

## An Intelligent Auto Parking System for Vehicles

Anastasia Razinkova<sup>1</sup>, Hyun-Chan Cho<sup>1\*</sup>, Hong-Tae Jeon<sup>2</sup>

<sup>1</sup> Department of Electrical and Electronics Engineering, KoreaTech, Cheonan, 330-708, South Korea

<sup>2</sup> Department of Electrical and Electronics Engineering, Chung-Ang University, Seoul, 156-756, South Korea

### Abstract

Autoparking assistant systems are a new and very promising area in automotive systems engineering. Since the traffic in modern cities becomes more intense, it is getting harder for a driver. Those systems are necessary for an inexperienced one to find a proper parking slot, or to park in a narrow parking slot without damaging his car or the vehicles around. The implementation of autoparking assistant systems may reduce drivers' stress and make parking generally more comfortable. In addition, such system can be extremely useful for senior or disabled people or for drivers with reduced mobility. The implementation of autoparking assistant systems may increase the safety of the parking, and therefore the development of such systems is a highly-demanded task.

We introduce an intelligent autoparking system that automatically generates trajectory for parking using a fuzzy logic. This paper consists of three parts. In first part we introduce trajectory generation method for parallel parking without collisions. Fuzzy-logic based trajectory generation algorithm is described in second part. Experimental results presented in the third part of the paper prove effectiveness of the proposed method.

**Keywords :** Autoparking assistance system, fuzzy logic, intelligent systems

### 1. Introduction

Parallel parking is a method of the parking in line with other parked cars. Cars parked in parallel are in one line, to the curb, with the front of each car facing the back bumper of the adjacent one. Parallel parking requires driving the car in reverse gear into the parking space.

Parallel parking is considered to be one of the hardest skills for new drivers to learn. Parallel parking allows a vehicle to park in a smaller space than would be true of forward parking. The automatic parking aims to enhance the comfort and safety of driving in constrained environments where much attention and experience is required to steer the car. The parking maneuver is achieved by means of coordinated control of the steering angle and speed which takes into account the actual situation in the environment to ensure collision-free motion within the available space.

First ideas of automatic parking systems were issued in the middle of 80-th. Sugeno M., K. Murakami [1] proposed a fuzzy algorithm for garage parking system. A. Ohata and M. Mio [2] extended Sugeno's work by introducing vehicle dynamical model during parking and one of the first nonlinear algorithms for parking control. In 1996, I. E. Paromtchik, and C. Laugier proposed main statements for parallel parking problem and parking maneuver methods [3] with repeated backward and forward motions. The research works in [4-5] are also applied

in the simulation results on the parallel parking. Despite of the efficiency of the generated path, it is not clear for the practical implementation as the centers of the circles are not determined by a practical parking scenario, and the parking lot dimensions. Basic parking assistant system (PAS) architecture is given in Fig 1. At first, the sensors are used for receiving information about the environment (Part I in Fig 1). It measures the distance to the obstacles, current car velocity and detects parking space. The next step after receiving data is map building and current relative vehicle position estimation (Part II, a). Based on map building the system checks the possibility of parking the car. If the parking is considered possible, the car should stop so that the control block can generate the parking trajectory using a particular algorithm. Trajectory generation algorithm generates desired trajectory and converts it to desired law for steering angle (Part II, b). Path tracking control part is using the desired law for steering angle generated on the previous step as an input to calculate desired position of the car at a given time sample (Part II, c). Data from steering angle sensor and wheel speed sensor is used for position estimation based on the dynamic model of the car.

Next, the position of vehicle is changed by the means of changing steering angle, which is controlled by steering motor (Part III). A lot of modern cars have electric motors as a part of steering system to ease steering control for a human driver; these motors can be used in parking systems too. Tracking controller manages the steering motor motion, depending on the angle, speed and time parameters of steering law. Part IV is responsible for graphic user interface, since every parking system is working together with driver.

---

Manuscript received Jun. 12, 2012; revised Sep. 23, 2012; accepted Sep. 24, 2012

\*Corresponding Author: cholab@kut.ac.kr

This paper was supported by the Education and Research Promotion Program of KOREATECH.

© The Korean Institute of Intelligent Systems. All rights reserved.

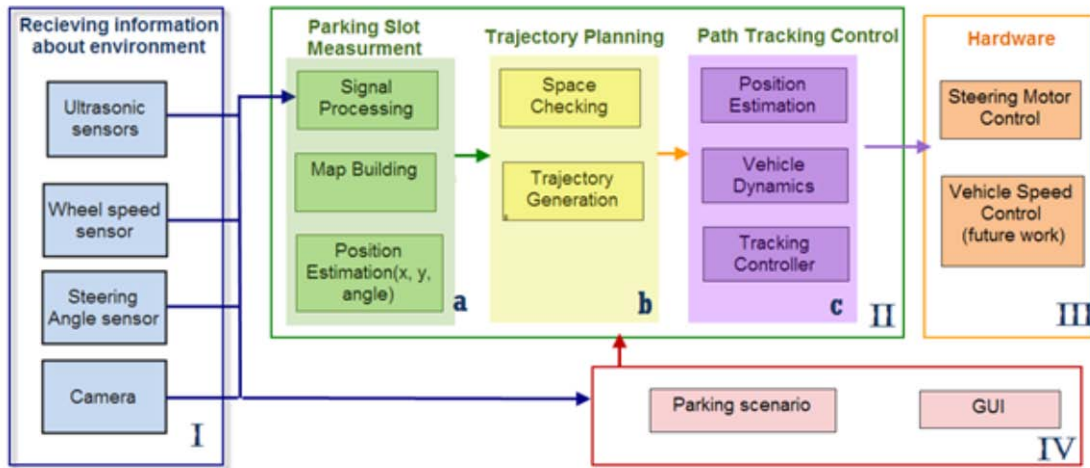


Fig.1 Basic Architecture of Parking Assistant System

Despite of the fact that there are several manufactured systems exist, they still have some disadvantages such as sensitivity to non-ideal environment, slow parking speed, the need in expensive and vehicle, reliable sensors and require further improvement. Since man can usually perform parallel parking easily after an appropriate experience without deep understanding about the vehicle then an intelligent scheme such as fuzzy logic controller will be effective to achieve an automated parallel parking [6].

In this paper, a novel parking assistant system using fuzzy logic is presented. In the proposed system geometric constraints are directly considered to design a collision-free trajectory, therefore system does not need to recursively detect collisions while parking. Also usage human-like intelligent fuzzy logic-based algorithm allows choosing trajectory that is the optimal for current environment conditions, that improves parking performance. The trajectory generation algorithm used in the developed parking assistance system is able to compute the direct motion into parking lot where other methods[1] require the multiple maneuvers.

The rest of the paper is organized as follows. In Section 2 we give the details on the collision-free trajectory generation for parallel parking. The design of the proposed intelligent trajectory generation system is discussed in Section 3. Simulation and experimental results are described in Section 4, and conclusions are drawn in Section 5.

## 2. Trajectory generation for parallel parking problem

### 2.1. Vehicle kinematics model

Wheeled vehicles have been used in many autonomous vehicle applications. The kinematics of these types of vehicles was described in [7].

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ v(t)\rho(t) \end{bmatrix} \quad (1)$$

The kinematics of a car having single-track model with axle base  $l$  described by Equation 1 (with an assumption that velocity is small). There  $(x,y)$  are the coordinates of the reference point  $i$  (the midpoint of the rear axle in a Euclidean reference frame), angle  $\theta$  means the orientation of the car relative to the  $x$ -axis,  $v(t)$  is the linear velocity of point  $i$ , and  $\rho(t)$  is the curvature of the path described by point  $i$ .

The curvature radius for any path can be written as

$$R(t) = \frac{1}{\rho(t)} = \frac{l}{\tan \varphi} \quad (2)$$

Where  $l$  is the distance between the front and back wheels (axle base), and  $\varphi$  (namely the steering angle) is the angle defined by the main axis of the vehicle and the central front point (see Fig. 2).

In the cars the value of  $R$  is usually bounded by  $R_{\min}$ , the minimum curvature radius, restricted by maximum steering angle for certain car.

### 2.2. Desired reference trajectory for parallel parking

Assume that a collision-free start configuration  $q_0$  and a collision-free goal configuration  $q_f$  are given. Find a desired trajectory from  $(x_0, y_0, \theta_0)$  to  $(x_f, y_f, \theta_f)$  using following

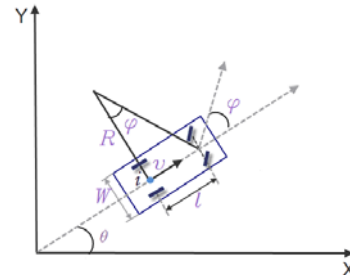


Fig. 2. General kinematics of a vehicle

constraints:  $v(t)$  is piece-wise constant and equal to some unit value, trajectory  $q(t)$  satisfy non-collision conditions for anytime.

The desired trajectory is shown on Fig. 3, car starts parking by driving straight from point  $(x_0, y_0)$  to point  $(x_s, y_s)$  (straight segment  $s_{st}$ ), next at point  $(x_s, y_s)$  it changes steering angle and moves along arc  $s_1$  of radius  $R_1$ , car changes direction and value of steering angle at point  $(x_t, y_t)$  and drive along another arc segment  $s_2$  of radius  $R_2$  until it stops at final point  $(x_f, y_f)$ .

The parameters  $s_{st}, s_1, s_2$  represent the length of the segments [8]:

$$s_1 = R_1\theta_1 \quad s_2 = R_2\theta_2 \quad (3)$$

$$s_{st} = x_{in} - (R_2 + R_1)\sin\theta = x_{in} - L_p$$

where  $L_p$  – displacement to point  $(x_s, y_s)$  when straight segment ends (named up-to-straight displacement below).

It can be proved that angle  $\theta$  can be defined as:

$$\theta = a \cos\left(l - \frac{y_0}{R_1 + R_2}\right) \quad (4)$$

For given  $y_0$ , infinite number of possible maneuvers can be found, each of them determined by a pair of  $R_1$  and  $R_2$ . As  $L_p$  represents the up-to-straight displacement, the relation between  $R_1$  and  $R_2$  can be obtained as:

$$R_2 + R_1 = \frac{L_p^2 + y_0^2}{2y_0} \quad (5)$$

It means that, for a chosen  $L_p$ , all possible trajectories are determined by  $R_1$  and  $R_2$  which satisfy expression above.

### 2.3. Non-collision conditions for parallel parking trajectory generation

Assume rectangular models of the vehicles and obstacles. Let  $R_{min}$  be the minimum curvature radius of the vehicle. Assume that a parking to the right is considered, as illustrated in Fig. 4 Initial statement for parallel parking. In order to park the vehicle, a restricted maneuver has to be designed in way described above.

In order to design collision free trajectory information about geometry of vehicle and environment is given (see Fig. 4). After desired trajectory is defined, our main concern is what kind of

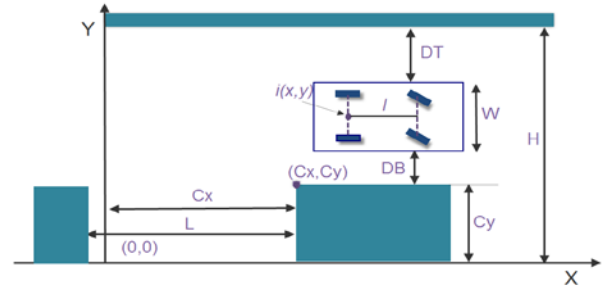


Fig. 4 Initial statement for parallel parking

collisions may occur and how radiuses  $R_1$  and  $R_2$  will be bounded by them [8].

There are three possible collisions during parking (Fig. 5).

Collision of the first kind may occur when a vehicle contacts an obstacle in front of it while passing  $s_2$ , is shown in Fig. 5 (a). In addition, the vehicle may have collision with obstacle while moving along the path  $s_1$  (see Fig. 5 (b)). The last possible collision kind is collision with the wall or line of vehicles opposite to the parking slot (see Fig.5 (c)).

It can be proved that the requirements for avoiding those three collisions will give us following bound for radiuses  $R_1$  and  $R_2$  and the minimum and maximum up-on-straight displacement [8]:

$$R_{2max} = \left| \frac{C_x^2 + C_y^2 - l^2}{2C_y} \right| \quad (6)$$

$$R_{1max}(R_2) = \left| \frac{y_0}{1 - \cos\theta_m} - R_2 \right| \quad (7)$$

$$R_{1min} = \left| \frac{W^2 + l^2 - (H^2 - y_0^2)}{2(H - y_0 - W)} \right| \quad (8)$$

$$L_{pmin} = \sqrt{(R_{min} + R_{1min})2y_0 - y_0^2} \quad (9)$$

$$L_{pmax} = \sqrt{(R_{min} + R_{1max})2y_0 - y_0^2} \quad (10)$$

where  $\theta_m$  - the value of when the path is tangent to the point C  $(C_x, C_y)$ ,  $[L_{pmin}; L_{pmax}]$  is a range of points where the vehicle could stop and perform a obtained collision-free maneuver. One more margin for radiuses is minimum curvature radius.

All those conditions are represented graphically in Fig. 6, a bounded region of the maneuver solution set is obtained. The filled region is the collision-free area. Each point belonging to this region represents a free parallel parking maneuver. Furthermore, for a given forward displacement  $L_p$ , the collision-free area is reduced to a straight line (see Fig 6). An operating point on this line usually is chosen by driver based on his experience. Therefore fuzzy-logic algorithm is proposed for performing this task, it incorporates both driver experience and calculated non-collision boundaries for radiuses.

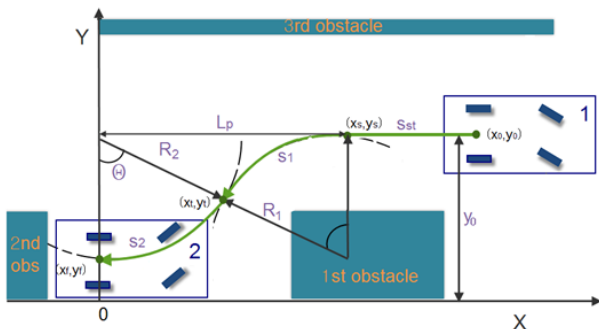


Fig. 3. Parameters of parallel parking

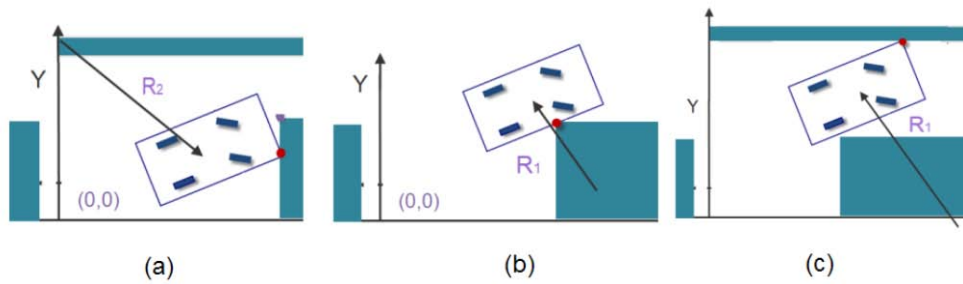


Fig. 5 Possible collisions during parking

### 3. Design of intelligent trajectory generation system

#### 3.1. Trajectory generation algorithm using Fuzzy Logic

The proposed trajectory generation system with fuzzy logic works as follows. At first, the information from the ultrasonic sensors is being gathered and fed to the host PC, where it is further processed in the next step of the algorithm. Based on the sensor information obtained at the previous step, the algorithm calculates parking lot sizes ( $L$ ,  $DB$ ,  $DT$  in Fig. 4) and builds the map. Next, the algorithm calculates the feasible solution sets of the turning radiuses  $R_1$ ,  $R_2$  and their corresponding minimum and maximum boundaries ( $R_{imin}$  and  $R_{imax}$ ,  $i = 1, 2$ ), along with the sets of starting points margins ( $Lp_{min}$ ,  $Lp_{max}$ ) based on the environment parameters and parking lot sizes obtained earlier (see previous step).

The algorithm checks if the parking is possible or not; the

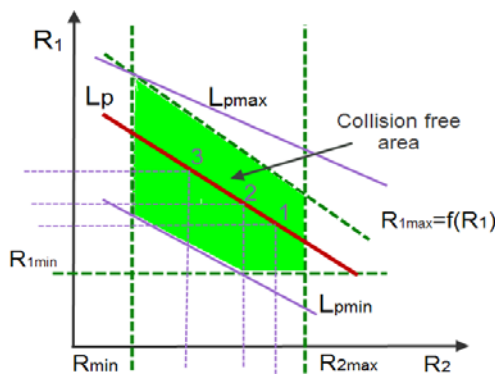


Fig. 6 Collision-free area

parking is considered possible based on two conditions. First, the radiuses  $R_{imax}$  and  $R_{imin}$ ,  $i = 1, 2$ , obtained at the previous step, satisfy the following inequality:  $R_{imax} > R_{imin}$ .

If this inequality holds, the algorithm proceeds with verification of the feasibility of the current starting point, and the parking is considered possible if the latter is true. If the parking is considered impossible due to the wrong starting point, the algorithm generates the command to move the car backward or forward, depending on the current position. However, if the parking is considered impossible due to the unacceptability of the turning radiuses, we need to find a new parking slot. If the parking is considered possible, the algorithm selects the radius  $R_1$  using fuzzy logic controller based on the following parameters: vehicle velocity ( $V$ ), distance to the top obstacle ( $DT$ ) and the starting point ( $Lp$ ). After that, the algorithm calculates  $R_2$  using equation (3). Finally, the system estimates all parameters for trajectory ( $s_1$  and  $s_2$ ) and generates the law for steering angle.

#### 3.2. Design of Fuzzy Logic System

As it was mentioned above, the first input of the fuzzy system is the velocity of the vehicle. During parking velocity is relatively small (less than 10 km/h or 2 m/s for RC car). Thus, if velocity is low it means that we can follow the trajectory with relatively big curvature and we can choose smaller radius for trajectory; comparatively high velocity means that following the trajectory with relatively big curvature will be poor and it's better to choose bigger radius for trajectory.

However, having only one parameter - velocity - is not enough for choosing the radius of the trajectory, because we need to consider possible collisions, therefore, some additional

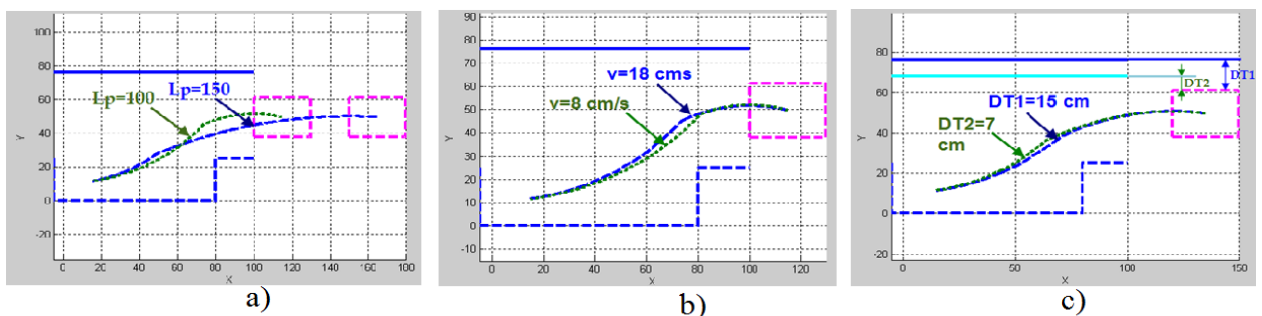


Fig. 7 Two trajectories: a) for different  $Lp$ ; b) for different  $V$ ; c) different  $DT$

geometrical parameters should also be used as auxiliary inputs.

Hence, the second input of fuzzy system is chosen to be the distance to the top obstacle (DT). If the distance to the top obstacle is small (DT = small), we need to choose big radius  $R_1$  of the trajectory in order to avoid collision with top obstacle, in case of the big distance to the top obstacle (DT = big), we can choose any radius (even small).

The third input in the system is the starting point  $L_p$  (coordinate  $x$  of initial point for vehicle). If the starting point  $L_p$  is close, we need to choose small radius  $R_1$  of the trajectory in order to avoid collision with bottom obstacle; in case when the starting point  $L_p$  is in the average distance, we need to choose medium radius  $R_1$  of the trajectory in order avoid collision with bottom obstacle; and in case when the starting point  $L_p$  is far, we need to choose big radius  $R_1$  of the trajectory due to the same reason with two previous cases. Therefore we can obtain the following membership functions for the starting point  $L_p$ .

Since range for  $L_p$  defined as a constant values and minimum/maximum values for  $L_p$  change depending on the information about parking environment, Fuzzy Logic system may interpret input  $L_p$  incorrectly related to  $L_{pmin}$ ,  $L_{pmax}$ . That's why instead of absolute value of starting point  $L_p$ , relative difference for  $L_p$  (0 -100) % of  $\Delta L_{pmax}$  was chosen as the third input of controller. We can calculate relative difference as follows:

$$\begin{aligned} \Delta L_p &= L_p - L_{pmin} ; \\ \Delta L_{pmax} &= L_{pmax} - L_{pmin}; \\ \Delta L_{pr} &= \Delta L_p / \Delta L_{pmax} \text{ (in \%)} \end{aligned} \tag{13}$$

The output of the fuzzy system is the radius  $R_1$ , but since range for  $R_1$  defined as a constant values and minimum/maximum values for  $R_1$  change depending on the information about parking environment, fuzzy logic system may choose incorrect values for  $R_1$  which are not satisfied non-collision conditions, that's why relative difference of  $R_1$  an  $R_{1max}$  is used as output of fuzzy system.  $R_1$  may be found from  $\Delta R_1$  using the similar way with  $L_p$  input. Output value  $\Delta R_1$  in % multiply by  $\Delta R_{1max}$  and add we result to the  $R_{1min}$ . The result is desired  $R_1$ . After  $R_1$  is defined,  $R_2$ .can be calculated easily by using (5).

Fuzzy inference generated with Mamdani method is used. The method of defuzzification is centroid. For more details on the parameters of the fuzzy logic the reader is referred to [8].The output of Fuzzy Logic Controller is the set of radiuses ( $R_1$ ,  $R_2$ ), which give us desired trajectory for parking.

The final rules set for choosing  $R_1$  value is given in Tables 1 and 2.

Table 1. Fuzzy rules for case when DT is "big"

Velocity \ delLp	Low	High
Close	Small	Medium
Average	Medium	Big
Far	Big	Big

Table 2. Fuzzy rules for case when DT is "small"

Velocity \ delLp	Low	High
Close	Medium	Big
Average	Medium	Big
Far	Big	Big

## 4. Simulation and experimental results

### 4.1. Computer simulation

The MATLAB was used for the computer simulations. Three different kinds of simulations were conducted.

At first, the effect of the distance to the starting point was investigated. In this simulation, the parameter DT was set to 15 cm and the velocity was chosen to be  $V = 15$  cm/s. The starting point was varied between 160 mm and 90 mm. The resulting parking trajectories for the values of  $L_p$  close to its maximum and minimum values are shown in Fig.7 (a). It can be clearly seen that for the closer starting point the smaller radius was chosen, and vice versa.

In the second series of experiments, the distance to the obstacle was fixed, DT was assumed to be 15 cm, and the starting point was fixed at  $L_p = 100$  cm. The velocity has been varied between 8 cm/s and 18 cm/s. The resulting parking trajectories calculated for two extreme velocity cases are shown in Fig. 7(b). One can see the effect of different values of the car speed on the resulting parking trajectory. When the parking lot dimensions are fixed, higher parking velocity results in the increase of the radii  $R_1$  and  $R_2$  which improves parking safety.

Finally, in the last simulation case, the effect of change in DT on the parking trajectory was investigated. The velocity was chosen to be  $V = 15$  cm/s and the starting point was set to  $L_p = 120$  cm. The calculated parking trajectories are shown in Fig. 7(c). One can see that for the smaller distance bigger values of



Fig. 8 Various RC-car positions during the experiments

the radii radius  $R_1$  and  $R_2$  were chosen, and vice versa. It should be also noted that DT has a smaller effect on the parking trajectory in comparison to the starting point  $L_p$  (first simulation case) and car velocity  $V$  (second simulation case). Unlike the majority of existing parking assistant systems which consider only the minimum value of DT, the proposed system generates a different parking trajectory even in the cases when DT is big enough. This results in a parking trajectory which is easier to for the vehicle to follow.

#### 4.2. Experimental system

System prototype was designed in order to verify the performance of the proposed algorithm. A remotely controlled (RC) car was used as a reference model of real car. The footprint of the RC car was 30x23 cm.

The prototype system consists of a servomotor and DC motor, ultrasonic sensors, speed sensor (encoder) and microcontroller, Zigbee communication block and the battery pack.

In a series of experiments, the computer simulation environment has been reconstructed, and the same tasks with the simulation cases described above have been given to the trajectory generation controller. The goal was to park the car in a designated parking lot avoiding the collisions. The trajectory of the car has been recorded for further comparison. Different stages of RC car parking are shown in Fig. 8.

During the experiments, all required tasks have been completed, and the car parked in the parking lots with different parking parameters without any collisions. The simulation and experimental results have an error of approximately 5 %, which is mainly caused by hardware nonlinearities. Therefore, we can conclude that our trajectory generation algorithm works properly and generates satisfactory steering law for different parameters of parking environment or vehicle settings.

### 5. Conclusion

In this work, a design of an original auto parking system based on intelligent trajectory generation algorithm using fuzzy logic is described. The performance of the algorithm has been verified in a series of computer simulations prior to hardware implementation. For the hardware verification of the proposed algorithm's performance, the experimental testbed based on a car has been designed and assembled. The results shown by the test model in hardware experiments were very close to the simulated results, which led to a quality non-collision parking of the vehicle.

The proposed intelligent trajectory generation algorithm has several advantages, such as high robustness to the variance in parking parameters and the fact that the algorithm takes into account the velocity of vehicle. This brings the algorithm closer to the human drivers' logic. One of the main disadvantages of this algorithm should be considered the lack of the analytical solution for continuous parking problem. This issue will be in

the scope of our future work.

### References

- [1] M. Sugeno and K. Murakami, "Fuzzy parking control of a model car," *Proc. IEEE Conf. Dec. Control*, pp. 902–903, 1984.
- [2] A. Ohata and M.Mio, "Parking control based on nonlinear trajectory control for low speed vehicles," *Proc. of IECON*, Kobe, Japan, pp. 107-112, 1991.
- [3] I. E. Paromtchik and C. Laugier, "Motion generation and control for parking an autonomous vehicle," *Proc. of the 1996 IEEE Int. Conf. on Robot. Automat.*, Minnesota, pp. 3117–3122, 1996.
- [4] T. Inoue, M. Q. Dao, and K. Liu, "Development of an Auto-Parking System with Physical Limitation," SICE 2004 Annual Conference, Hokkaido Institute of Technology, Japan, pp. 1015-1020, 2004.
- [5] M. Q. Dao, and K.-Z. Liu, "Development of A Practical Automatic Parking Technology for Automobiles," *Proc. of the 45th IEEE Conference on Decision & Control*, USA, pp. 1727-1732, 2006.
- [6] M.Hanafy, M.M. Gomaa, M.Taher, A.M. Wahba, "Path generation and tracking for car automatic parking employing swarm algorithm," *Computer Engineering & Systems (ICCES)*, pp. 99 – 104, 2011.
- [7] R. Mukherjee, A. Anderson, "A surface integral approach to the motion planning of nonholonomic systems," *Journal of Dynamic System, Measurement, and Control*, vol.116, no.3, pp. 315–325, 1994.
- [8] F. Gomez-Bravo, F. Cuesta, A. Ollero, "Parallel and diagonal parking in nonholonomic autonomous vehicles," *Engineering Applications of Artificial Intelligence*, vol. 14, no.1, pp. 419-434, 2001.
- [9] A. Razinkova and H.-C. Cho, "A trajectory generation algorithm in autonomous parking system using fuzzy logic," *Poc. of KISS Fall Conference 2009*, vol.1, pp. 83-86, 2009.

---

#### Anastasia Razinkova

Ph.D. Student of the Korea University of Technology and Education  
 Research Area: Intelligent algorithms, Computer vision, Robotics.  
 E-mail : anastasia@kut.ac.kr

#### Hyun-Chan Cho

Professor of the Korea University of Technology and Education  
 Research Area: Fuzzy, Neural Network, System Control, Robotics.  
 E-mail : cholab@kut.ac.kr

#### Hong-Tae Jeon

Professor of the Chung-Ang University  
 Research Area: Fuzzy, Neural Network, System Control, Robotics.  
 E-mail : htjeon@cau.ac.kr