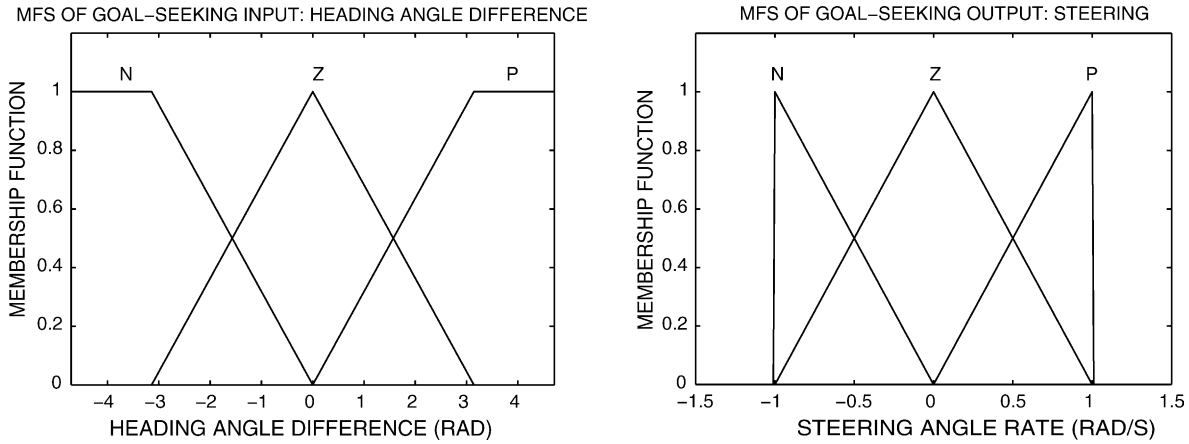

Fig. 9. Membership functions for the goal-seeking—input ϕ (left) and Output $\dot{\theta}$ (right).

tion adjusting behavior. The fuzzy controller designed for the goal-seeking behavior has one input and one output. The input is the heading angle difference ϕ, which is the angle from the line connecting the vehicle's center to the target to the heading direction of the vehicle as shown in Fig. 8. The angle ϕ is positive when the line connecting the center of vehicle and target rotates counterclockwise to the heading direction. The output of the fuzzy controller is the steering rate of the vehicle $\dot{\theta}$, where θ is the angle between the positive x-axis and the main axis of the vehicle (see Fig. 8). The angle θ is positive when the positive x-axis rotates counterclockwise to the main axis of the vehicle. The membership functions for both the input and output are illustrated in Fig. 9, where N, P and Z represent negative, positive and zero, respectively.

The fuzzy inference rules for the goal-seeking behavior are listed in Table 1. For a skid steering vehicle, since $\dot{\theta}$ is positive when the right wheels' velocity is bigger than the left wheels' velocity, the fuzzy rules actually require that if the target is to the right of the robot's heading direction, the robot should turn to its right. The basic principles of goal-seeking are as follows:

- If the target is to the right of the robot's heading direction, the robot should turn to its right.
- If the target is to the left of the robot's heading direction, the robot should turn to its left.
- If the target is directly to the front of the robot, the robot should keep straight.

The task of the orientation adjusting sub-step is to adjust the heading of the vehicle while reaching the desired x position for the center of the vehicle. The heading angle will be very close to zero after this adjustment. For this fuzzy controller the input is the orientation angle θ, and the output is the steering rate $\dot{\theta}$. The membership functions are defined as shown in Fig. 10, where NB, NM, Z, PM and PB represent negative big, negative middle, zero, positive middle and positive big respectively. The fuzzy inference rules proposed are shown in Table 2. This set of fuzzy rules require that the sign of $\dot{\theta}$ should always be opposite to the sign of θ. The basic notions of the orientation adjusting is as follows:

- If the steering angle is positive, the steering angle rate should be negative.
- If the steering angle is negative, the steering angle rate should be positive.

Table 1
Fuzzy rules for goal-seeking

ϕ	N	Z	P
$\dot{\theta}$	P	Z	N

Table 2
Fuzzy rules for orientation adjustment

θ	NB	NM	Z	PM	PB
$\dot{\theta}$	PB	PM	Z	NM	NB

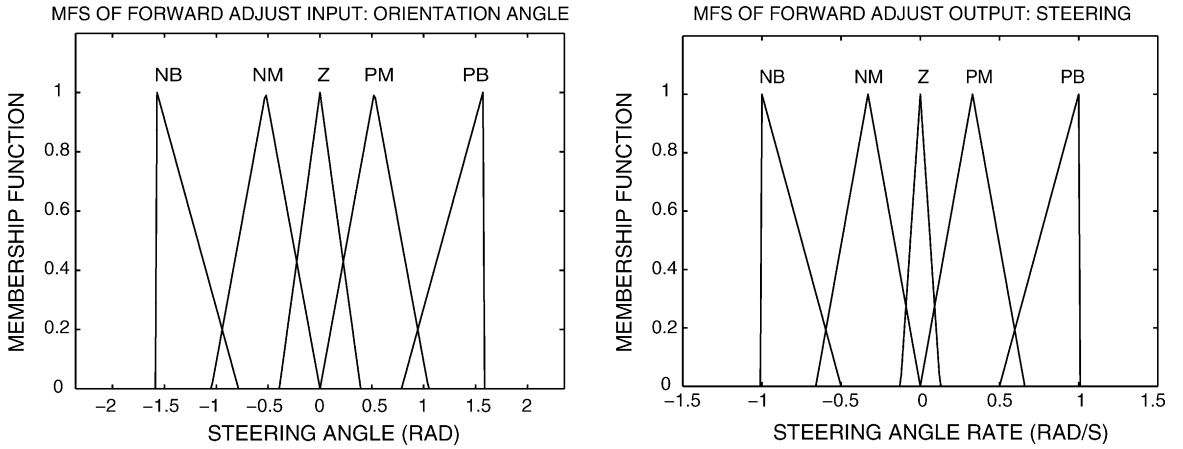

Fig. 10. Membership functions for orientation adjustment—input θ (left) and output $\dot{\theta}$ (right).

- If the steering angle is zero, the steering angle rate should be zero.

3.3. FLC for reversing the vehicle into the parking space

The reverse maneuvering into the parking space requires a more complex fuzzy logic controller. As shown in Fig. 3, the coordinate of the left rear corner of the vehicle in the local coordinate system is defined as (x_a, y_a), and the coordinate of the right rear corner of the vehicle is defined as (x_d, y_d). Two new variables, x_{a1} and y_{d1}, are defined by $x_{a1} = x_a/l_p$ and $y_{d1} = y_d/h_p$; they represent the relative position of the rear of the vehicle with respect to the origin of the space. The fuzzy logic controller for this step has three inputs, x_{a1}, y_{d1} and the orientation angle θ. The output is the steering rate $\dot{\theta}$. Figs. 11 and 12 show the proposed membership functions, where S, B, VB represent small, big and very big respectively. The three dimensional fuzzy rules are shown in Table 3; there are a total of 18 rules. Empty rules in Table 3 mean the corresponding combination of inputs is invalid (i.e., they imply that either the vehicle is moving away from the parking space or it has entered one of the shaded regions in Fig. 3). The rationale behind several of the rules is presented here.

- If θ is negative and x_{a1} is small and y_{d1} is small, then $\dot{\theta}$ is positive big, i.e., when the vehicle is very close to both of the boundaries, and its orientation angle

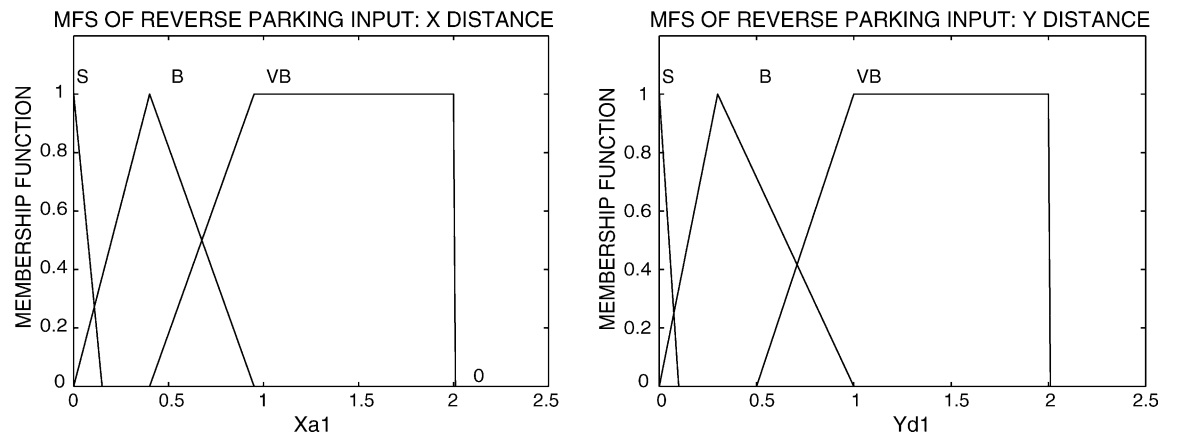

Fig. 11. Membership functions for the backing up—input x_{a1} (left) and y_{d1} (right).

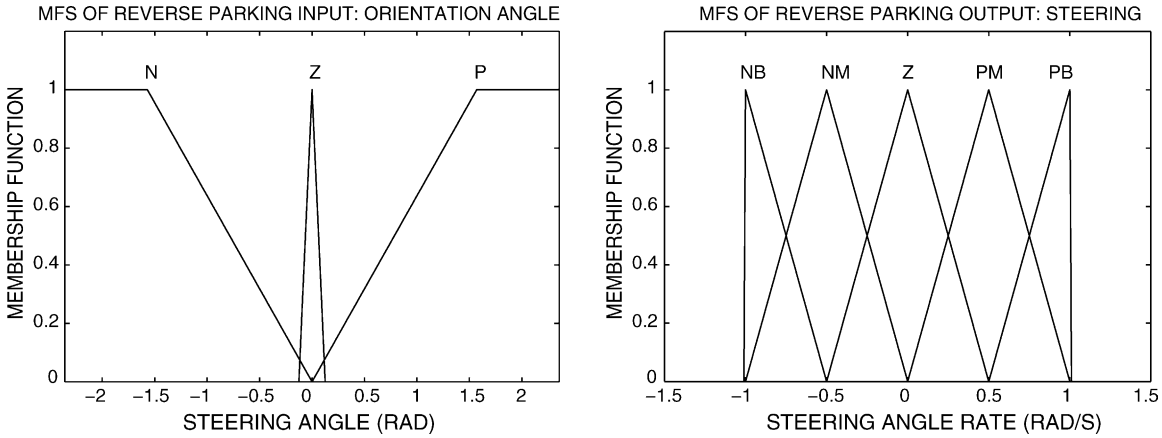

Fig. 12. Membership functions for the reverse maneuver—input θ (left) and output $\dot{\theta}$ (right).

is negative, the steering rate should be a big positive number to make the orientation angle positive.

- If θ is zero and x_{a1} is very big and y_{d1} is very big, then $\dot{\theta}$ is zero, i.e., when the vehicle is parallel to the parking space, and the vehicle is outside the parking space, the vehicle should continue to reverse in the same direction.
- If θ is positive and x_{a1} is big and y_{d1} is big, then $\dot{\theta}$ is zero, i.e., when the vehicle is in the middle of the parking space, and the orientation angle is positive, the vehicle should keep the same steering angle.

3.4. FLC for navigating the vehicle forward inside the parking space

The developed fuzzy logic controller has one input, orientation angle θ, and one output, the steering rate $\dot{\theta}$. The membership functions and fuzzy inference rules are exactly the same as that of the orientation adjustment described in the first step since this step accomplishes essentially the same task, that is adjusting the orientation while simultaneously moving the vehicle forward.

Table 3
Fuzzy rules for the backing up step

	$X_{a1} \backslash y_{d1}$	S	B	VB
	S	PB	PB	
$\theta = N$	B	PM	PB	PB
	VB			PM
	S	Z	Z	
$\theta = Z$	B	Z	PB	PB
	VB			Z
	S	NB	Z	
$\theta = P$	B	NM	Z	PM
	VB			NB

3.5. Algorithm summary

The three major steps of the parallel parking algorithm were overviewed in Section 2.2. The fuzzy logic controllers designed for each of the three steps were described in Sections 3.2–3.4. As a summary, Fig. 13 shows the flowchart of the entire algorithm. Using the fuzzy logic algorithm of Section 3.2, the vehicle first moves to a ready-to-reverse position until the target position, described in Section 2.2, is reached within a predefined tolerance. Secondly, using the fuzzy logic algorithm of Section 3.3, the vehicle reverses into the parking space until it is very close to the parking boundary described in Section 2.2. Thirdly, using the fuzzy logic algorithm of Section 3.4, the vehicle navigates forward inside the parking space. The forward navigation stops when either the target parking position is reached or when the vehicle is very close to the front boundary of the parking space. In the latter case, the vehicle will perform the second and third steps repeatedly until the target parking position is reached.

4. Experimental implementation

The developed algorithm was first successfully demonstrated under simulation environments using

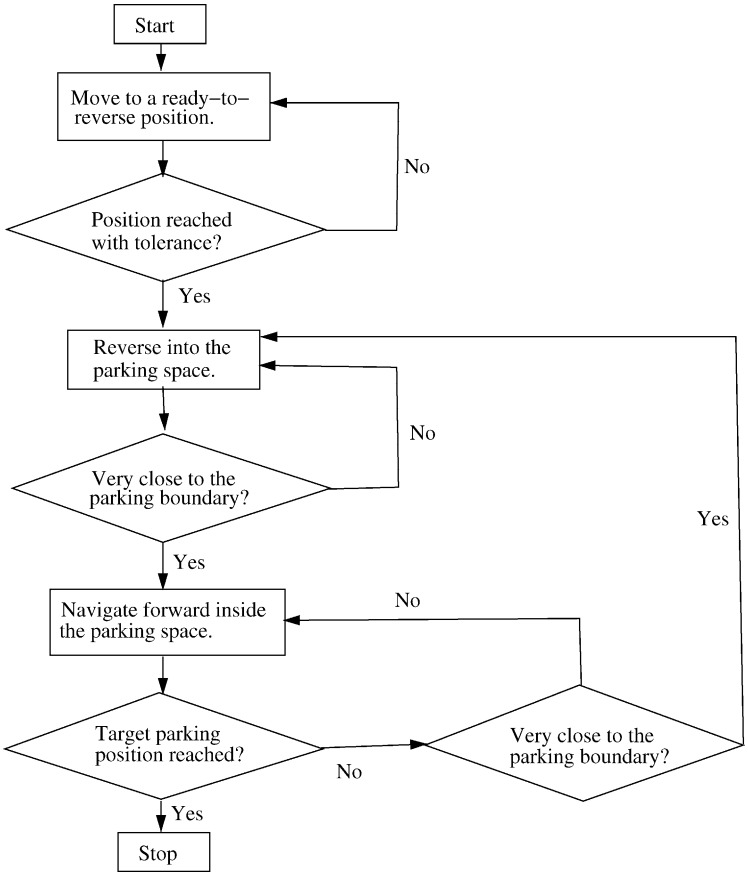

Fig. 13. Flowchart of the parallel parking algorithm.

Matlab and the Matlab Fuzzy Logic Toolbox. Then the algorithm was implemented on an ATRV-Jr mobile robot. The experimental implementation related issues are discussed in detail here.

The ATRV-Jr is equipped with a suite of sensors including wheel encoders, ultrasonic sonar, a CCD camera, a laser range finder, an inertial navigation sensor, a compass and DGPS. The onboard Pentium II PC runs RedHat Linux, and the communication with the robot is over wireless Ethernet.

4.1. Vehicle localization

In order to autonomously navigate and perform useful tasks, a vehicle needs to know its exact position and orientation. The wheel encoders mounted on the motor are used to perform vehicle localization in the experiment for low cost implementation.

It was experimentally observed that errors in odometry accumulate over time due to inaccuracies in the kinematic model, precision limitations of the encoders, and wheel slippage. Several methods can be used to improve the robustness of the fuzzy algorithm under vehicle localization. For high implementation cost alternatives, Kalman filter or extended Kalman filter technique can be used to carefully calibrate odometric data with state measurement signals provided by a redundant sensor (e.g., a gyroscope) to improve the robustness of localization [8,22]. For low implementation cost alternatives, sonar sensor can be integrated with the system to enhance the robustness as discussed below.

4.2. Sonar sensor integration

The reverse-motion maneuvering fuzzy controller requires variables describing the relative position of the

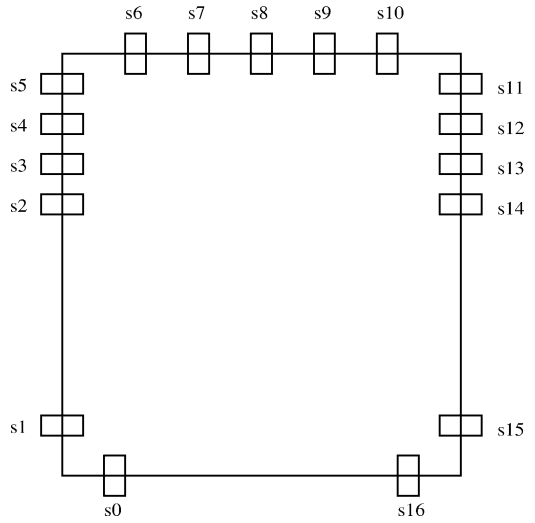

Fig. 14. Sonar configuration on ATRV-Jr.

vehicle to the parking space as input variables. These relative positions can be obtained by vehicle localization with respect to the detected parking space. To compensate parking space detection errors and to prevent collision due to vehicle localization errors during the parking process, information from other sensors (e.g., ultrasonic sensors) can be used.

The ATRV-Jr used in this research is mounted with a ring of 17 sonar: five in the front, five on each side and two at the rear, as shown in Fig. 14. The sonar sensors' information can be integrated in the algorithm for two tasks. First, to compensate for parking space detection and localization errors, sonar *s*15 is used to detect the edge FT of the maneuvering space (as shown in Fig. 3) during the reverse-motion step. In Matlab simulation this edge is solely determined by vehicle localization. In experimental implementation the data obtained from *s*15 is checked at each sampling time and whenever the detected distance is larger than 90% of the width of the parking space, it is assumed the edge FT is detected. Once the ready-to-reverse position is reached, the vehicle should reverse parallel to the parking space until the edge FT is detected. Second, to avoid possible collision of the vehicle with edges BK and FT while maneuvering within the parking space, data from sonar *s*0 and *s*8 are used to determine the switching between reversing and forward-adjusting. In particular, if sonar *s*0 detected an obstacle within 15 cm during the reverse maneuvering, the vehicle was switched to forward maneuvering, and if sonar *s*8 detected an obstacle within 15 cm during forward maneuvering, the vehicle was switched to reverse maneuvering. (It should be mentioned that the above algorithm works for a rectangular parking space; for irregular parking spaces more advanced sensor integration algorithms are needed.)

Fig. 15. Initial position.

4.3. Experimental results

The ATRV-Jr has a basic length of 77.5 cm and a width of 64 cm. Considering the installation of a laser range finder in the front, the actual length of the vehicle is 100.5 cm. The size of the maneuvering space used in the experiments is 1.4 times the length and 1.2 times the width of the vehicle. The fuzzy logic controllers were programmed in C^{++} and the code was uploaded to the onboard PC of the ATRV-Jr. The experimentally implemented fuzzy logic controllers were the same controllers as used in simulation except the output scaling factors were adjusted for the real dynamics of the mobile robot. The translation speed used in the experiment is 8 cm/s (i.e., 0.288 km/h). Figs. 15 to 20 show the

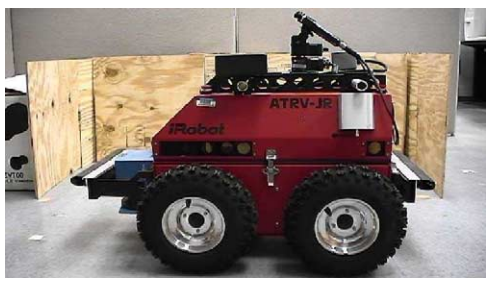

Fig. 16. Passing by the space.

Fig. 17. Ready to reverse position.

Fig. 18. Back up maneuvering.

Fig. 19. Adjust forward maneuvering.

maneuvering process of the ATRV-Jr from an initial position to the final position. Experiments conducted under different initial positions achieved similar results.

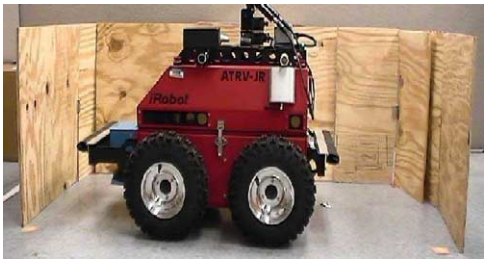

Fig. 20. Final position reached.

4.3.1. Robustness under vehicle localization errors

The vehicle localization errors in odometry accumulate over time due to inaccuracies in the kinematic model, precision limitations of the encoders, and wheel slippage. The errors can be cleared whenever the onboard PC is restarted. In one set of parking experiments, the entire parking process was repeated five times without reinitializing the robot odometry. It was observed that, due to localization errors, the ready-to-reverse position reached by the first parking step was different for each parking process. The desired ready-to-reverse position was $x = 1.951$ m, $y = 1.248$ m and $\theta = 0$ rad relative to the local coordinate system. Table 4 compares the odometry readings from the onboard PC and manually measured data from the five experiments. It is seen that while the data from odometry reading matched the desired target position well, the data from the real (i.e., the manual) measurement did not. In particular, localization errors along the x-axis were much bigger than that along the y-axis and orientation angle, and the later experiments yielded larger errors. The differences in the ready-to-reverse positions led to different reverse-motion trajectories in the second step of the parking process. In the first four experiments, the mobile robot was successfully parked inside the parking space by the fuzzy controller. In the fifth experiment,

Table 4
Vehicle localization errors

Experiment no.	Odometry x (m)	Odometry y (m)	Odometry θ (rad)	Measured x (m)	Measured y (m)	Measured θ (rad)
1	1.9596	1.2416	-7.7182×10^{-4}	1.7650	1.2365	0.0120
2	1.9525	1.2402	-6.0643×10^{-4}	1.7095	1.2440	0.0159
3	1.9579	1.2455	-10.1991×10^{-4}	1.6613	1.2290	−0.0649
4	1.9599	1.2456	-5.7887×10^{-4}	1.6075	1.2370	−0.0156
5	1.9559	1.2485	-9.0965×10^{-4}	1.5560	1.2485	−0.0097

the vehicle localization error along x-axis was already $1.951 - 1.5560 = 0.395$ m. Under this case, the robot hit the edge BK showing the limitation of the algorithm.

4.3.2. Robustness under parking space detection errors

As described above sonar sensor $s15$ is used to detect edge FT, and the reverse-motion fuzzy controller takes effect only when this edge is detected. Thus the possible collision of the robot with edge FT due to space detection errors is avoided. For example, it was assumed that the detected length of the parking space was 1.6 times the length of the vehicle and the edge BK was properly detected, while the actual length of the parking space was only 1.4 times the length of the vehicle. In this case, the robot bumps the edge FT if sonar $s15$ is not integrated for the edge detection of FT. However, due to the integration of sonar $s15$, the vehicle was able to successfully park.

5. Fine-tuning of the FLCs with genetic algorithms

The size of the parking space has significant impact on the degree of difficulty of reverse-motion maneuvering. The smaller the space, the more difficult it is to perform the reverse-motion maneuvering smoothly. The ability to parallel park in small parking spaces of the reverse-motion maneuvering algorithm was obtained by significant tuning of the proposed fuzzy controllers.

Fine-tuning of fuzzy logic controllers can be performed by either manual trial and error or by using artificial intelligence. In this research manual tuning was first tried and later genetic algorithms (GAs) were used to learn effective parameters for the reverse-motion fuzzy controller, the most complex controller in the parallel parking algorithm. GAs can also be used to tune the remaining FLCs although this was not carried out in this research.

GAs search the solution space of a function through the use of simulated evolution, i.e., the survival of the fittest strategy. In general the fittest individuals of any population tend to reproduce and survive in the next generation, thus improving successive generations [3]. The learning ability of genetic algorithms can be used to choose the most effective parameters for fuzzy logic systems. In this research, a genetic algorithm was used to determine the membership functions and scaling factors only. This is because the expert-based fuzzy rules for parallel parking are standard and there is not much room to modify them. Hence, adjusting these rules along with the other parameters leads to inefficient tuning efforts.

For the specific problem studied, the first priority is to keep the AGV from colliding with the boundaries of the maneuvering space, i.e., constraints need to be enforced in the optimization process. Thus, a constrained optimization problem was considered. A penalty term was added to the fitness function to enforce the constraints. This term was zero if no collision occurred, and was given an extremely high value if the AGV collided with any of the boundaries. In this way, the feasibility of the optimal result was guaranteed.

5.1. Design variables

The first issue of GA tuning is to select the design variables. As shown in Figs. 11 and 12 the membership functions of the inputs and output proposed for the fuzzy controller are either triangular-shaped or trapezoidal-shaped. The triangular curve can be represented by the following function,

$$f(x; a, b, c) = \begin{Bmatrix} 0, x \le a \\ \dfrac{x-a}{b-a}, a \le x \le b \\ \dfrac{c-x}{c-b}, b \le x \le c \\ 0, c \le x \end{Bmatrix} \tag{6}$$

while the trapezoidal curve can be represented by

$$f(x; d, e, f, g) = \begin{Bmatrix} 0, x \le d \\ \dfrac{x-d}{e-d}, d \le x \le e \\ 1, e \le x \le f \\ \dfrac{g-x}{g-f}, f \le x \le g \\ 0, g \le x \end{Bmatrix} \tag{7}$$

Thus the shape of a triangle membership function is determined by the three parameters (a, b, c) and the shape of a trapezoidal membership function is determined by the four parameters (d, e, f, g). Tuning the membership functions requires adjustment of the values of these parameters.

Based on the structure of the inputs and output of the fuzzy logic system, the membership functions of the three input variables x_{a1}, y_{d1}, θ, and the output, steering rate $\dot{\theta}$ can be defined as:

$$\mu_{x1} = f(x; 0, \mathrm{d}x1, \mathrm{d}x2),\ \mu_{x2} = f(x; \mathrm{d}x3, \mathrm{d}x4, dx5),\ \mu_{x3} = f(x; \mathrm{d}x6, \mathrm{d}x7, 2, 2),$$
$$\mu_{y1} = f(y; 0, \mathrm{d}y1, \mathrm{d}y2),\ \mu_{y2} = f(y; \mathrm{d}y3, \mathrm{d}y4, \mathrm{d}y5),\ \mu_{y3} = f(y; \mathrm{d}y6, \mathrm{d}y7, 2, 2),$$
$$\mu_{\theta 1} = f(\theta; -a1, 0, a1),\ \mu_{\theta 2} = f(\theta; -\pi, -\pi, -\tfrac{1}{2}\pi, -a2),\ \mu_{\theta 3} = f(\theta; a2, \tfrac{1}{2}\pi, \pi, \pi),$$
$$\mu_{\dot{\theta}1} = f(\dot{\theta}; -1, -s6, -s5),\ \mu_{\dot{\theta}2} = f(\dot{\theta}; -s4, -s3, -s2),\ \mu_{\dot{\theta}3} = f(\dot{\theta}; -s1, 0, s1),\ \mu_{\dot{\theta}4} = f(\dot{\theta}; s2, s3, s4),\ \mu_{\dot{\theta}5} = f(\dot{\theta}; s5, s6, 1)$$

In these expressions, some of the parameters are fixed (e.g., 0 for μ_{x1}, 2 for μ_{x3}). These parameters were not optimized because they define the boundary limit of the universe of discourse of membership functions.

The output scaling factor is also very important to the fuzzy system. It plays a role equivalent to that of the control gain in a conventional control system and is therefore included in the design variable set. The final design variable set selected is (d$x1$, . . ., d$x7$, d$y1$, . . ., d$y7$, $a1$, $a2$, $s1$, . . ., $s6$, sf], where sf represents the output scaling factor. There are a total of 23 parameters to be optimized in this design.

Another important issue for the optimization problem is to determine the cost function. In this research, the tuning was first performed for a skid-steering system. Later, the tuning was extended for a front-wheel steering system to test the suitability of the developed algorithm for other vehicle platforms. For both of the steering systems, the design variables were chosen as described above. However, different cost functions were chosen for different steering models as shown below.

5.2. Tuning for a skid-steering vehicle

For a skid-steering vehicle, in particular the ATRV-Jr, the cost function was chosen as $(3\frac{x_a}{l_p} + 2\frac{y_d}{h_p} + \frac{y_c}{h_p} + penalty)$, where y_c is the right front y coordinate of the vehicle. The *penalty* term is zero if no collision occurs during the reverse-motion, and it is a very high value if there is a collision. The cost function considers not only the position of the AGV relative to the boundary of the maneuvering space (by the terms $3\frac{x_a}{l_p}$ and $2\frac{y_d}{h_p}$), but also the orientation of the vehicle (by the term $\frac{y_c}{h_p}$). The weights 3, 2 and 1 represent the relative importance of the three terms.

A range for each of the design parameters was defined before the optimization. A GA was implemented with a population size of 20 and 100 generations for this tuning. The chromosome encoding is with a floating-point representation. To illustrate the tuning results more clearly, Figs. 21–24 show the membership functions obtained by GA optimization.

The obtained membership functions and scaling factor were used for the simulation of the reverse motion. Even though there is no obvious improvement relative to the results obtained by extensive manually tuning, the tuning process with the GA is more systemic and results in significant time savings for the designers.

5.3. Tuning for a front-wheel steering vehicle

One advantage of fuzzy logic control is that it can be easily transferred from one platform to another with few modifications. The reverse-motion maneuvering algorithm previously developed for a skid-steering system was also tuned for an AGV with front-wheel steering (i.e., Ackerman steering) to test its suitability.

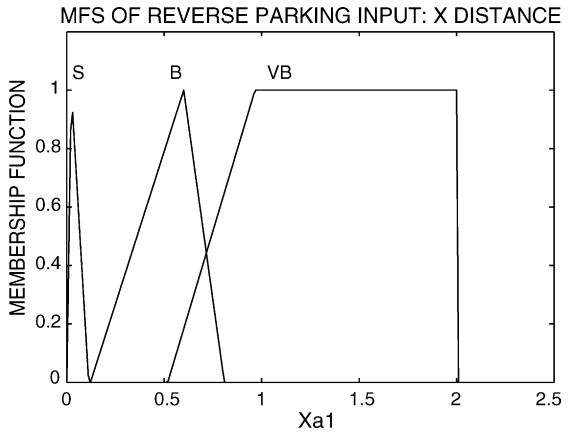

Fig. 21. Membership functions for input variable x_{a1}—skid steering system.

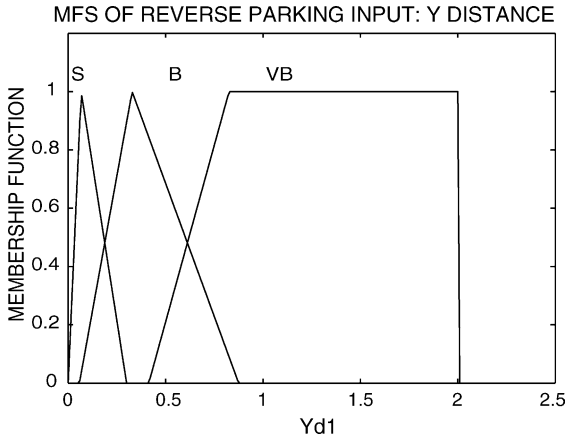

Fig. 22. Membership functions for input variable y_{d1}—skid steering system.

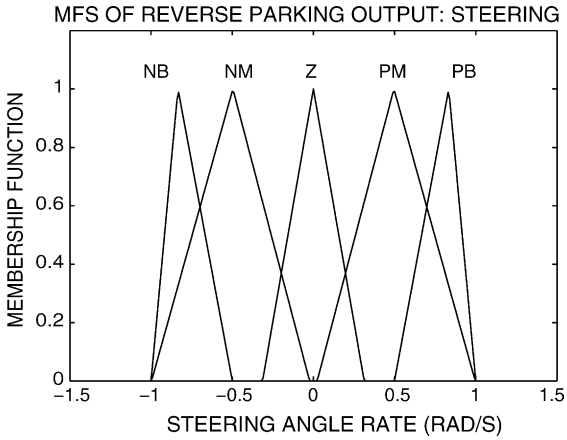

Fig. 24. Membership Functions for Output Variable $\dot{\theta}$—skid steering system.

The kinematics of a front-wheel steering vehicle are significantly different from those of a skid steering vehicle, and can be described as:

$$\dot{x} = v\,\cos\theta\,\cos\phi, \tag{8}$$

$$\dot{y} = v\,\sin\theta\,\cos\phi, \tag{9}$$

$$\dot{\theta} = \frac{v\,\sin\phi}{L}, \tag{10}$$

where ϕ is the angle between the main axis of the vehicle and the front wheels, and L is the length of the wheel-base of the vehicle.

The previously developed parallel parking system basically simulates the rules of a human driver for car-like front-wheel steering. Thus, the fuzzy control algorithm was expected to perform well with specifically tuned membership functions and scaling factors for the front-wheel steering AGV while using the same set of rules (This expectation was verified by simulation results as described below).

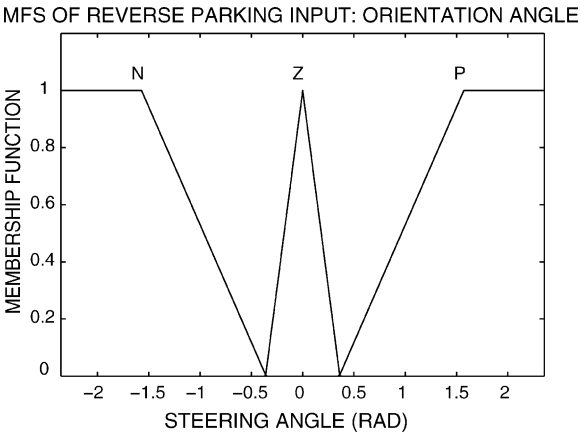

Fig. 23. Membership functions for input θ—skid steering system.

The fitness function chosen for front-wheel steering system was selected as $(x_a/l_p + y_d/h_p + |\theta| + \mathit{penalty})$. This fitness function is different from that of the skid steering system due to the different kinematics of the two types of systems.

The dimension of the vehicle used in the simulation has a length over width ratio of 2.3846, obtained from a Pontiac Sedan. Assume the length of the vehicle is 0.775 m, the same length as the ATRV-Jr, then the width of the vehicle is 0.325 m. A population size of 10 and 50 generations were used for the GA implementation. Figs. 25–28 show the membership functions obtained by GA optimization. The simulation results with the obtained membership functions and scaling factor clearly showed that a fuzzy logic system developed for a skid steering vehicle performed well for a front-wheel steering vehicle. If a GA was not used for the tuning, manual tuning with trial and error would have been necessitated.

5.4. Size of the parking space

The size of parking space has significant impact on the degree of difficulty of parallel parking. Jiang and

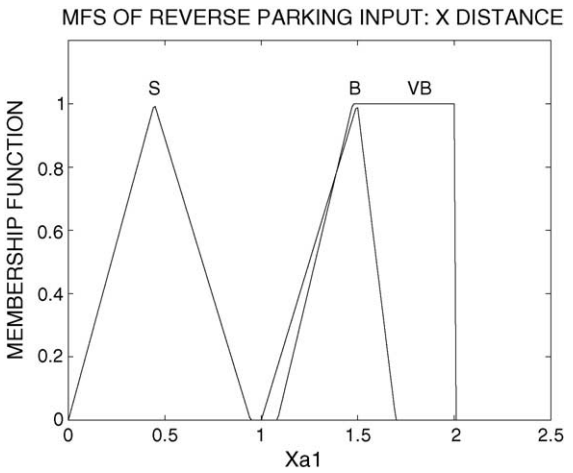

Fig. 25. Membership functions for input variable x_{a1}—front-wheel steering system.

Seneviratne [12] studied parallel parking in tight situations by path planning for a front-wheel steering vehicle. The minimum length of the parking space used was 100 cm with a vehicle length of 60 cm (this corresponds to 1.67 times the length of the vehicle). Considering the width of this vehicle is 34 cm, the ratio of length over width is 1.765.

It was verified by simulation that the larger the ratio of length over width, the easier the vehicle maneuvering into a parking space for front-wheel steering vehicles. To make the vehicles comparable, a front-wheel steering vehicle with the same ratio was simulated with the fuzzy logic parallel parking algorithm. Simulation results showed that the developed algorithm has the ability to park the vehicle into a space with length 1.4 times the length of a vehicle.

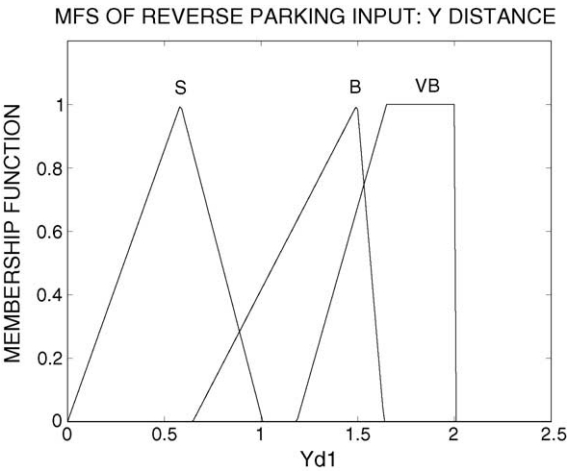

Fig. 26. Membership functions for input variable y_{d1}—front-wheel steering system.

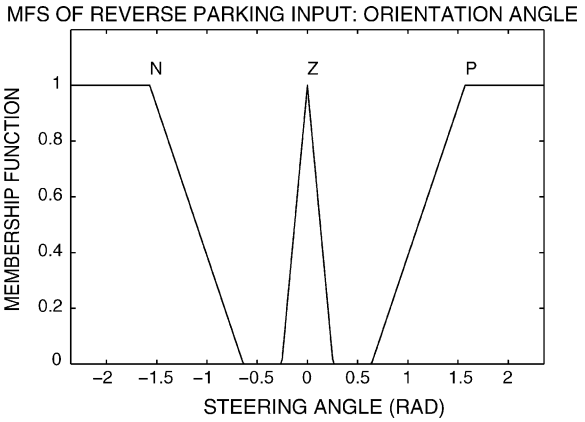

Fig. 27. Membership functions for input variable θ—front-wheel steering system.

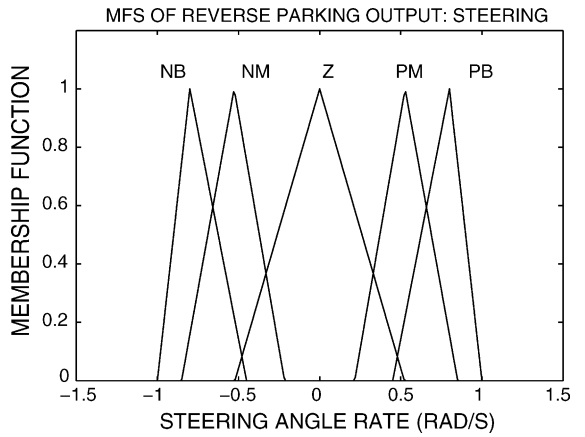

Fig. 28. Membership functions for output variable $\dot{\theta}$—front-wheel steering system.

6. Conclusions

This paper developed and tested a tight space, automatic parallel parking algorithm. The algorithm is described in detail and experimental results show the effectiveness of the controllers. In this paper, a genetic algorithm was also used for fine-tuning of the fuzzy

logic controllers. The developed genetic fuzzy system cannot only exempt human designers from the time-consuming manual tuning process, but also provides a means to transfer the developed algorithm to different vehicle platforms.

Disclaimers

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government.

References

[1] M.-R. Akbarzadeh-T, K. Kumbla, E. Tunstel, M. Jamshidi, Soft computing for autonomous robotic systems, Comp. Electrical Eng. 26 (1) (2000) 5–32.
[2] A. Arslan, M. Kaya, Determination of fuzzy logic membership functions using genetic algorithms, Fuzzy Sets Syst. 118 (2001) 297–306.
[3] D. Beasley, D.R. Bull, R.R. Martin, An overview of genetic algorithms. Part 1. Fundamentals, Univ. Comput. 15 (2) (1993) 58–69.
[6] W.A. Daxwanger, G.K. Schmidt, Skill-based visual parking control using neural and fuzzy networks, in: Proceedings of IEEE International Conference on Systems, Man and Cybernetics, Vancouver, BC, Canada, October 1995, pp. 1659–1664.
[7] D. Driankov, A. Saffiotti, Fuzzy Logic Techniques for Autonomous Vehicle Navigation, Physica-Verlag, Heidelberg, New York, 2001.
[8] P. Goel, S.I. Roumeliotis, G.S. Sukhatme, Robust localization using relative and absolute position estimates, in: Proceedings of IEEE International Conference on Intelligent Robots and Systems, Kyongju, South Korea, October 1999, pp. 1134–1140.
[9] F. Hoffmann, G. Pfister, Evolutionary design of a fuzzy knowledge base for a mobile robot, Int. J. Approximate Reasoning 17 (4) (1997) 447–469.
[10] R. Holve, P. Protzel, Reverse parking of a mobile car with fuzzy control, in: Proceedings of the 4th European Congress on Intelligent Techniques and Soft Computing, 1996, pp. 2171–2175.
[11] A. Homaifar, E. McCormick, Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms, IEEE Trans. Fuzzy Syst. 3 (2) (1995) 129–139.
[12] K. Jiang, L.D. Seneviratne, A sensor guided autonomous parking system for nonholonomic mobile robots, in: Proceedings of the 1999 IEEE International Conference on Robotics and Automation, Detriot, MI, May 1999, pp. 311–316.
[13] C. Laugier, T.H. Fraichard, P.H. Garnier, I.E. Paromtchik, A. Scheuer, Sensor-based control architecture for a car-like vehicle, Autonomous Robots 6 (1999) 165–185.
[14] R. Li, Y. Zhang, Fuzzy logic controller based on genetic algorithms, Fuzzy Sets Syst. 83 (1996) 1–10.
[15] S.H. Lian, Fuzzy logic control of an obstacle avoidance robot, in: Proceedings of the 5th IEEE International Conference on Fuzzy Systems, New Orleans, LA, September 1996, pp. 26–30.
[16] L. Magdalena, Adapting the gain of an FLC with genetic algorithms, Int. J. Approximate Reasoning 17 (4) (1997) 327–349.
[17] H. Miyata, M. Ohki, Y. Yokouchi, M. Ohkita, Control of the autonomous mobile robot DREAM-1 for a parallel parking, Math. Comput. Simul. 41 (1–2) (1996) 129–138.
[18] M. Ohkita, H. Miyata, M. Miura, H. Kouno, Travelling experiment of an autonomous mobile robot for a flush parking, in: Proceedings of the 2nd IEEE International Conference on Fuzzy Systems, San Francisco, CA, March 1993, pp. 327–332.
[19] I.E. Paromtchik, C. Laugier, Autonomous parallel parking of nonholonomic vehicle, in: Proceedings of the IEEE Intelligent Vehicles Symposium, Minneapolis, MN, April 1996, pp. 3117–3122.
[20] I.E. Paromtchik, C. Laugier, Motion generation and control for parking an autonomous vehicle, in: Proceedings of the 1996 IEEE International Conference on Robotics and Automation, Tokyo, Japan, September 1996, pp. 13–18.
[21] M.F. Selekwa, E.G. Collins Jr., Centralized fuzzy behavior control for robot navigation, in: Proceedings of the 2003 IEEE Symposium on Intelligent Control, Houston, TX, October 2003, pp. 602–607.
[22] R. Thrapp, C. Westbrook, D. Subramanian, Robust localization algorithms for an autonomous campus tour guide, in: Proceedings of the IEEE International Conference on Robotics and Automation, Seoul, South Korea, May 2001, pp. 2065–2071.
[23] K. Wiktorowicz, R. Zajdel, A fuzzy navigation of a mobile robot, Syst. Sci. 23 (4) (1997) 87–100.
[24] J. Xu, G. Chen, M. Xie, Vision-guided automatic parking for smart car, in: Proceedings of the IEEE Intelligent Vehicles Symposium, Dearbon, MI, 2000, pp. 725–730.
[25] J. Yen, N. Pfluger, A fuzzy logic based extension to Payton and Rosenblatt's command fusion method for mobile robot navigation, IEEE Trans. Syst., Man Cybernetics 25 (6) (1995) 971–978.
[26] New Jersey Driver Manual, New Jersey Department of Transporation, see http://www.state.nj.us/mvs/dm99/99ch3g.htm.
[27] ATRV-Jr All-Terrain Mobile Robot User's Guide, Robot Corp., Somerville, MA, 2001.

Yanan Zhao received the PhD degree in Mechanical Engineering from the Florida State University in 2001. She worked for three years as a Research Associate in the Department of Mechanical Engineering at the Florida A&M University—Florida State University College of Engineering. Her research interests include intelligent control systems for autonomous vehicles, automated controller tuning, fluidic thrust vector control, and evolutionary methods for control design.

Dr. Collins received the PhD degree in Aeronautics and Astronautics from Purdue University in 1987. He worked for seven years in the Controls Technology Group at Harris Corporation, Melbourne, FL before joining the Department of Mechanical Engineering at the Florida A&M University—Florida State University College of Engineering, Tallahassee, FL, where he is currently the John H. Seely Professor. His current research interests include intelligent control systems for autonomous vehicles, robust fault detection and isolation, control in manufacturing, automated controller tuning, automated weight selection in modern control, and fluidic thrust vector control. Dr. Collins is the author of over 160 technical publications in controls and robotics. He served as an associate editor of the IEEE Transactions on Control System Technology from 1992 to 2001. In 1991 he received the Honorary Superior Accomplishment Award from NASA Langley Research Center for "contributions in demonstrating active control of flexible spacecraft."