

1-1-1998

# Cell Mapping Based Fuzzy Control of Car Parking

Tea-Quin Kim

Ming-Chuan Leu

Missouri University of Science and Technology, mleu@mst.edu

Follow this and additional works at: [http://scholarsmine.mst.edu/mec\\_aereng\\_facwork](http://scholarsmine.mst.edu/mec_aereng_facwork)



Part of the [Aerospace Engineering Commons](#), and the [Mechanical Engineering Commons](#)

---

## Recommended Citation

T. Kim and M. Leu, "Cell Mapping Based Fuzzy Control of Car Parking," *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, Institute of Electrical and Electronics Engineers (IEEE), Jan 1998.

The definitive version is available at <https://doi.org/10.1109/ROBOT.1998.680716>

This Article - Conference proceedings is brought to you for free and open access by Scholars' Mine. It has been accepted for inclusion in Mechanical and Aerospace Engineering Faculty Research & Creative Works by an authorized administrator of Scholars' Mine. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact [scholarsmine@mst.edu](mailto:scholarsmine@mst.edu).

# Cell Mapping Based Fuzzy Control of Car Parking

Ming C. Leu<sup>1</sup> and Tea-Quin Kim<sup>2</sup>

1. National Science Foundation, Arlington, Virginia, U.S.A., on leave from New Jersey Institute of Technology, Newark, New Jersey, U.S.A.
2. Samsung Inc., Seoul, KOREA.

*Abstract* – This paper describes the development of a near-optimal fuzzy controller for maneuvering a car in a parking lot. To generate the rules of the fuzzy controller, a cell mapping method is utilized to systematically generate near-optimal trajectories for all possible initial states in the parking lot. Based on the input-output relations of these trajectories, which represent the states and controls of the corresponding cells, a set of fuzzy rules are generated automatically. In order to result in a small number of fuzzy rules from the large amount of numerical information generated by cell mapping, grouping of trajectories is proposed and each rule applies to the cells in one group. This reduces substantially the number of rules in the fuzzy controller compared with establishing the rules directly using the control data of individual cells.

## I. INTRODUCTION

Fuzzy control is one of the most successful application areas of the fuzzy theory invented by Zadeh [1]. It is effective for solving control problems which are difficult or even impossible to solve by developing precise mathematical models. To construct a fuzzy controller, the generation of fuzzy rules is the main issue. Fuzzy rules are often generated by extracting knowledge from skillful human operators.

The application of fuzzy control for a car parking control problem was introduced by Sugeno et al. [2]. Several methods have since been proposed for the generation of fuzzy rules applicable to solving this problem. Kosko [3] developed a space clustering method to obtain fuzzy rules with the fuzzy associative memory. Wang and Mendel [4] proposed a fuzzy rule generation method by assigning a degree to each rule from input and output sample data. Lin and Lee [5] constructed the input-output mapping using a neural network. These research works were all based on the numerical input and output data obtained by a human expert with lots of experiments for the states that the car encounters.

Cell mapping can generate global optimal controls for nonlinear dynamic systems. This approach involves dividing the continuous state space into finite discrete cells. Hsu [6] developed the original cell-to-cell mapping method which first generates all possible paths and then searches for optimal paths with a systematic search algorithm. A modified method developed by Zhu [9] was used by Zhu and Leu [10] to solve the optimal trajectory planning problem for industrial manipulators. To implement, at each sampling time a system state is identified and the cell corresponding to this state is checked against the controller table to look for the control to use in the next time interval. A problem with the table-based control is that the table can be very large, especially if the system has many inputs and outputs. Table based control may also give a bumpy response as the controller jumps from one table value to another.

From the perspective of fuzzy control, the optimal control strategy obtained from cell mapping can be incorporated into the fuzzy controller. The near-optimality is realized by deriving fuzzy rules based on the numerical data of optimal cell mapping. A fuzzy controller constructed from the cell mapping data provides a promising solution for the global optimal control of nonlinear dynamic systems. In our work described herein, the fuzzy controller construction is based on cell mapping generated optimal control data of a dynamic system, instead of using data from human experiments.

To construct a fuzzy controller based on the numerical information of the optimal cell mapping data, there are two main issues: first, how to design the rules of the fuzzy controller; second, how to manage the great amount of numerical information generated by cell mapping. Kang and Vachtsevanos [7] generated fuzzy rules based on cell mapping with a tree search approach. In their method, each cell of the state space is used as a fuzzy region, i.e. a region for a fuzzy rule. However, the number of fuzzy rules can be enormous if the amount of optimal cell mapping data is very large. It is desirable to

have a fuzzy controller with a small number of fuzzy rules for the entire state space.

In this paper, a cell mapping based systematic method is developed to construct a fuzzy controller for car parking control. Near-optimal car trajectories are created from the cell mapping data, and trajectories with similar features are collected to form groups. Fuzzy control rules and membership functions are then expressed with respect to the trajectory groups instead of individual cells. The developed fuzzy controller is shown in simulation to have a performance similar to that of the table-based controller generated from optimal cell mapping.

## II. CELL MAPPING

To generate the global optimal trajectory, the description of trajectories (or paths) in task space is converted to a description in discrete cell state space. In the discrete cell state space, a cell mapping algorithm [6, 9] plans the optimal trajectory with the dynamic equations of a system and its constraints. The cell mapping algorithm constructs tabular numerical information which represents trajectories and their control actions.

To generate global optimal trajectories of a dynamic system, the continuous state space is divided into finite discrete cells. Then, the behavior of the dynamic system is transformed into a description in the discrete cell space. The discrete cells are in the form of rectangular shape. To construct the discrete cell space, first each axis of the continuous space is divided into a number of intervals, with size  $h_i$  for axis  $x_i$ . An interval is denoted by an integer  $z_i$ , i.e.

$$\left(z_i - \frac{1}{2}\right) h_i \leq x_i \leq \left(z_i + \frac{1}{2}\right) h_i \quad (1)$$

The  $n$ -tuple  $(z_1, \dots, z_n)$  is called a cell and is denoted by  $z$ .  $n$  is the dimension of the cell space; for example,  $n$  is 2 for  $(x, y)$  and is 3 for  $(x, y, \theta)$ .

Let the control output from actuators be a  $p$ -dimensional vector,  $u$ , given by:

$$u = [u_1, u_2, \dots, u_p] \quad (2)$$

The differential equation of an  $n$ -dimensional dynamic system can be generally expressed as

$$\dot{x} = f(x, u(t), t) \quad (3)$$

where  $x \in R^n$  and  $t \in R$ . By solving this set of differential equations, a point mapping can be established in the form of

$$x(k+1) = x(k) + \int_{t_k}^{t_{k+1}} f(x(k), u(k)) dt \quad (4)$$

$$k = 1, \dots, N$$

A cell mapping can be constructed from a point mapping and written in the form

$$z(k+1) = H(z(k), u(k), t(k)) \quad (5)$$

where  $z \in I^n$  is an  $n$ -tuple of integers.

The dynamics of the system are expressed in terms of cell-to-cell mappings. Determining the image of a cell can proceed as follows: For a given cell  $z(k)$ , first find the coordinates of its center  $x(k)$ . Under control  $u(k)$  and time duration  $t(k)$ ,  $x(k+1)$  is determined as the image of  $x(k)$  by point mapping. If the corresponding cell of point  $x(k+1)$  is  $z(k+1)$ , then  $z(k+1)$  is the image cell of  $z(k)$ .

The functional to be minimized for finding an optimal path assumes the form

$$J = \sum_{k=1}^m (c_k \int_{t_k}^{t_{k+1}} f_0(x_k, u_k) dt + t_k) \quad (6)$$

where  $k$  is a number used to indicate the mapping step,  $m$  is the total number of steps from the initial cell to the target cell, and  $c_k$  is a coefficient used to weight the two terms in  $J$ . For time optimal control, all  $c_k$ 's are zero. If it is important to save control energy,  $c_k$ 's should be large.

After the discrete cell state space is constructed and the goal state (target) given, optimal cell-to-cell mappings are performed to search for optimal trajectories from various initial states to the target in the discrete cell state space satisfying all the conditions and dynamic equations.

## III. FUZZY SYSTEMS

A fuzzy controller consists of a set of fuzzy control rules processed with a system of logic inference. The fuzzy rule base consists of a collection of rules which represents all possible situations and their corresponding control actions. The logic inference [11] converts the linguistic labels of the rule base into a final crisp control action. Figure 1 illustrates the structure of a rule-based fuzzy controller.

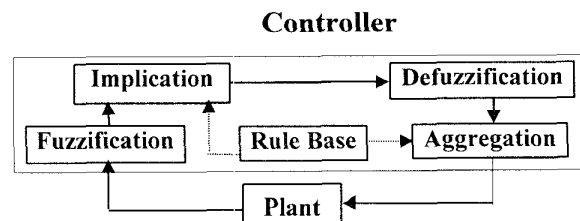


Fig. 1 Structure of rule-based fuzzy controller.

The rules in a fuzzy controller take the following form:

$$\text{If } \langle \text{condition} \rangle, \text{ then } \langle \text{control action} \rangle \quad (7)$$

The  $\langle \text{condition} \rangle$  is the premise part and the  $\langle \text{control action} \rangle$  is the consequence part. For example, a fuzzy control rule to steer a vehicle toward a parking target may be: If Parking Target  $x$  is ‘ahead,’ then steering angle  $y$  is ‘straight.’ The terms  $x$  and  $y$  are referred to as the state variable and control variable, respectively. The terms “ahead“ and “straight” are linguistic labels. The adjectives such as (ahead, straight) in the rules do not correspond to precisely defined ranges.

A fuzzy variable is a linguistic term (or label) associated with a membership function. The fuzzy rule base represents qualitative knowledge in a fashion which can be interpreted by a fuzzy logic process. Among the common rule-based representations, one is the Mamdani type:

$$\begin{aligned} R_1: & \text{ if } x_1 \text{ is } F_1^1 \text{ and } x_2 \text{ is } F_2^1 \dots x_n \text{ is } F_n^1, \text{ then } y \text{ is } A_1; \\ R_2: & \text{ if } x_1 \text{ is } F_1^2 \text{ and } x_2 \text{ is } F_2^2 \dots x_n \text{ is } F_n^2, \text{ then } y \text{ is } A_2; \\ & \dots \\ R_l: & \text{ if } x_1 \text{ is } F_1^l \text{ and } x_2 \text{ is } F_2^l \dots x_n \text{ is } F_n^l, \text{ then } y \text{ is } A_l; \end{aligned} \quad (8)$$

where  $F_1^i, F_2^i, \dots$ , and  $A_i$  are premise and consequence linguistic labels, respectively.

The process of fuzzification is assigning a “belief” value to a real-valued input variable with respect to each membership function. It maps each coordinate  $x_i$  of a crisp point  $x$  into a fuzzy set. Implication maps the belief values of the premise part to the output of the fuzzy logic system, which is the consequence part. Each fuzzy IF-THEN rule defines a fuzzy implication. Many fuzzy implication rules have been proposed in the fuzzy logic literature. Two commonly used fuzzy implication rules are the Min-operation implication and Product-operation implication. Defuzzification generates real valued controls. There are many defuzzification methods. Among them the center of gravity method and the mean of maximum method are used frequently. Since multiple rules of a fuzzy control system can be active simultaneously, all of the active rules are combined to create the final result. One simple way to combine the rules is taking the weighted average of the outputs. This is the aggregation step.

#### IV. CONSTRUCTION OF NEAR-OPTIMAL FUZZY CONTROLLER

We present in this section a method which uses the numerical data obtained from optimal cell-to-cell mapping to efficiently generate fuzzy rules for constructing a near-optimal fuzzy controller. We first discuss how to group trajectories for this purpose.

##### IV.1 Defining Trajectories

To define trajectories from the optimal control table and group them based on similarity features, we introduce the concept of initial cell, simple cell, and merged cell. A cell  $z_i$  is an initial cell (IC) if there exists no  $z_j$  such that  $H(z_j) = z_i$  and  $z_i \neq z_j$  for  $j=1, \dots, N_C$ . The cell  $z_i$  is a simple cell (SC) if there exists only one  $z_j$  such that  $H(z_j) = z_i$  and  $z_i \neq z_j$ . If there exists more than one  $z_j$ , then  $z_i$  is a merged cell (MC). A trajectory is a set of connected cells, each evolves either from an initial cell to a merged cell or from a merged cell to another merged cell.

The following procedure is used to derive trajectories in a cell state space for the purpose of grouping:

Step 1: Each cell in the set  $\{z_1, z_2, \dots, z_r\}$  is assigned as an IC, SC, or MC.

Step 2: Construct the set  $\mathfrak{R}$  consisting of IC’s and MC’s.

Step 3: From any cell  $S_i \in \mathfrak{R}$ , various cells are linked according to Eq. (5) until  $S_j \in \mathfrak{R}, S_i \neq S_j$ , is encountered. This forms one trajectory from  $S_i$  to  $S_j$ .

Step 4: Step 3 is performed repetitively for every element in  $\mathfrak{R}$ .

For illustration, the gray rectangles in Figure 2 are initial cells, the white rectangles are simple cells, and the black rectangles are merged cells. There are four trajectories in Figure 2: one from IC<sub>k</sub> to MC<sub>p</sub>, one from IC<sub>j</sub> to MC<sub>h</sub>, one from IC<sub>k</sub> to MC<sub>h</sub>, and one from MC<sub>h</sub> to MC<sub>p</sub>.

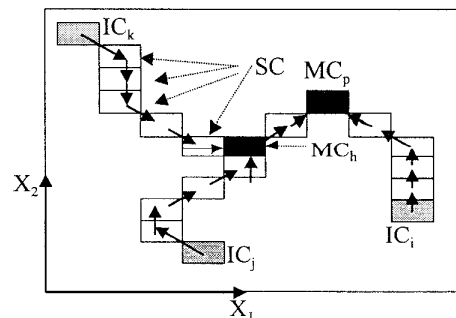


Fig. 2 Illustration of simple cells, initial cells, merged cells, and trajectories on cell state space.

#### IV.2 Features of Trajectories

To group similar trajectories, we consider features which distinguish trajectories by the overall locations and controls of their cells. If the difference between the overall locations and controls of the individual cells of two trajectories is small enough, these two trajectories are said to be similar. Since two trajectories generally have different numbers of cells, it is not possible to compare the locations and controls of their individual cells. We thus use the mean location and the mean control of the cells for each trajectory. Let  $T_a$  and  $T_b$  be the two trajectories under examination. The features of them are the mean locations,  $\bar{x}_a$  and  $\bar{x}_b$ , and the mean controls,  $\bar{u}_a$  and  $\bar{u}_b$ . Based on these features, we can compute the difference between the two trajectories as follows:

$$\eta_1 = |\bar{x}_a - \bar{x}_b| \quad (9)$$

$$\eta_2 = |\bar{u}_a - \bar{u}_b| \quad (10)$$

$$\beta = \gamma_1 \eta_1 + \gamma_2 \eta_2 \quad (11)$$

The values of  $\gamma_1$  and  $\gamma_2$  represent the weights of individual features and they can be chosen depending on the relative importance of the features.

#### IV.3 Grouping Similar Trajectories

The basis for grouping similar trajectories is Equation (11). The grouping of trajectories is achieved by beginning with empty group lists. The first trajectory to be classified forms the first group. Assume that second trajectory considered for grouping has a feature difference of  $\beta$  from the trajectory in the first group. If this difference is less than or equal to the predefined threshold  $\beta\mathcal{T}$  then the considered trajectory is placed in the first group. Otherwise it forms another group which is the second group. New groups are generated whenever the minimum feature difference between a newly considered trajectory and the trajectories of all established groups exceeds the threshold. The process continues until all the trajectories in the cell space have been categorized.

#### IV.4 Fuzzy Rule Base and Membership Functions

The main purpose of grouping trajectories is to obtain simplified state and control information for the generation of fuzzy rules and membership functions. Cell states are in the IF-part and control levels are in the THEN-part of a fuzzy rule. There could be different ways of using the cell states and control levels of a trajectory group. An intuitive one is to use a representative trajectory of a group, such as the first trajectory or the longest trajectory of a group. Another is to consider all cells of a group and extract their statistical

properties, which may include the mean location of the cells, its standard deviation, mean control, etc. Based on these statistical properties, fuzzy membership functions corresponding to the various groups can be created.

In this study, we generate the fuzzy rules and membership functions based on the statistical properties of the trajectory groups. If there are  $k$  groups,  $G_j$  ( $j = 1, \dots, k$ ), then the number of fuzzy rules is  $k$ . The membership functions we use have trapezoidal or triangular shape. Figure 3 illustrates the generation of fuzzy membership functions which are projected onto input axes. The projections of a group onto the input axes become the fuzzy membership functions of the IF-part. The mean location,  $m_{x_{i,j}}$ , obtained by calculating the coordinates of all cells of group  $G_i$  is the middle point of the membership function of this group. The standard deviation,  $\sigma_{x_{i,j}}$ , of the mean location of the group is the width of the fuzzy membership function. Based on these two statistical values, a set of fuzzy premises can be created for the fuzzy rule base. The membership functions for the fuzzy consequences are similarly created.

After designing the fuzzy rule base and membership functions, a global fuzzy controller is constructed. Constructing the global fuzzy controller consists of fuzzification, implication, defuzzification, and aggregation, as described previously. In the simulation described in Section V, the fuzzy control system is constructed using the following: 1) the rule base has the Mamdani type, 2) the implication uses the Min-operation, 3) the defuzzification is based on the center of gravity method, and 4) the aggregation uses the weighted average of the outputs.

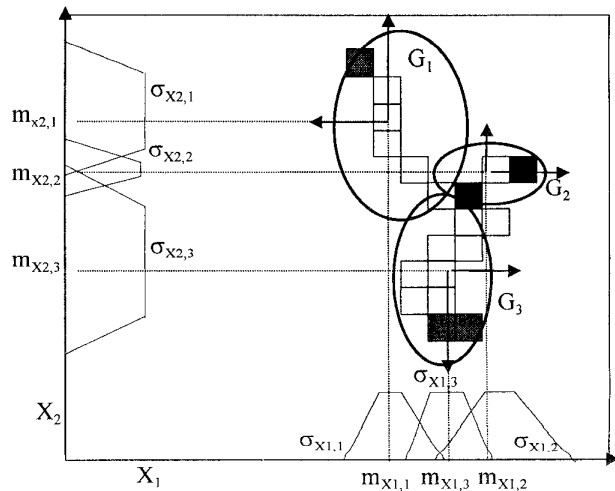


Fig. 3 Each group defines a fuzzy rule region and its membership functions on the input axes.

## V. FUZZY CAR PARKING CONTROL

### V.1 The Car Parking Control Problem

The car parking control problem is illustrated in Figure 4 which shows a car and its parking lot. The state variables representing the car are  $x$ ,  $y$ , and  $\phi$ , which are the Cartesian coordinates of the center of the rear wheels and the angle of the vehicle with respect to the horizontal axis. The objective of the control is to move the center of the rear wheels,  $(x, y)$ , to the target,  $(x_d, y_d)$ , with the vehicle perpendicular to the dock, i.e.  $\phi_d = \pi/2$ , at the target position.

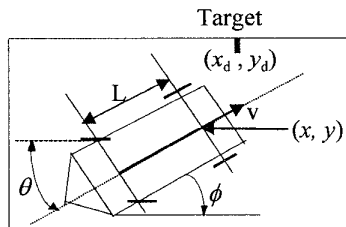


Fig. 4 The car and its parking lot.

Let  $\theta$  denote the steering angle and  $v$  denote the speed of the rear center point of the vehicle. The equations of motion are:

$$\dot{x} = v \cos \phi, \dot{y} = v \sin \phi, \dot{\phi} = \frac{v}{L} \tan \theta \quad (12)$$

where  $L$  is the distance between the center of the front wheel and the center of the rear wheels.

### V.2 Generation of Optimal Trajectories by Cell Mapping

To generate optimal trajectories, the region of interest in the state space is taken to be  $[x_{min}, x_{max}] \times [y_{min}, y_{max}] \times [\phi_{min}, \phi_{max}] = [0, 12] \times [0, 6] \times [-\pi/2, 3\pi/2]$ . The dimensions of each cell are:  $\Delta x = 12/27$ ,  $\Delta y = 6/14$ , and  $\Delta \phi = 2\pi/27$ . The total number of cells is 10,206 ( $N_c = 27 \times 14 \times 27$ ). The cell coordinates of the target are  $[13, 13, 13]$ , which represent the physical coordinates of  $(x, y, \phi) = [6, 3, \pi/2]$ .

In the sorting of cell mappings for optimal trajectories, we use minimization of path length as the objective function. Only one switching of the car moving direction is allowed in this example study. And when it is necessary for the car to switch its moving direction, the switching is always from forward to backward direction because of the prespecified car orientation at the target.

In the cell mapping, first only backward movements of the car are allowed, until all the cells in the cell space are processed. Among the processed cells, the cells are

(target) reachable cells if their trajectories are connected to the target. The other cells are unreachable cells. The trajectories of the unreachable cells are not connected to the target but to the sink cell (the entire region outside the domain region). If there are unreachable cells after the process of allowing only backward movements, then forward movements of the car are allowed to connect previously unreachable cells with previously reachable cells.

After the cell mapping procedure is completed, a total of 5,229 cells are found as reachable cells to the target when one switching of the vehicle moving direction is allowed. Among them, 2,371 cells are reachable cells to the target with only backward movements.

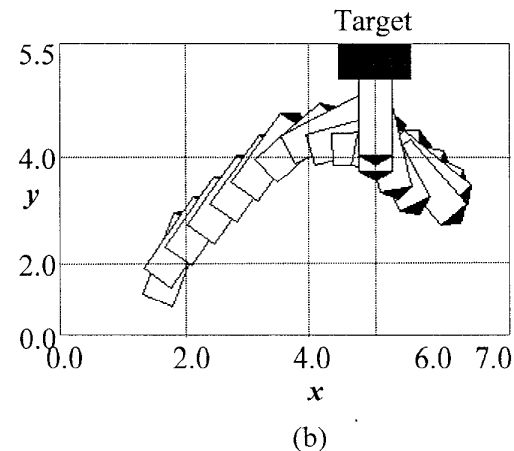
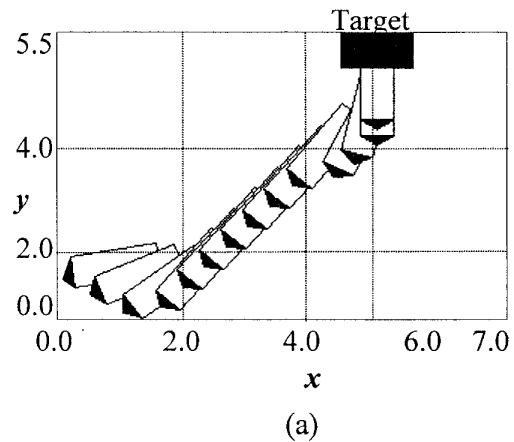


Fig. 5 Car paths for 2 different initial locations, with the same position but different orientations.

Figure 5 shows the car trajectories for 2 different initial locations. These locations are set such that the vehicle is initially close to the boundary of the loading dock. At the position of Figure 5 (a), where  $x = 1.6$ ,  $y = 1.6$ , and  $\phi = 5$  degrees, the car successfully arrives at the

target with only backward movements. At the position of Figure 5 (b), where  $x=1.6, y=1.6$ , and  $\phi=260$  degrees, the car initially moves forward 14 steps to avoid collision with the left wall of the loading dock and then changes its moving direction from forward to backward.

### V.3 Grouping of Trajectories and the Generated Fuzzy Controller

The total number of trajectory groups is 68, obtained using the trajectory determining procedure previously described, with the threshold value  $\beta_T$  set at 3.0. Among the 68 groups, the 6 groups of the longest trajectories are shown in Figure 6.

Figure 7 compares the simulated trajectories using the table-based controller with those using the fuzzy controller for the car parking problem. There are 14 different trajectories starting from different locations in the parking lot. It can be seen that the result of the fuzzy controller approximates that of the table-based optimal controller, despite that only 68 rules are used in the fuzzy controller while a data table for 2,371 cells are used in the table-based controller.

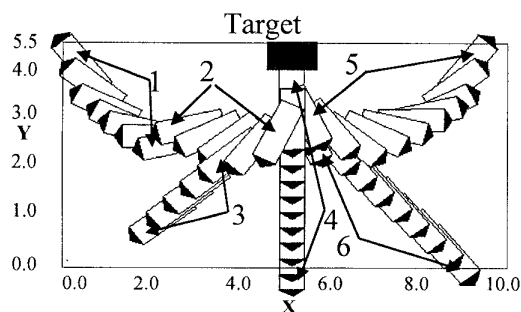


Fig. 6 Representative groups of the six longest trajectories.

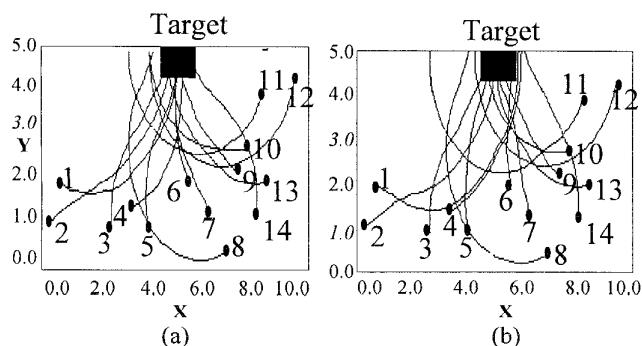


Fig. 7 Simulated trajectories using (a) the table-based controller and (b) the fuzzy controller.

## VI. CONCLUSION

A cell mapping based method has been developed to systematically generate the rules of a near-optimal fuzzy controller for autonomous car parking. In this method, the optimal trajectories in the state space are grouped using the data obtained from cell mapping. The fuzzy rules and membership functions are then generated using the statistical properties of the individual trajectory groups. This method significantly reduces the number of rules in the fuzzy controller compared with generation of fuzzy rules directly using the control data of all individual cells. Simulation results show that the performance of the near-optimal fuzzy controller approximates that of the table-based optimal controller, despite a small number of rules used in the fuzzy controller compared with a large amount of cell data used in the table-based controller.

## REFERENCES

- [1] L.A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338-353, 1965.
- [2] M. Sugeno and M. Nishida, "Fuzzy control of model car," *Fuzzy Sets and Systems*, vol. 16, pp. 103-113, 1985.
- [3] B. Kosko, *Neural networks and fuzzy systems*. Prentice Hall, Englewood Cliffs, New Jersey, 1992.
- [4] L.X. Wang and J. M. Mendel, "Generating fuzzy rules from numerical data from examples," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, no. 6, pp. 1414-1427, 1992.
- [5] C.T. Lin and C.S.G. Lee, "Neural-network based fuzzy logic control and decision system," *IEEE Trans. Comput.*, vol. 40, no. 12, pp. 1320-1336, 1991.
- [6] C.S. Hsu, "A discrete method of optimal control based upon the cell state space concept," *J. of Appl. Mech.*, vol. 46, pp. 547-569, 1985.
- [7] H. Kang and G.V. Vachtsevanos, "Nonlinear fuzzy control based on the vector fields of the phase portrait assignment algorithm," *Proc. of American Control Conf.*, 1990.
- [8] T.Q. Kim, *Global Fuzzy Control based on Cell Mapping*, Ph. D. Dissertation, New Jersey Institute of Technology, Newark, New Jersey, 1996.
- [9] W.H. Zhu, "An applied cell mapping method for optimal control systems," *J. of Opt. Theo. Appl.*, vol. 60, no. 3, 1989.
- [10] W.H. Zhu and M. C. Leu, "Planning optimal robot trajectories by cell mapping," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, 1990.
- [11] Integrated Systems, Inc., *SystemBuild/Workstation Real-Time Fuzzy Logic Block User's Guide*, 1991.