

# Automatic Design of Fuzzy Controllers for Car-Like Autonomous Robots

Illuminada Baturone, Francisco J. Moreno-Velo, Santiago  
Sánchez-Solano, and Aníbal Ollero

IEEE TRANSACTIONS ON FUZZY SYSTEMS, VOL.12, NO.4, AUGUST 2004

Soft Computing, HI 2005

Project 1: Presentation of Journal

Mia Westerlund

# Introduction



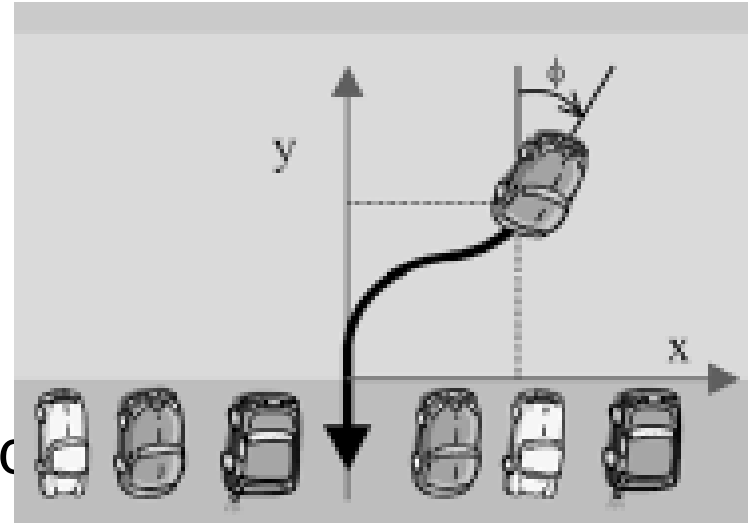
- One of the main "problems" in robotics is motion planning; an autonomous robot has to decide which motions to carry out in order to achieve a certain task
- This paper describes the design and implementation of a fuzzy control system for an autonomous car-like robot to solve the diagonal parking problem
- Computer-aided design tools in Xfuzzy 3.0 (based on specification language XFL3) is used in the design, and the system is implemented and tested on the robot ROMEO 4R

# Moving from A to B

- If there are no obstacles, the problem is to find the shortest path between two points; a finite sequence of two elementary components: arcs of circle (with minimum turning radius) and straight line segments.
- Following this *discontinuous* shortest path between two points means we have to stop the car and reorient the wheels at each discontinuity point.
- Continuous-curvature paths has earlier been obtained by geometrical methods (e.g.  $\beta$ -splines), followed by some path-tracking technique (e.g. predictive control methods)
- Another approach is fuzzy logic-based planners based on heuristic knowledge of expert drivers by fuzzy rules and/or learning these rules with training data taken from the human driving behaviour

# Diagonal Parking Maneuver

- This system doesn't consider the avoidance of unforeseen obstacles but focuses on parking on a constrained parking place defined by static obstacles.
- Starting from a given position  $(x,y)$  and orientation  $\phi$ , the vehicle shall arrive backwards at the desired parking place with right angle and stop there.
- References: Line of parked cars below  $y=0$ , the center of the parking place  $x=0$ , target orientation  $\phi=0$  ( $-180^\circ \leq \phi \leq 180^\circ$ ), the curvature of the trajectory is positive if steering wheel is turned to right, and negative otherwise



# Diagonal Parking Maneuver (2)

- The robot knows its position, orientation and driving speed.
- Kinematic model:

$$\begin{cases} \dot{X} = v \cdot \sin(\phi) \\ \dot{Y} = v \cdot \cos(\phi) \\ \dot{\phi} = \gamma \cdot v \end{cases}$$

where  $(x,y)$  are the coordinates of the vehicle rear axle midpoint,  $\phi$  is the robot orientation with the vertical,  $v$  is the speed, and  $\gamma$  is the vehicle curvature

- Taking into consideration dynamics (traction and direction engines react with some delay) we obtain the first-order model:

$$\begin{cases} v(t) = v_{ref} + [v(t - \Delta t) - v_{ref}] \cdot \exp\left(\frac{-\Delta t}{\tau_v}\right) \\ \gamma(t) = \gamma_{ref} + [\gamma(t - \Delta t) - \gamma_{ref}] \cdot \exp\left(\frac{-\Delta t}{\tau_\gamma}\right) \end{cases}$$

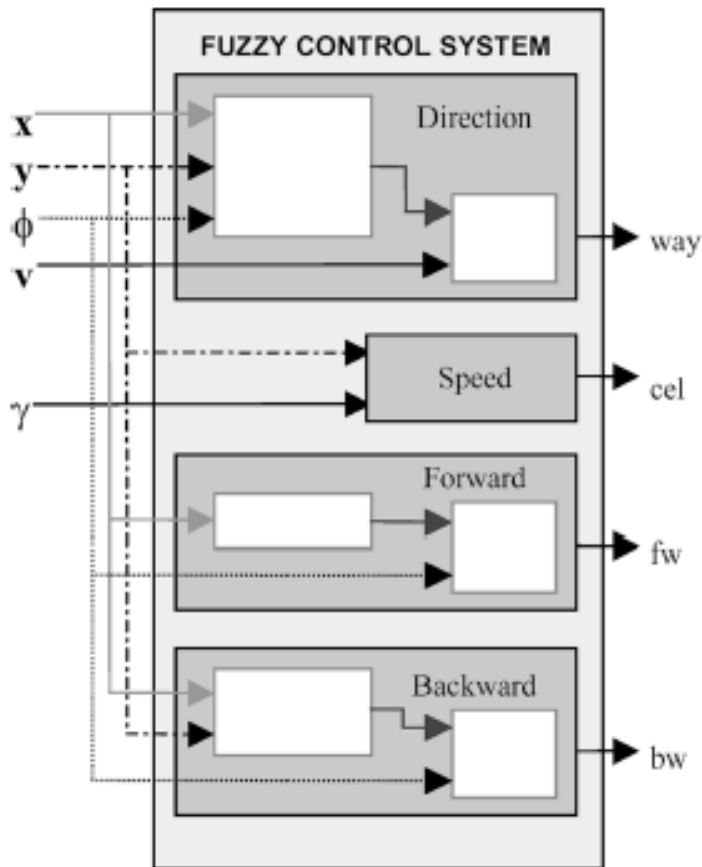
where  $v_{rel}$  and  $\gamma_{rel}$  are the speed and curvature values from the controller, and  $\tau_v$  and  $\tau_\gamma$  are the response times of the traction and direction engines

- Values to the controller  
(see next slides)

$$v_{ref} = way \cdot cel, \quad \gamma_{ref} = \begin{cases} fw, & \text{if } v > 0 \\ bw, & \text{if } v \leq 0 \end{cases}$$

# Structure of the Control System

- Distinguish between two control actions: deciding the driving direction (backward or forward) and the magnitude of the speed, and selecting the proper angle of the wheels (depending on the driving direction).

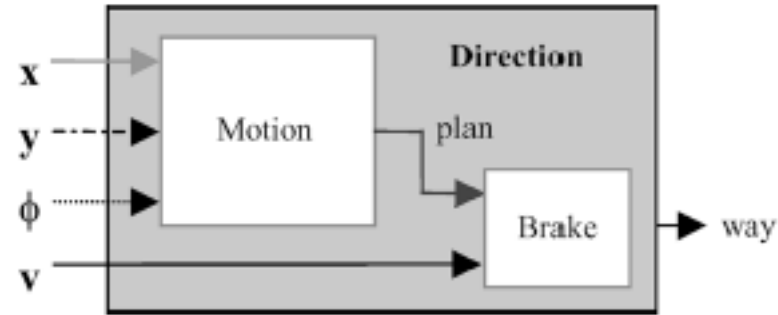


- There are four main modules connected in parallel; *way*, *cel*, *fw*, *bw*
- Heuristic knowledge expressed linguistically has been used when designing the direction and speed modules (translated by XLF3).
- Geometric-based techniques has been used for optimizing the two path planning modules to generate continuous near-minimal-length paths

# Direction Fuzzy Control Module

Two rule bases connected in series

1. *motion*: selects to drive forward, backward, or as previously done depending on the position and orientation of the robot (its goal is to avoid collision with the two lines of cars and with the cars limiting the target parking place)
  2. *brake*: takes into account the current speed of the robot to select the driving direction (without forcing abrupt changes in the robot speed)
- Defuzzification method in both rule bases is of maximum type, it selects as output the consequent of the rule whose activation degree is maximum (if no rule is active, default driving is backwards)

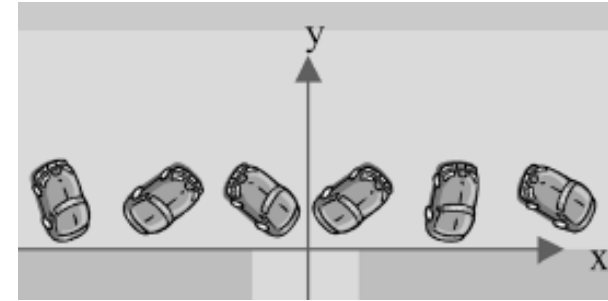


# Direction Fuzzy Control Module (2)

- *motion:*

- 1) If the  $y$  robot coordinate is near the cars of the parking place and the robot orientation  $\phi$  is greater than  $-90^\circ$  and smaller than  $90^\circ$  and the  $x$  coordinate is not zero or the orientation is not zero, the driving direction should be forward

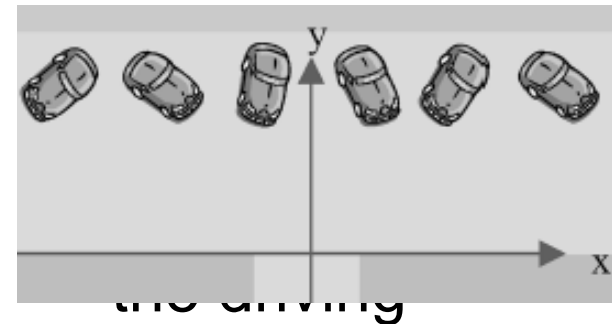
If( $y \leq \text{nearparking} \ \& \ \phi > \text{mf90n} \ \& \ \phi < \text{mf90} \ \& \ (x \neq \text{zero} \ | \ \phi \neq \text{zero})) \rightarrow \text{plan} = \text{forward}$



translation of the rules into the language XFL3 of Xfuzzy 3.0

- 2) If the  $y$  robot coordinate is near the cars the sidewalk in front of the parking the robot orientation  $\phi$  is  $90^\circ$  or smaller than  $-90^\circ$ , direction should be forward

If( $y \geq \text{nearwall} \ \& \ (\phi > \text{mf90} \ | \ \phi < \text{mf90n})) \rightarrow \text{plan} = \text{forward}$



where *nearparking*, *nearwall*, *zero* are represented by triangular membership functions; *mf90*, *mf90n*, *forward* are crisp values represented by 90, -90, 1



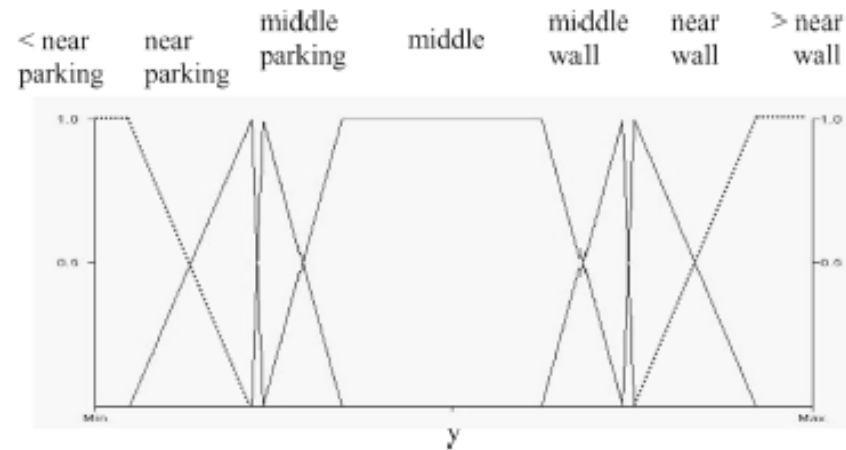
# Direction Fuzzy Control Module (3)

- The driving direction in the middle is the same as in the previous iteration, if the  $x$  coordinate or the orientation is non-zero, otherwise the direction should be backward (in the middle area the vehicle can go backward to approach the parking place or forward to move away from the obstacles; in the last case, the forward direction should be maintained until centering the vehicle approximately ( $x=phi=zero$ )).

If( $y > \text{nearparking} \ \& \ y < \text{earwall} \ \& \ (x \neq \text{zero} \ | \ \phi \neq \text{zero})$ )  $\rightarrow$  plan = previous

If( $x == \text{zero} \ \& \ \phi == \text{zero}$ )  $\rightarrow$  plan = backward

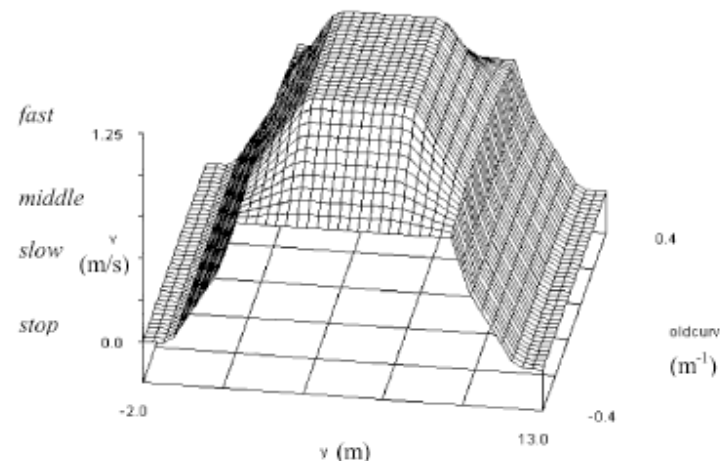
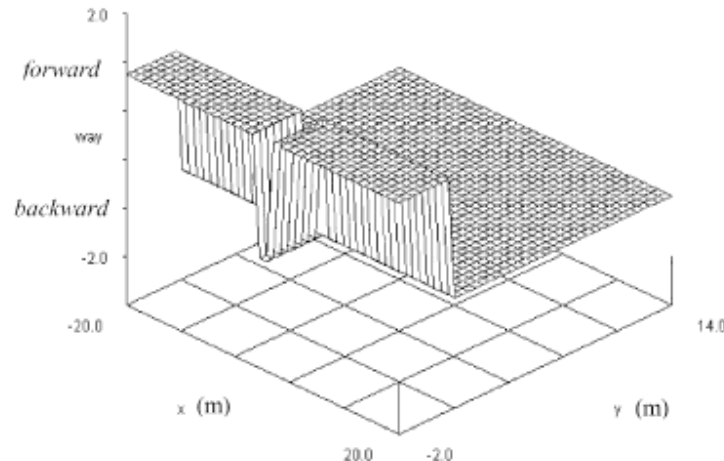
where *previous*, and *backward* are crisp values represented by the singleton values  $-2$  and  $-1$ , respectively.



- By assigning weights to the rules one can simplify the rules..
- Rules for the second base *brake* can be obtained similarly..

# Speed Fuzzy Control Module

- *speed* reduces the the speed when the the curvature (curv) of the trajectory is big and/or the robot is approaching obstacles ( $y$  is small or big); it also stops the vehicle at the target
- rules in XLF3 can be written similarly to the direction module
- Defuzzification: the output of this rule base is calculated as an average of the singleton consequents by applying the fuzzy mean defuzzification method (equivalent to a Takagi-Sugeno inference of zero order); the changes in the speed commands are soft
- Comparing the decision surfaces for *direction* and *speed*:



# Forward Fuzzy Control Module

- *forward* is in charge of generating the reference curvature when the vehicle is avoiding an obstacle; the goal is to drive the robot toward a configuration so that the parking place can be reached easily by driving backward with a zero curvature.
- the suboptimal goal is that the forward maneuvers involve short trajectories made up of arcs of circle of minimum turning radius and straight horizontal segments, which meet the kinematic constraints
- It is very difficult to design a heuristic rule base which accomplishes this objective so the approach is to generate several of these trajectories numerically to obtain a set of training data  $(x, \phi, \gamma)$  covering the universes of discourse of  $x$  and  $\phi$
- Once these training data are available, fuzzy rules have been extracted from them by using the identification and learning tools of Xfuzzy 3.0

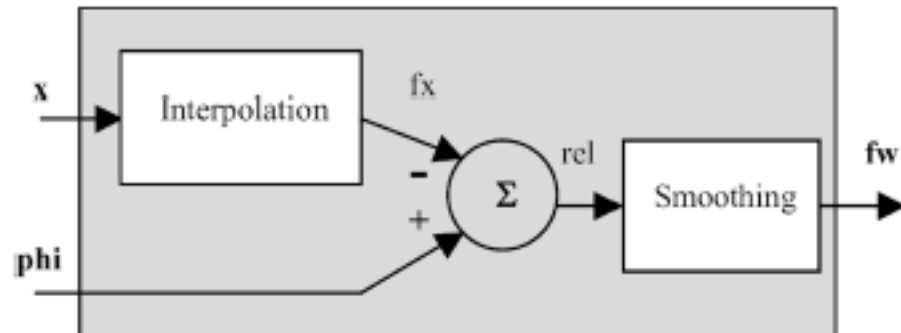
# Forward Fuzzy Control Module (2)

## *Monolithic approach:*

- Wang-Mendel grid-based technique employing membership functions equally distributed over the input universes of discourse
- The initial module is selected to have 7 linguistic labels for  $x$  and 9 for the vehicle angle (represented by Gaussian MF's). A zero-order Takagi-Sugeno system can be identified, approximation to training data improved by supervised learning and the number of rule consequents can be reduced by clustering

## *Hierarchical approach:*

- Structure made up of two rule bases, *interpolation* and *smoothing*, both bases can be designed manually with the previous results (triangular fuzzy mean defuzzification

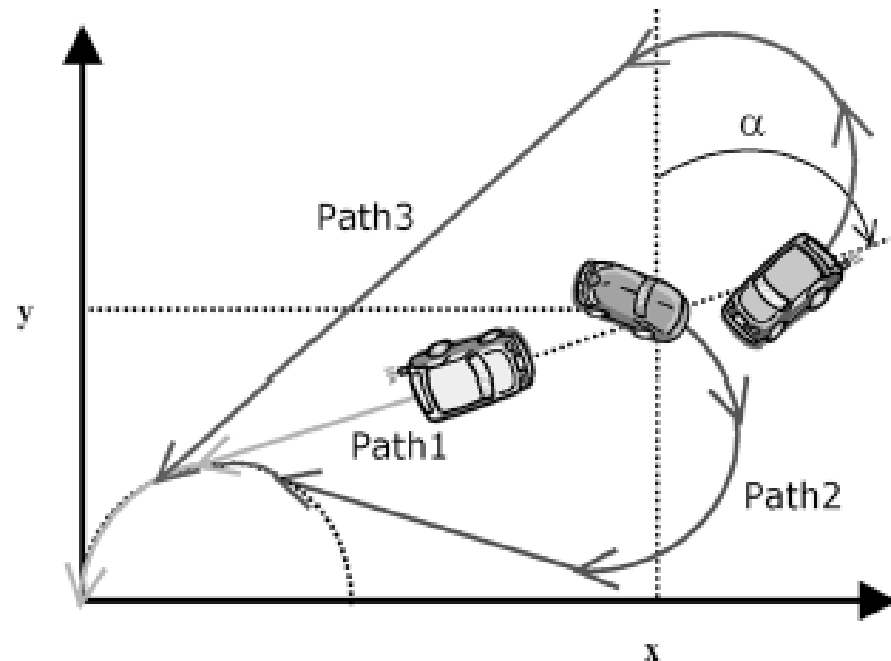
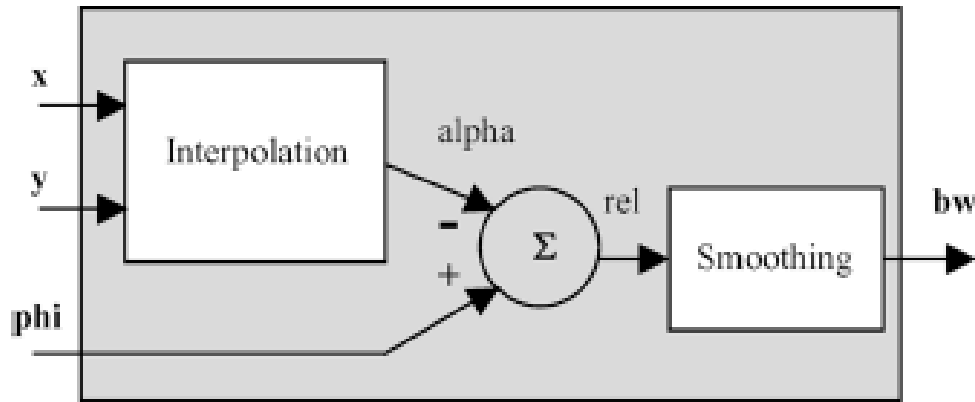


# Backward Fuzzy Control Module

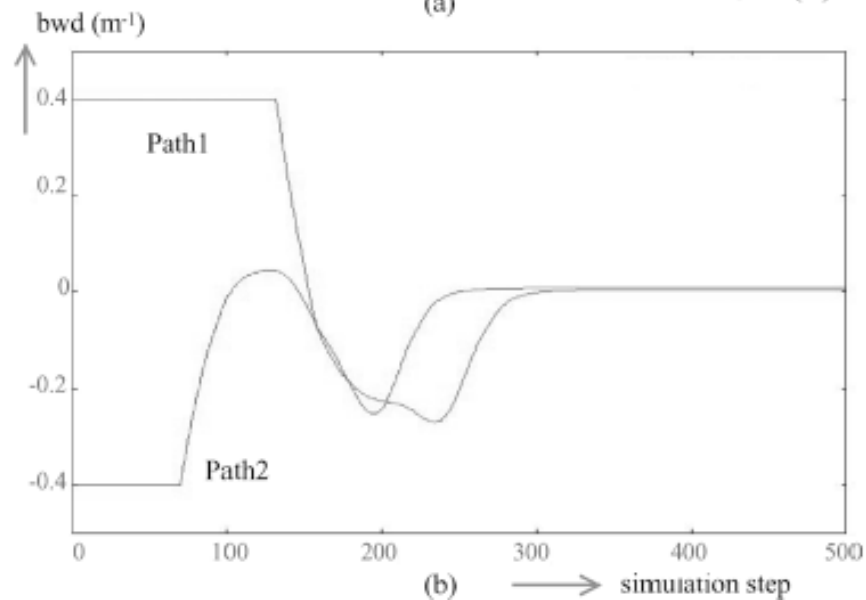
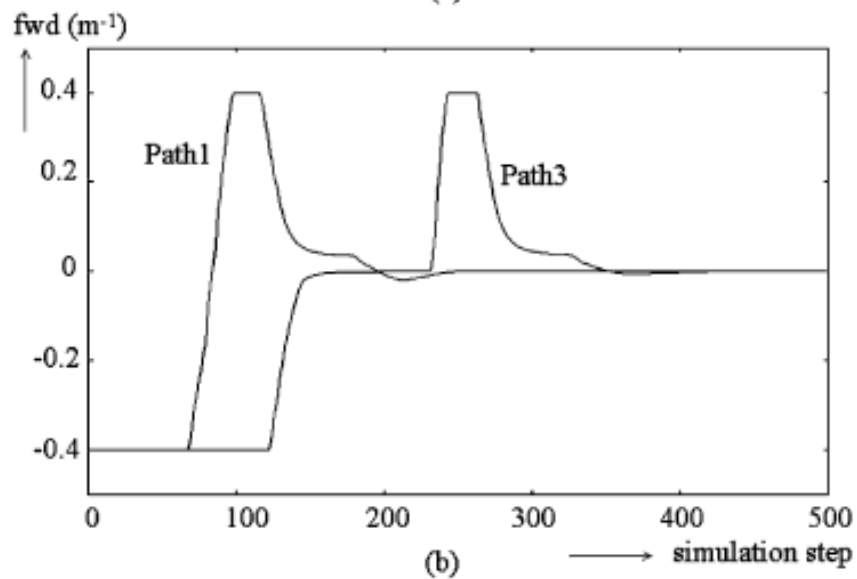
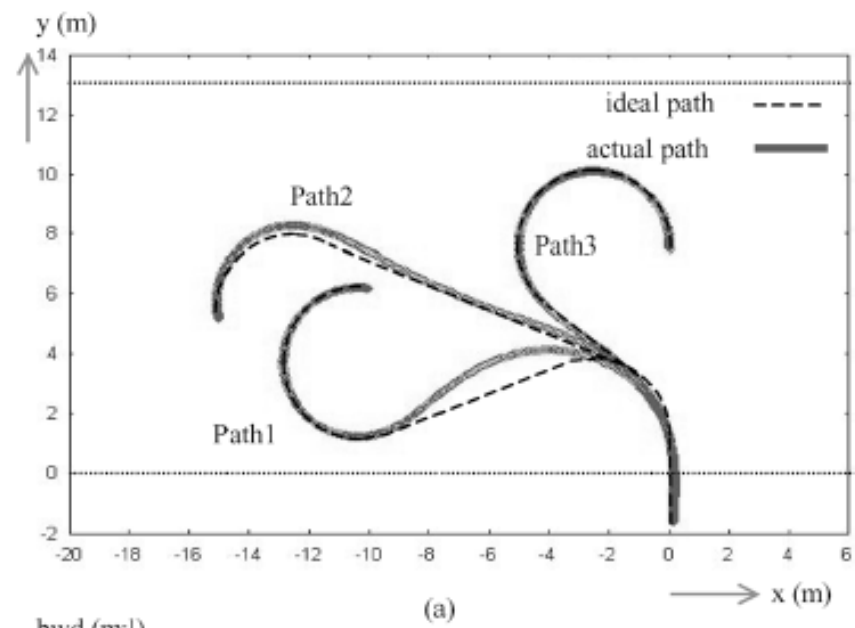
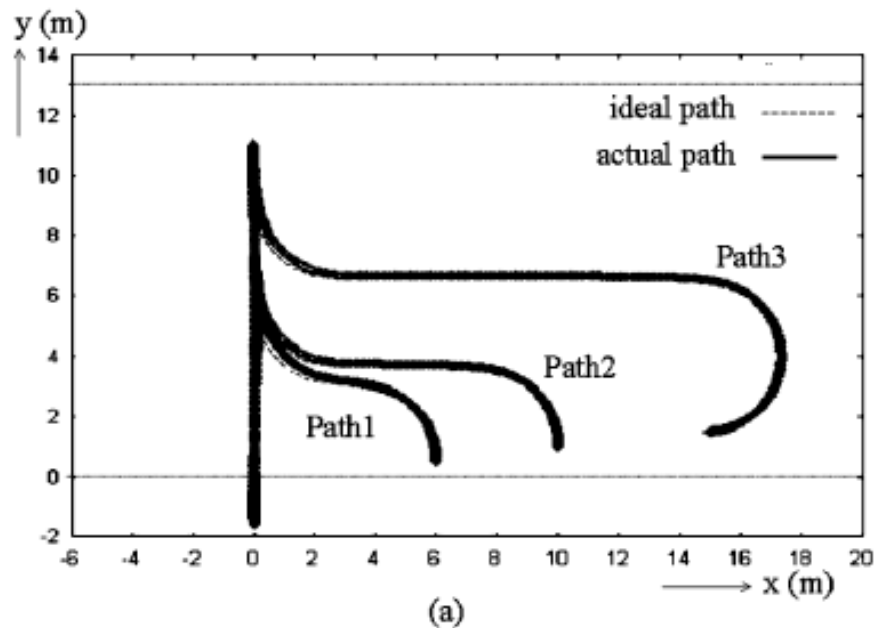
- *backward* generates the reference curvature when the robot is driving backward; the goal of the backward maneuver is to reach the parking place with the shortest path (more restrictive than that the forward maneuvers because the shortest paths which meet the kinematic constraints are made of arcs of circle of minimum turning radius connected by straight segments tangent to both of them; the straight segments are not always horizontal; the curvature depends on  $x, y, \phi$ )
- Approach: exploit geometric considerations to generate a set of training data from which to identify and adjust the rules of the module; hierarchical approach
- A grid-based fuzzy system with normalized triangular MF's in its antecedent rules has been identified, and a first-order instead of a zero-order Takagi-Sugeno inference engine has been used to obtain a good approximation degree with a small number of rules

# Backward Fuzzy Control Module (2)

- Again we have two rule bases connected in series
- Geometrical analysis shows there is angle  $\alpha$  (depending on  $x$  and  $y$ ) associated with switching of the curvature sign



# Simulated results (fw, bw)



# Testing..

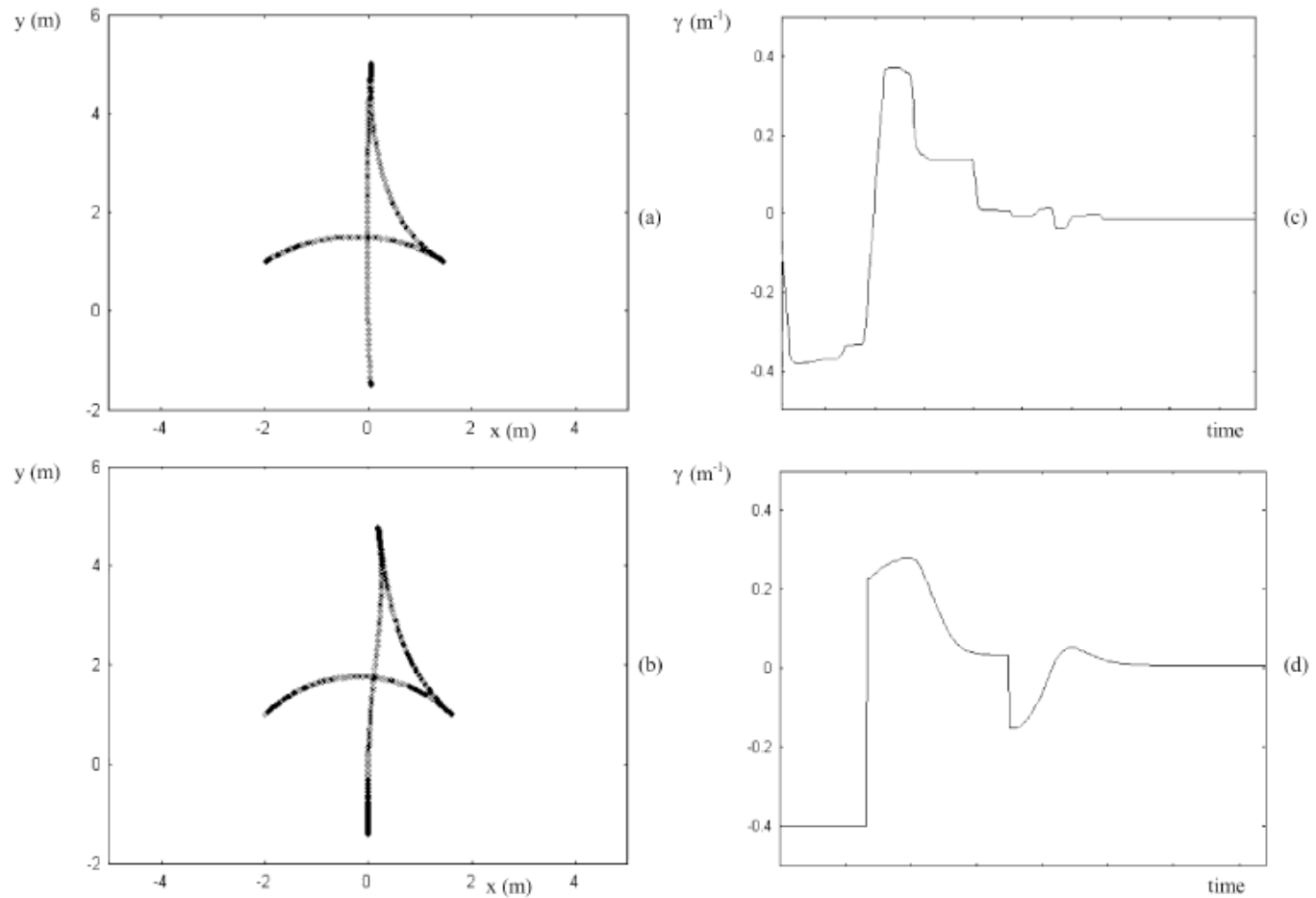


Fig. 29. (a) Experimental path of a parking maneuver. (b) Corresponding simulated results. (c) Experimental curvature transition for the path in (a). (d) Corresponding simulated results.

- "The experimental results obtained confirm that the designed control system meets its specifications: the robot is stopped at the parking target with the adequate orientation, collisions are avoided with the parking lot constraints, and short as well as continuous-curvature paths are generated during the forward and backward maneuvers."