Fuzzy Logic Control of Autonomous Vehicles for Parallel Parking Maneuver

by

Yanan Zhao and Emmanuel G. Collins, Jr.

Department of Mechanical Engineering
Florida A&M University - Florida State University, Tallahassee, FL
yzhao@eng.fsu.edu, ecollins@eng.fsu.edu

Abstract

In this paper, the automatic parallel parking problem is first described and then fuzzy logic controllers are developed for each step of the parking process. The focus of the research is to develop fuzzy controllers that can park vehicles in tight spaces. The whole parking algorithm is simulated based on the model of a skid steering autonomous ground vehicle. The simulation results under a variety of scenarios illustrate the effectiveness of the developed controllers. The performance of the developed controller is also demonstrated by experimental implementation on an ATRV-Jr mobile robot.

Prepared through collaborative participation in the Communications and Networks 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.

1. Introduction

This research is aimed at developing an intelligent reverse-motion maneuvering controller for military autonomous ground vehicles (AGVs) in tight spaces; the objective is to hide AGVs during missions, for example between trees or in crevices or small buildings, hence preventing them from being detected or attacked. To develop a maneuvering algorithm that applies to most if not all vehicles, the reverse-motion maneuvering is designed to emulate the parallel parking process of an experienced human driver.

The automatic parallel parking problem has attracted a great deal of attention among researchers. Current approaches to solving this problem can be classified into two main groups: 1) the path tracking 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 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 tracking Paromtchik and Laugier [1, 2, 3] 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 [4] 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, Chen and Xie [5] planned a quintic polynomial curve for the reference path, where the steering angle was obtained by the instant turning radius of the vehicle. Cheng, Chang and Li [6] also used a fifth-order polynomial curve as the reference path, and a fuzzy logic technique was employed to follow the path.

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 [7]. In the fuzzy logic approach of Miyata, Ohkita etc. and Holve [8, 9, 10], 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 [8, 9].

The path tracking approach is model-based. In particular both the planning and following of the reference path rely on the environmental model and the dynamic model of the vehicle. However, the prior knowledge about the environment is in general incomplete, uncertain, and approximate, and the effect of control actions is not completely reliable, for example, wheels may slip. A robust control algorithm is desired to reliably perform complex tasks in spite of uncertainties and control errors.

Fuzzy logic has found many successful applications in the domain of autonomous vehicle navigation [6, 8, 9, 10, 11, 13, 14]. Due to the nature of fuzzy logic, no precise mathematical model of the vehicle or its environment is needed for fuzzy logic based navigation. Also fuzzy control is robust with respect to errors in sensor data and to fluctuations in the dynamics of the system and environment. In addition fuzzy control can often be transferred from one platform to another with few modifications. Further, fuzzy logic navigation allows various behaviors to be easily combined through a command fusion process [11, 12]. To take advantage of the above features, fuzzy logic is used here for the parallel parking control.

The size of the available parking space has significant impact on the degree of difficulty of parallel parking maneuvering. The papers [8, 9, 10] do not explicitly discuss the size of the maneuvering space considered. However, it is observed that in [8] and [9], the size of the maneuvering space used was twice the length of the vehicle and 2.7 times the width of the vehicle. (The relative dimensions of the space and vehicle in [10] are unclear.) 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.

This paper is organized as follows. Section 2 describes the automated parking process. Section 3 develops the fuzzy logic controllers. Section 4 illustrates the simulation and experimental results. Finally, Section 5 presents some conclusions.

2. Automated Parking Process

In an effort to use human knowledge and experience most efficiently in the design of the controller, the parking process was divided into three steps and a fuzzy controller was designed for each of the steps. The three steps are: 1) parking space scanning while reaching a ready to reverse

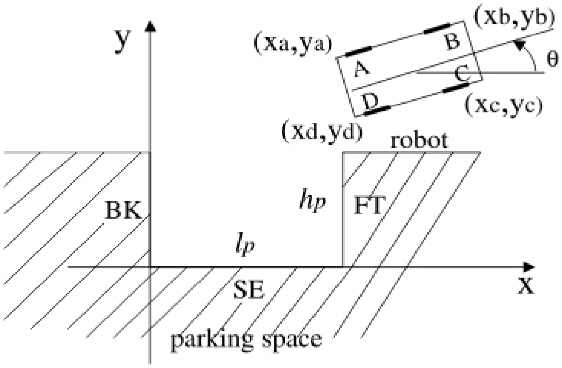


Figure 1: Maneuvering Space and the Local Coordinate System

position, 2) reversing the vehicle into the parking space, and 3) adjusting the vehicle forward inside the parking space.

In the first step, the vehicle is navigated forward to reach a ready-to-reverse position with the vehicle's orientation parallel to the available space. The parking space is also scanned using either image sensors or ultrasonic sensors. During this step, the vehicle moves slowly to pass the parking space, builds a local map of the environment, and detects obstacles [3, 4, 5].

The detected space can be described as shown in Figure 1. 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. To focus the study on the controller design part, it is assumed below that the available space is detected and to the right of the vehicle.

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 at the same time requires a complex fuzzy system. Hence, to yield a simpler and more easily implementable algorithm, this step is divided into two substeps.

The task of the first substep is to have the vehicle reach an intermediate position ($(0.9l_p, h_p + 0.65b)$ for the vehicle's center) without considering the orientation angle of the vehicle. The desired y position of the vehicle is reached at this stage.

In the second substep, 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.

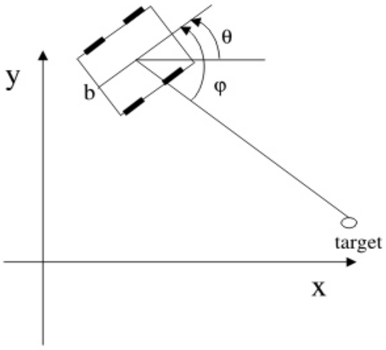


Figure 2: The Vehicle Kinematics and Heading Angle Difference

In the second step of parallel parking, the vehicle is first backed up into the parking space with an increasing θ until its right rear wheel is at a certain distance from the boundary SE of the space. Then the vehicle is backed up 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. 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 with some tolerance.

3. Fuzzy Logic Controllers (FLCs)

The ATRV-Jr is the autonomous vehicle considered here. It maneuvers using 4-wheel skid steering, i.e., the wheels on one side of the ARTV-Jr are mechanically coupled, and the left and right side's wheels can take different velocities. To easily generalize the research results to vehicles with front-wheel steering, the outputs of all the fuzzy logic control systems proposed here are taken as the steering angle rate. An alternative is to have the fuzzy controller output the left and right wheel velocities for a skid steering system.

3.1. Reaching a Ready-to-Reverse Position

This first step is composed of a goal-seeking behavior followed by an orientation adjusting behavior. The input of the goal-seeking behavior is the heading angle difference ϕ, which is the angle between the line connecting the vehicle's center to the target and the heading direction of the vehicle as shown in Figure 2. The output of the goal-seeking fuzzy controller is the steering rate of the vehicle $\dot{\theta}$. Here θ is the angle between the positive x axis and the main axis of the vehicle. The membership functions for both the input and output are illustrated in Figure 3, where fuzzy sets N, P and Z represent negative, positive and zero respectively.

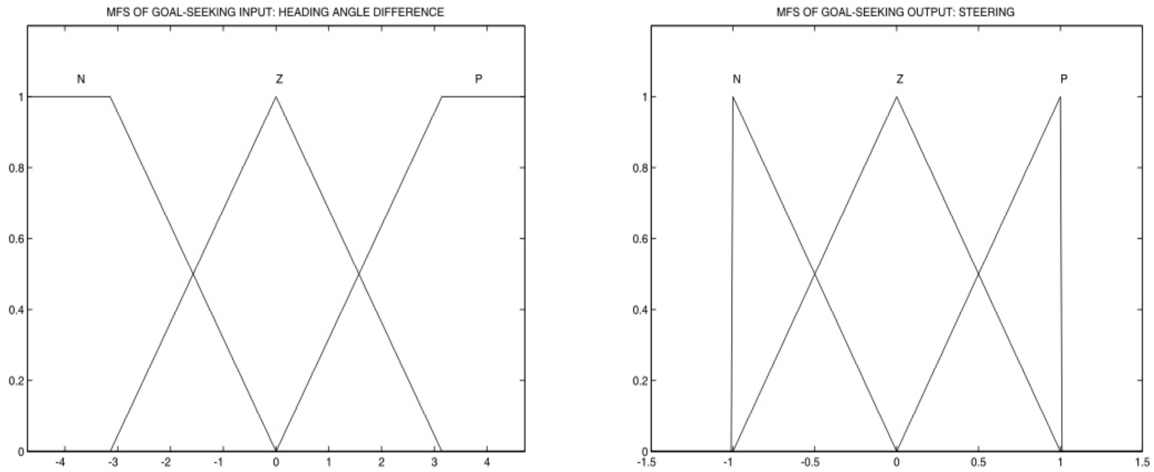


Figure 3: Membership Functions for the Goal-seeking Input ϕ (left) and Output $\dot{\theta}$ (right)

The fuzzy inference rules for the goal-seeking behavior are listed in Table 1. The angle ϕ is positive when the line connecting the center of vehicle and target rotates counterclockwise to the heading direction (as shown in Figure 2), while $\dot{\theta}$ is positive when the right wheels' velocity is bigger than the left wheels' velocity. The basic notions 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.

ϕ	N	Z	P
$\dot{\theta}$	P	Z	N

Table 1: Fuzzy Rules for Goal-seeking

The objective of the orientation adjusting 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 logic 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 Figure 4, where fuzzy sets NB, NM, Z, PM and PB represent negative big, negative medium, zero, positive medium 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.
- If the steering angle is zero, the steering angle rate should be zero.

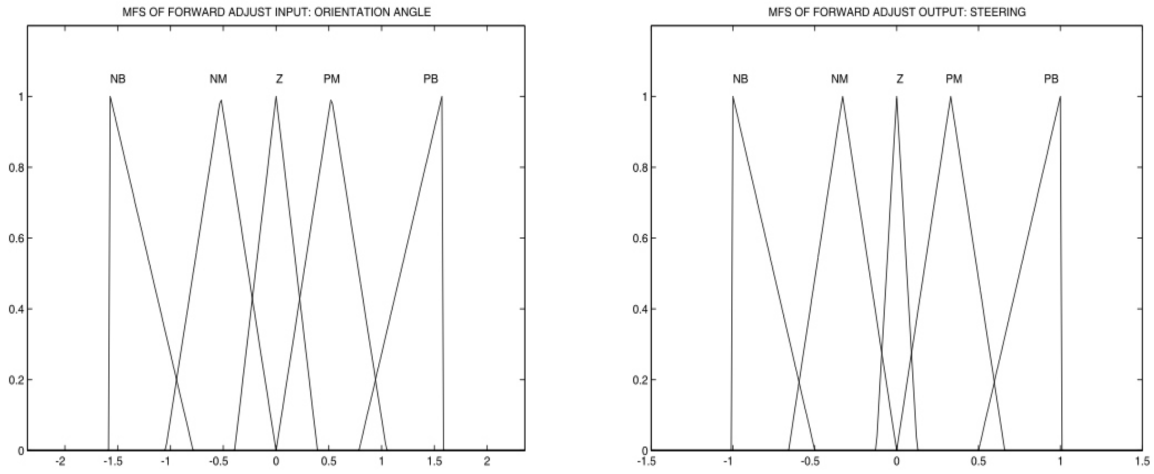


Figure 4: Membership Functions for Orientation Adjustment Input θ (left) and Output $\dot{\theta}$ (right)

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

Table 2: Fuzzy Rules for Orientation Adjustment

3.2. Reversing the Vehicle Into the Maneuvering Space

This reverse maneuvering requires a more complex fuzzy logic controller. This step has two basic sequential goals: 1) back up the vehicle while increasing the orientation angle until the vehicle is very close to the boundary SE of the parking space, and 2) then back up the vehicle while decreasing the angle.

The fuzzy logic controller is developed based on an expert human driver's knowledge. The position of the vehicle relative to the parking space and the vehicle's orientation are used to generate the control command. Here, two variables x_{a1} and y_{d1} are defined to represent the relative position, having $x_{a1} = x_a/l_p$ and $y_{d1} = y_d/h_p$, where x_a and y_d are the coordinate of the left and right rear corner of the vehicle in the local coordinate system, as shown in Figure 1.

The fuzzy logic controller has three inputs, x_{a1}, y_{d1} and the orientation angle θ. The output of the fuzzy controller is the steering rate $\dot{\theta}$. Figures 5 to 6 show the proposed membership functions, where S, B, VB represent small, big and very big respectively. The three dimensional fuzzy rules are shown in Tables 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 available space or it has entered one of the shaded regions in Figure 1). 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 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 back up 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.

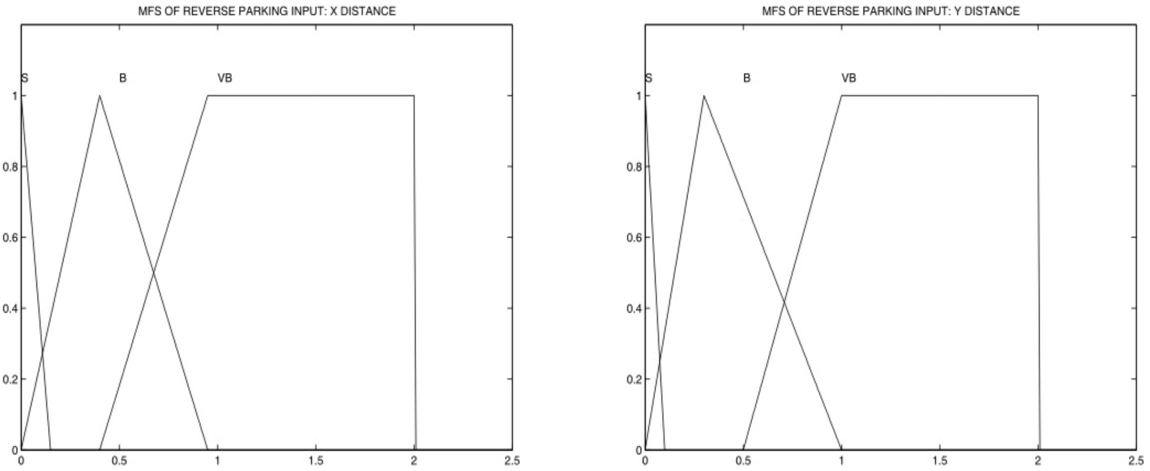


Figure 5: Membership Functions for the Backing Up Input x_{a1} (left) and y_{d1} (right)

3.3. Adjusting the Vehicle Forward in the Space

The task is to adjust the orientation of the vehicle while simultaneously move it forward. This is essentially the same task as that of the orientation adjustment described in the first step. Thus

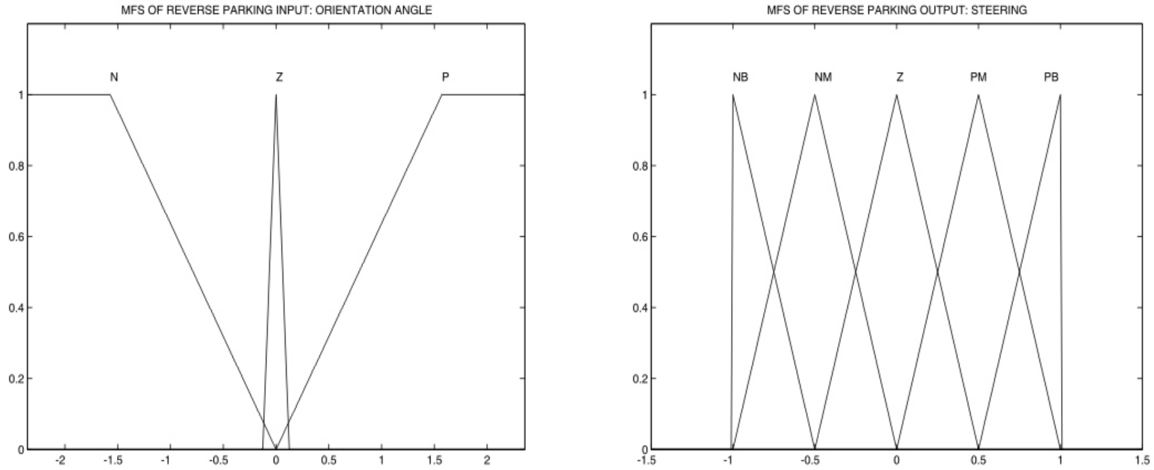


Figure 6: Membership Functions for the Backing Up Input θ (left) and Output $\dot{\theta}$ (right)

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

Table 3: Fuzzy Rules for the Backing Up Step

the orientation adjustment fuzzy controller of the first step is also used for this task.

Significant tuning of the proposed fuzzy controllers was conducted to yield a parking algorithm that fits AGVs into tight spaces. This includes tuning of the fuzzy rules, membership functions, and input and output scaling factors. The parking space used in the research is 1.4 times the length of vehicle and 1.2 times the width of the vehicle.

4. Simulation and Experimental Implementation

Simulations were first used to test the control algorithms. Then, the algorithms were tested experimentally on an ATRV-Jr.

4.1. Simulation Results

Extensive simulations were conducted for the above algorithms using Matlab and the Fuzzy Logic Toolbox [15]. The simulations were based on the kinematic model of the ATRV-Jr, which can be described as [16]:

$$
\begin{aligned}
\theta(i+1) &= \theta(i) + \dot{\theta}(i)dt, && (1)\\
x(i+1) &= x(i) + v(i+1)\cos(\theta(i+1))dt, && (2)\\
y(i+1) &= y(i) + v(i+1)\sin(\theta(i+1))dt, && (3)
\end{aligned}
$$

where $v(i) = \frac{v_r(i)+v_l(i)}{2}$, $\dot{\theta}(i) = \frac{v_r(i)-v_l(i)}{b}$, (x, y) denotes the coordinate of the center of the vehicle in the local coordinate system defined previously, v denotes the total translational velocity, v_r and v_l denote the velocity of the right side and left side wheels respectively. The algorithm was seen to always successfully maneuver the vehicle from any initial position if the desired ready-to-reverse position can be reached (which is true when the initial x position satisfies $x \leq 0.9l_p$).

Here an example is given with the initial position $(x, y) = (-l_p, h_p + 1.5b)$. Figures 7 to 10 show the whole parking process under a parking size of $1.2b \times 1.4l$. It was seen that the vehicle moves back and forth twice within the parking space to reach the desired position. The tolerance for the desired final parking position is defined by the designer. If a tighter tolerance is desired, more maneuvering will be required to reach the final position.

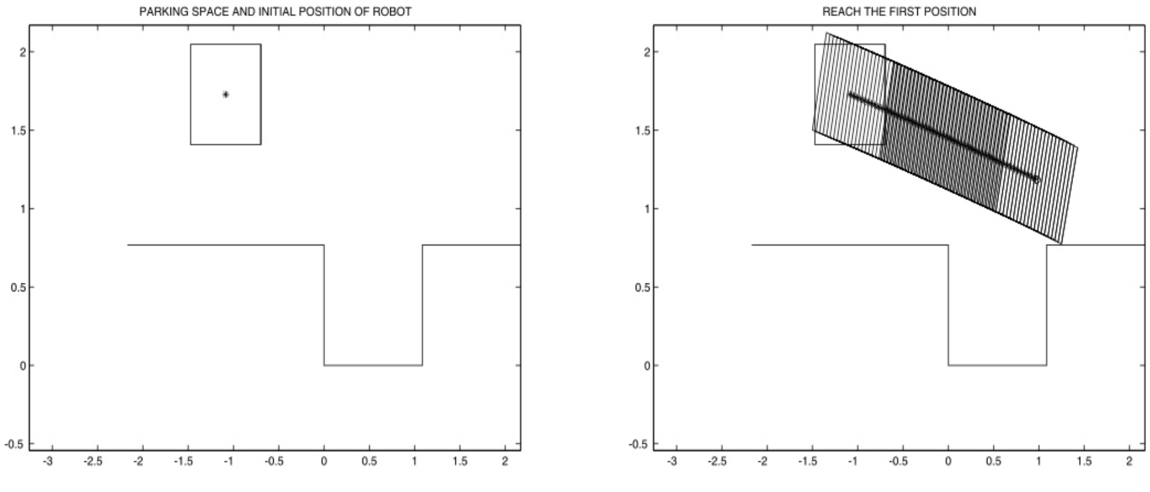
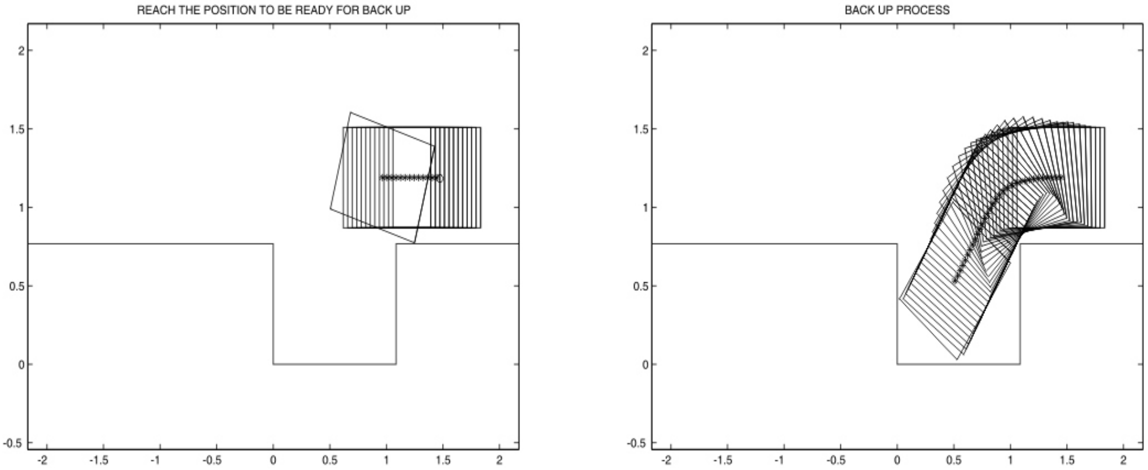


Figure 7: Initial Position (left) and Navigation to Ready-to-Reverse Position (right)

For a detected parking space which is larger than the above one $(1.2b \times 1.4l)$, we can either fit the smaller parking space into it, or use the larger space directly. (The latter option is more efficient, since it takes advantage of the additional space). Figures 11 through 12 illustrate the


REACH THE POSITION TO BE READY FOR BACK UP

BACK UP PROCESS

Figure 8: Adjustment to Ready-to-Reverse Position (left) and Backing Up Into the Parking Space (right)

parking process with a parking size of $1.2b \times 2l$ (assuming the initial position of the vehicle is the same as that of $1.2b \times 1.4l$).

4.2. Experimental Results

The algorithm was further verified by experimental implementation on an ATRV-Jr. The fuzzy logic controllers were programmed in C++ and the code was uploaded to the onboard PC of the ATRV-Jr. The output scaling factors of the fuzzy controllers were further tuned during the experiment to adapt to the real dynamics of the vehicle. The odometry information from wheel encoders was used for vehicle localization. The translation speed used in the experiment was 8 cm/s. Figures 13 to 18 show the maneuvering process of the ATRV-Jr from an initial position till the final position was reached. Experiments conducted under different initial positions achieved similar results.

5. Conclusions

In this paper a solution for parallel parking of autonomous ground vehicles was developed. The algorithm, which uses a fuzzy logic controller for each of the parking steps was described in detail. The simulation and experimental implementation results illustrate the effectiveness of the parallel parking scheme to maneuver the vehicle into tight enclosures.

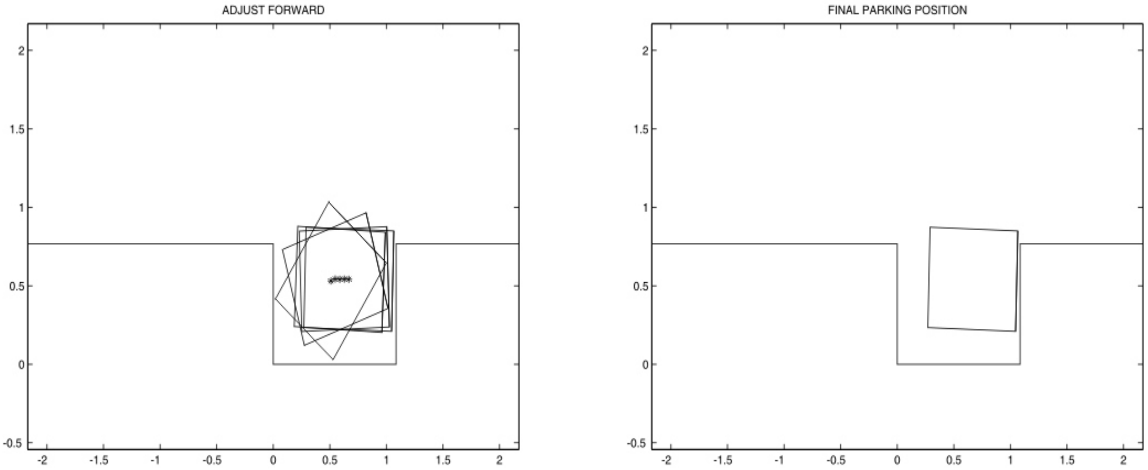


Figure 9: Adjusting Forward (left) and Position First Reached (right)

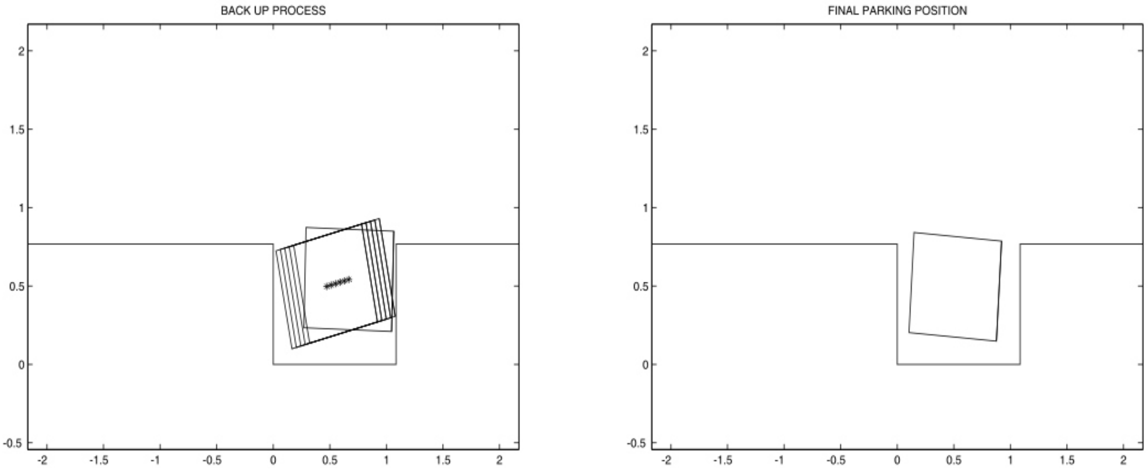


Figure 10: Backing Up Again (left) and Final Position Reached by Adjusting Forward Again(right)

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.

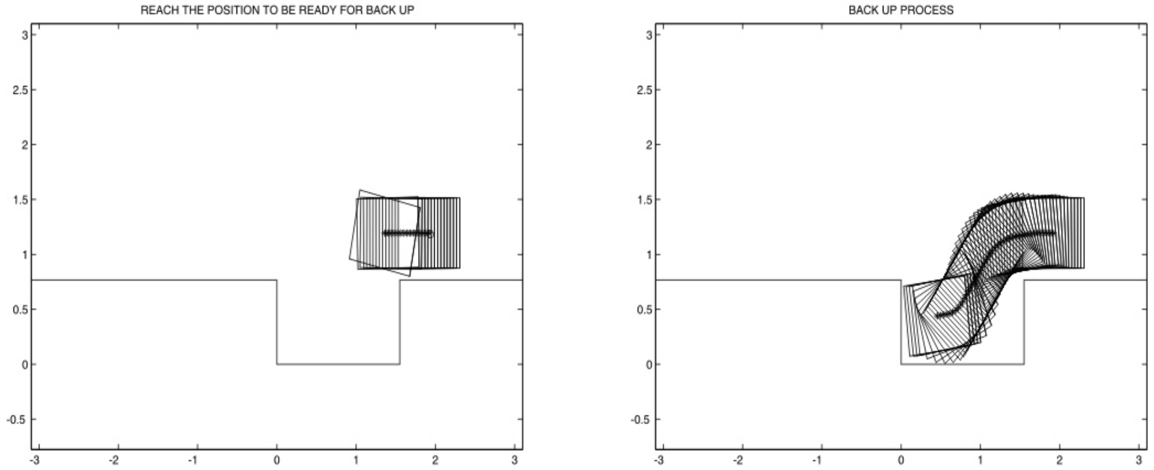


Figure 11: Adjustment to Ready-to-Reverse Position (left) and Backing Up Into the Parking Space (right) with $1.2b \times 2l$ Parking Space

References

[1] I. E. Paromtchik and C. Laugier, "Autonomous Parallel Parking of Nonholonomic Vehicle," *Proceedings of the IEEE Intelligent Vehicles Symposium*, Minneapolis, MN, pp. 3117-3122, April 1996.

[2] I. E. Paromtchik and C. Laugier, "Motion Generation and Control for Parking an Autonomous Vehicle," *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, Tokyo, Japan, pp. 13-18, September 1996.

[3] C. Laugier, TH. Fraichard, PH. Garnier, I.E. Paromtchik and A. Scheuer, "Sensor-based Control Architecture for a Car-like Vehicle," *Autonomous Robots*, Vol. 6, pp. 165-185, 1999.

[4] K. Jiang and L. D. Seneviratne, "A Sensor Guided Autonomous Parking System for Nonholonomic Mobile Robots," *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, Detriot, MI, pp. 311-316, May 1999.

[5] J. Xu, G. Chen and M. Xie, "Vision-Guided Automatic Parking for Smart Car," *Proceedings of the IEEE Intelligent Vehicles Symposium*, Dearbon, MI, pp. 725-730, 2000.

[6] C. Cheng, S. Chang and T. Li, "Parallel-Parking Control of Autonomous Mobile Robot," *Proceedings of the 23rd Annual International Conference on Industrial Electronics, Control, and Instrumentation*, New Orleans, LA, pp. 1305-1310, November 1998.

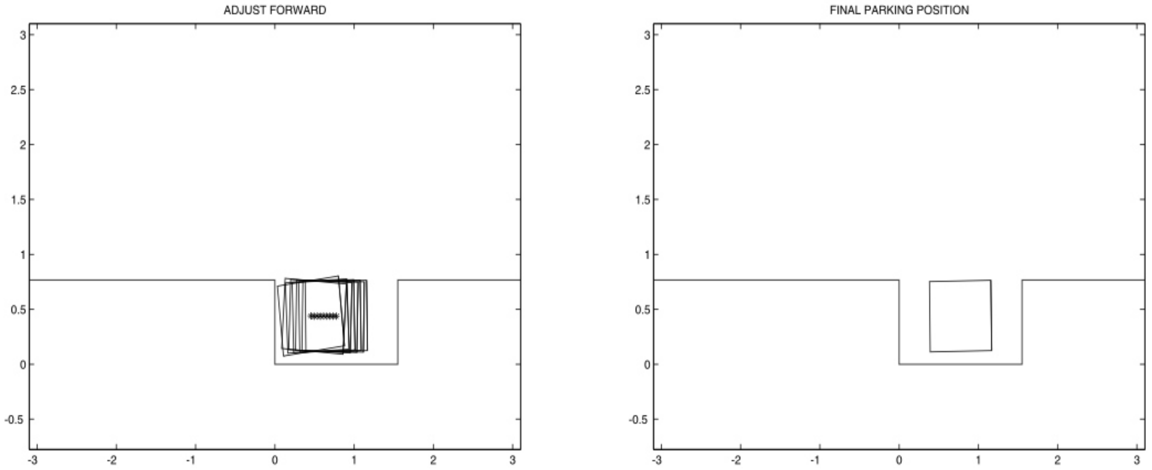


Figure 12: Adjusting Forward (left) and Final Position (right) with $1.2b \times 2l$ Parking Space

Figure 13: Initial Position

Figure 14: Passing By the Space

[7] W. A. Daxwanger and G. K. Schmidt, "Skill-Based Visual Parking Control Using Neural and Fuzzy Networks," *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, Vancouver, BC, Canada, pp. 1659-1664, October 1995.

[8] H. Miyata, M. Ohki, Y. Yokouchi and M. Ohkita, "Control of the Autonomous Mobile Robot DREAM-1 for a Parallel Parking," *Mathematics and Computers in Simulation*, Vol. 41, No. 1-2, pp. 129-138, 1996.

[9] M. Ohkita, H. Miyata, M. Miura and H. Kouno, "Travelling Experiment of an Autonomous Mobile Robot for a Flush Parking," *Proceedings of the 2nd IEEE International Conference on Fuzzy Systems*, San Francisco, CA, pp. 327-332, March 1993.

[10] R. Holve and P. Protzel, "Reverse Parking of a Mobile Car with Fuzzy Control," *Proceedings of the 4th European Congress on Intelligent Techniques and Soft Computing*, pp. 2171-2175, 1996.

Figure 15: Ready to Reverse Position

Figure 16: Back Up Maneuvering

Figure 17: Adjust Forward Maneuvering

Figure 18: Final Position Reached

[11] D. Driankov and A. Saffiotti, *Fuzzy Logic Techniques for Autonomous Vehicle Navigation,* Physica-Verlag Heidelberg, New York, 2001.

[12] J. Yen and N. Pfluger, "A Fuzzy Logic Based Extension to Payton and Rosenblatt's Command Fusion Method for Mobile Robot Navigation," *IEEE Trans. on Systems, Man and Cybernetics*, Vol. 25, No. 6, pp. 971-978, 1995.

[13] K. Wiktorowicz and R. Zajdel, "A Fuzzy Navigation of a Mobile Robot," *Systems Science*, Vol. 23, No. 4, pp. 87-100, 1997.

[14] S. H. Lian, "Fuzzy Logic Control of an Obstacle Avoidance Robot," *Proceedings of the 5th IEEE International Conference on Fuzzy Systems*, New Orleans, LA, pp. 26-30, September 1996.

[15] The MathWorks Inc., *Fuzzy Logic Toolbox, for use with MATLAB*, The MathWorks Inc., Natick, MA, 1998.

[16] P. Goel, S. I. Roumeliotis and G. S. Sukhatme, "Robust Localization Using Relative and Absolute Position Estimates," *IEEE International Conference on Intelligent Robots and Systems*, Kyongju, South Korea, pp. 1134-1140, October 1999.