3

Vision-Based Hierarchical Fuzzy Controller and Real Time Results for a Wheeled Autonomous Robot

Pourya Shahmaleki, Mojtaba Mahzoon,
Alireza Kazemi and Mohammad Basiri
Shiraz University
Iran

1. Introduction

One of the most important problems in robotics is motion planning problem, which its basic controversy is to plan a collision-free path between initial and target configurations for a robot. In the framework of motion planning for nonholonomic systems, the wheeled robots have attracted a significant amount of interest. The path planner of a wheeled autonomous robot has to meet nonholonomic constraints and then the movement direction must always be tangent to its trajectory (Paromtichk et. al., 1998; Latombe, 1991, Murray & Sastry, 1993; Lamiraux & Laumond, 2001; Scheuer & Fraichard, 1996). If no obstacles exist on path of the robot, then the robot task is finding the shortest path connecting two given initial and final configurations. The shortest paths for a car like vehicle consist of a finite sequence of two elementary components: arcs of circle (with minimum turning radii) and straight line segments. In any case, the problem is that the curvature is discontinuous between two elementary components, so that these shortest paths cannot be followed precisely without stopping at each discontinuity point to reorient the front wheels. To avoid these stops, several authors have proposed continuous-curvature path planners using differential geometric methods. These planners generate clothoids, cubic spirals, β-splines, quintic polynomials, etc., which are then followed by using a path-tracking technique based on, for example, pure-pursuit or predictive control methods (Lamiraux & Laumond, 2001; Scheuer & Fraichard, 1996). Stabilization issues of path-tracking methods for car-like vehicles using the Lyapunov method have been reported in (Walsh et. al., 1994; Tayebi & Rachid, 1996). One of the key technologies of future automobiles is the parking assist or automatic parking control. Control problems of a car-like vehicle are not easy because of the nonholonomic velocity constraints. The truck backer-upper control is a typical nonlinear control problem that cannot be solved by the conventional control techniques.

The goal of controller is to back up a truck to a loading dock from any initial position as quickly and precisely as possible. Backing a truck to the loading dock or parking spot is a difficult task even for a skilled truck driver. The research in parking problem is derived from the study of general motion planning for autonomous robots. In the past few decades, many algorithms have been developed for robot parking planning (Jiang & Seneviratne, 1999; Gomez-Bravo et. al., 2001; Cuesta et. al., 2004; Reeds & Shepp, 1990). The attempts to

Source: Motion Control, Book edited by: Federico Casolo,
ISBN 978-953-7619-55-8, pp. 580, January 2010, INTECH, Croatia, downloaded from SCIYO.COM

solve the truck backer-upper problem, rooted in computational intelligence, can be divided into two groups. The first group of methods seeks the solution through self tuning using neural networks, genetic algorithms or a combination of both. The second group of solutions, based on fuzzy logic, regards the controller as an emulator of human operator. The problem has become an acknowledged benchmark in non-linear control and as an example of a self-learning system in neural networks was proposed by Nguyen and Widrow in 1990 (Nguyen & Widrow, 1989). Careful experiments of their approach showed that the computational effort is very high (Kong & Kosko, 1990). Thousands (about 20000) of back-up cycles are needed before the network learns. Moreover the backpropagation algorithm does not converge for some sets of training samples. Numerous other techniques have been used, including genetic programming (Koza, 1992) Neuro-genetic controller (Schoenauer, & Ronald, 1994) and simplified neural network solution through problem decomposition (Jenkins & Yuhas, 1993). Very interesting contribution is (Tanaka et. al, 1998), where up to ten trailers can be controlled representing those as Takagi-Sugeno models and applying linear matrix inequalities method. A simplified version of the control problem has been extensively investigated in the field of fuzzy control (Ramamoorthy & Huang, 1991; Wang & Mendel, 1992; Ismail & Abu-Khousa, 1996; Kim, 1998; Dumitrache & Buiu, 1999). Also parking problem has been investigated by many researchers in the field of computational intelligence (for example; chang in (Chang et. al, 1995), Schoenauer in (Schoenauer & Ronald, 1994), Wang in (Wang & Mendel, 1992) and Li in (Li & Li, 2007)).

Fuzzy controllers, formulated on the basis of human understanding of the process or identified from measured control actions, can be regarded as emulators of human operators. Fuzzy logic control has more advantages because it can compensate the bad influence by nonlinearity and uncertainties based on advanced human expertise experience, also because it has strong robustness independent of a mathematical model. The other advantages of Fuzzy controllers are that their design is simple, fast, inexpensive, and easily maintained because the rules can be linguistically interpreted by the human experts. Riid & Rustern (Riid & Rustern, 2001) presented a fuzzy supervisory control system over the PID controller to reduce the complexity of the control problem and enhance the control performance. Riid & Rustern in (Riid & Rustern, 2002) demonstrate that problem decomposition leads to more effective knowledge acquisition and improved control performance in fuzzy control. The methodology allows solving complex control problems (truck backer-upper) without loss of functionality that is very difficult with all-in-one approaches and saves design expenses. Li & Chang in (Li & Chang, 2003) addressed the parking problem of a mobile robot by tracking feasible reference trajectories via a fuzzy sliding-mode control. Chen and Zhang in (Chen & Zhang, 1997) have reported a fuzzy controller to park a truck with suboptimal distance trajectories. They chose arcs of circle of minimum turning radii connected with parabolic curves as the optimal trajectories, but the desired parabolic curve to follow has to be given to the controller. More recently Li & Li in (Li & Li, 2007) have presented the fuzzy control system based on a hybrid clustering method and neural network. Sugeno & Murakami (Sugeno & Murakami, 1985) propose an experimental study on parking problem using model car, which is equipped with on-board microprocessor and two supersonic sensors for the measurements of the relative distance and direction. They derive fuzzy control rules by utilizing Sugeno-type fuzzy implications to model the parking experience of a skilled driver. Sugeno et al. (Sugeno et al, 1989) adopt the similar hardware arrangement as that in (Sugeno & Murakami, 1985) to execute the garage parking by employing fourteen fuzzy oral

instructions such as “go straight”, “slow down”, “enter garage” and “speed up”. Yasunobu & Murai (Yasunobu & Murai, 1994) exploit the state evaluation fuzzy control and the predictive fuzzy control to achieve the drive knowledge. Only the computer simulations are given to show the effectiveness of the proposed parking control. A skilled-based visual parking control using neural networks and fuzzy is discussed in (Daxwanger & Schmidt, 1995), where two control architectures, the direct neural control and the fuzzy hybrid control, are used to generate the automatic parking commands. The environment information is measured by a video sensor. The control architectures are validated by experiments with an autonomous mobile robot. Tayebi & Rachid (Tayebi & Rachid, 1996) deal with the parking problem of wheeled robot by using time-varying state feedback control law via the Lyapunov direct method. The control law is robust to ensure a global boundedness of the system states under measurement perturbations. The development of a near-optimal fuzzy controller for manoeuvring a car in a parking lot is described in (Leu & Kim, 1998). A cell mapping based method is proposed to systematically group near-optimal trajectories for all possible initial states in the parking lot.

The rules and membership functions of the fuzzy controller are generated using the statistical properties of the individual trajectory groups. An et al. (An et al., 1999) develop an online path-planning algorithm that guides an autonomous mobile robot to a goal with avoiding obstacles in an uncertain world. The established autonomous mobile robot cannot move omni-direction and run on two wheels equipped with a CCD camera. The path-planning algorithm is constructed by three modes: straight mode, spin mode, and avoidance mode. The simulation program and experimental results are developed to check this algorithm by using the garage parking motion. Shirazi & Yih (Shirazi & Yih, 1989) propose an expert’s knowledge including symbolic form and nonsymbolic form, where the former can be obtained from expert directly and the latter can be obtained only through an evolutionary process. The evolutionary process consists of three stages: novice, competency, and expert. The developed intelligent control system performs parallel parking to show validity and ability. The fuzzy traveling control of an autonomous mobile robot with six supersonic sensors has been provided in (Ohkita et al, 1993), where the flush problem is considered. The well-known fuzzy theory (Laumond et al, 1994) is now in widespread use such as system identification, function approximation, image compression, prediction, classification, and control. The general characteristic of the fuzzy control is that the IF-THEN fuzzy rules are on the basis of the conventional control strategy and the expert knowledge. It is shown in (Shahmaleki & Mahzoon, 2008; Shahmaleki et al, 2008) that hierarchical control system significantly improves control performance and reduces the design load compared to all-in-one approaches investigated by other researchers. Here, we recommend three approaches to solve parking problem. Finally we select and extend the hierarchical fuzzy control approach to the full truck backer-upper problem.

The path planners described in this research combine two fuzzy modules that provide desired angle value for front wheel so as to generate short paths with continuous curvatures. Approximated trajectories are composed of circular arcs of minimum turning radii and straight line segments. This chapter is structured as follows. Section 2 illustrates the parking problem addressed and its geometric and kinematical constraints. In Sections 3, Fuzzy control is described. In section 4 three control approaches described to solve parking problem. This section shows comparison between recommended approaches and the hierarchical structure of the fuzzy system is selected. Computer simulation results are given

to show the validity of the proposed fuzzy logic control algorithms. We have proposed a vision based approach in section 5 for estimating the robot position and direction. Some measures of the car-like robot is extracted from images that are captured using a ceiling mounted camera. These measurements together with the kinematic equations of the robot are used for estimating its position and direction using an extended Kalman filter. The control system has been made and tested on a mobile robot containing kinematics constraints. In Section 6, several experimental results of diagonal parking maneuvers are included to illustrate the efficiency and robustness of the designed controller. Finally, conclusions are given in Section 7.

2. The truck backing up problem

The problem addressed in this research is the diagonal parking of a truck in a constrained domain. The initial state of the truck position is represented by three state variables x, y and φ in Fig. 1. The truck kinematics model is based on the following system of equations (Li & Li, 2007):

$$\begin{cases} x(t+1) = x(t) - \cos(\varphi(t) + \theta((t)) - \sin(\theta(t))\sin(\varphi(t)), \\ y(t+1) = y(t) - \sin(\varphi(t) + \theta((t)) + \cos(\varphi(t))\sin(\theta(t)), \\ \varphi(t+1) = \varphi(t) - arcsin(\frac{2\sin(\theta(t))}{b}) \end{cases} \quad (1)$$

where (x,y) are the coordinates of the vehicle rear axle midpoint, φ is the truck orientation with respect to the horizontal line, b is the length of the truck and the control variable is the steering angle θ, that is the angle of the front wheel with respect to the truck. The truck only moves backward with fixed speed.

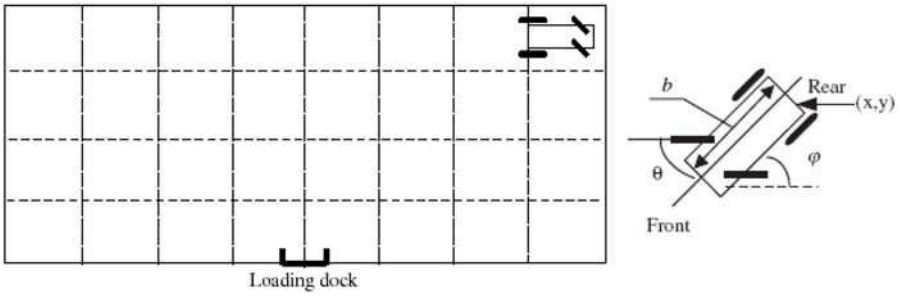


Fig. 1. Diagram of truck and loading dock

3. Fuzzy logic

A typical fuzzy control system consists of four components and the descriptions are stated as follows:

1. Fuzzification Interface: The fuzzification interface performs a conversion from a crisp point into a fuzzy set. The shapes of the membership functions of the linguistic sets are determined according to the expert experience.
2. Knowledge Base: The knowledge base commonly consists two sections: a database and a rule-base. The database contains the membership functions of the fuzzy sets used in

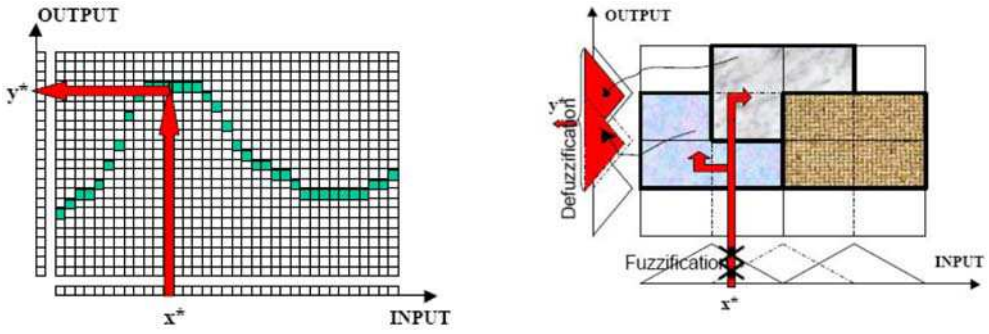
the fuzzy rules and the rule-base contains a number of fuzzy IF-THEN rules. The typical form of fuzzy rules can be expressed as IF precondition, THEN consequence. The canonical fuzzy IF-THEN rules are usually made from the following conditions: (a) Obtaining by the expert knowledge and/or operators experiences. (b) According to the control behavior of the users. (c) According to the characteristic of the plant. (d) Obtaining by self-learning.

3. Inference Engine: The inference engine that performs the fuzzy reasoning upon the fuzzy control rules is the main component of the fuzzy controller. There are varieties of compositional methods in fuzzy inference, such as max-min compositional operation and max-product compositional operation etc.
4. Defuzzification Interface: The defuzzification interface converts the fuzzy output of the rule-base into a non-fuzzy value. The center of area (COA) is the often used method in defuzzification. Suppose $\widetilde{B}$ is a discrete set as $\widetilde{B} = \{y_1, y_2, ..., y_n\}$ then COA method can be described as:

$$y^* = \sum_{k=1}^{n} y_k \cdot \mu_{\widetilde{B}}(y_k) \Big/ \sum_{k=1}^{n} \mu_{\widetilde{B}}(y_k) \tag{2}$$

where y* is the crisp value defuzzified from COA.

In this section we illustrate a brief comparison between Classic control and Fuzzy control. Classic control is based on a detailed I/O function OUTPUT= F (INPUT) which maps each high-resolution quantization interval of the input domain into a high-resolution quantization interval of the output domain. Finding a mathematical expression for this detailed mapping relationship F may be difficult, if not impossible, in many applications (Fig 2(a)). But, Fuzzy control is based on an I/O function that maps each very low-resolution quantization interval of the input domain into a very low-low resolution quantization interval of the output domain. As there are only 7 or 9 fuzzy quantization intervals covering the input and output domains the mapping relationship can be very easily expressed using the "if-then" formalism. (In many applications, this leads to a simpler solution in less design time.) The overlapping of these fuzzy domains and their linear membership functions will eventually allow achieving a rather high-resolution I/O function between crisp input and output variables (Fig 2(b)).



(a) (b)

Fig. 2. Comparison between Classic and Fuzzy controls

In Fig 3 structure of fuzzy control is shown.

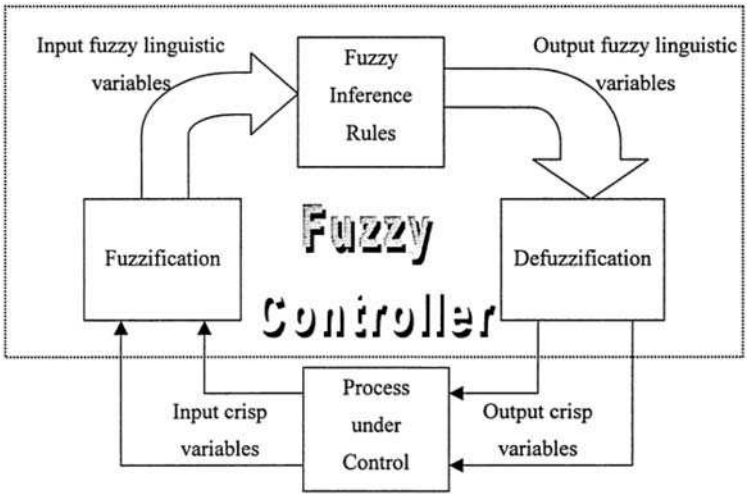


Fig. 3. Structure of Fuzzy control system

4. Designing fuzzy control system

The truck and the loading zone are shown in Fig. 4. The truck has two front steering wheels and two rear driven wheels that cannot move sideways. The coordinate pair (x,y) specifies the rear center position of the truck in the plane. The angle φ increases from -90 toward 270 in a clockwise direction and the steering angle θ is taken as positive if the steering wheel is turned to the right and negative, otherwise. The loading zone is the plane x:[-25,25], y:[0,25]. The goal of this research is to design a Fuzzy Logic Controller (FLC) able to back up the truck into a docking situation from any initial position that has enough clearance from the docking station. The controller should produce the appropriate steering angle θ = [-40°, 40°] at every stage to make the truck back up to a configuration with x=0, y=0, φ=90 (that is the desired parking space) from any initial position (x, y and φ) and to stop there. Thus controller is a function of state variables:

$$\theta = f(x,y,\varphi), \tag{3}$$

The y coordinate is not used because the straight segments of approximated trajectories are always horizontal. Also typically it is assumed that enough clearance between the truck and the loading dock exists so that the truck y-position coordinate can be ignored, simplifying the controller function to:

$$\theta = f(x,\varphi), \tag{4}$$

Hence only x-position and truck orientation angle φ are inputs of the fuzzy controller and the steering angle θ is the output.

As shown in Fig. 4, the suboptimal goal is that the backward driving involves short trajectories made up of arcs of circle of minimum turning radii and straight line segments, which meet the kinematic constraints in (1).

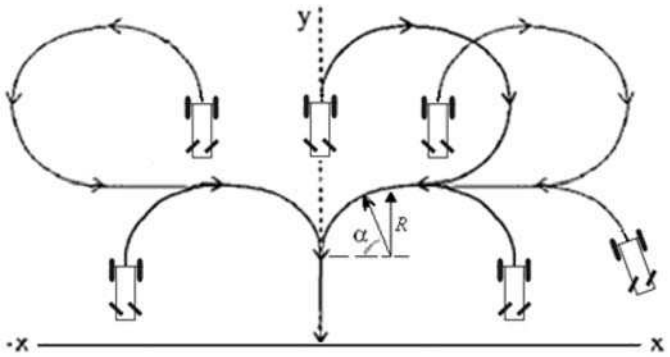


Fig. 4. Ideal trajectories

Analyzing the shortest paths geometrically, a mathematical expression for the steering angle θ which produces curvature of these short paths can be found:

$$\theta = \begin{cases} 40° & if \quad \varphi > \alpha \\ 0° & if \quad \varphi = \alpha \\ -40° & if \quad \varphi < \alpha \end{cases} \tag{5}$$

where the angle α (depends on x)associated with the switching in the steering angle(θ) sign can be calculated as follows:

$$\alpha = \begin{cases} sign(x) \cdot \cos^{-1}(\frac{R-|x|}{R}) & if \quad |x| < R \\ sign(x) \cdot \frac{\pi}{2} & if \quad |x| \geq R \end{cases} \tag{6}$$

R being the minimum turning radius corresponding to the maximum curvature (γ) which has a constant value (γ =1/R).

4.1 Integrated approach

Structure of this approach consists of a single module with two inputs (x and φ) and one output (θ) (Fig. 5). It contains five linguistic labels to cover the input variable x and seven labels for the vehicle angle φ (Fig. 6).

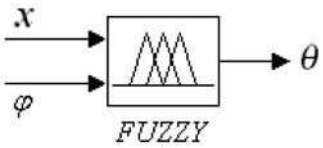


Fig. 5. Structure of fuzzy controller

One triangular and four trapezoidal membership functions (LE, LC, CE, RC, RI) are selected to cover the x variable. Also five triangular and two trapezoidal membership functions (LB, LU, LV, VE, RB, RU, RV) are selected to cover the φ variable.

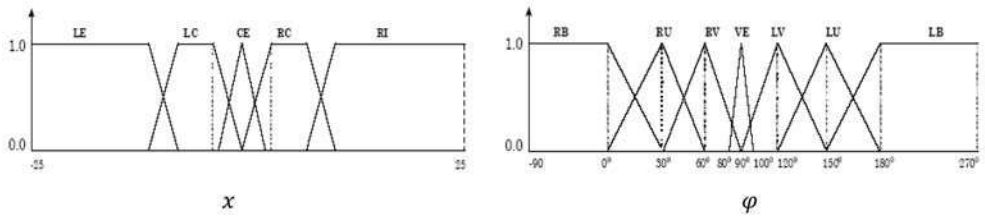


Fig. 6. Mebership functions for x and φ variables (Integrated approach)

The rules are shown in Table 1. Also the "center of gravity" method is used as defuzzification method. The rule base implements a zero-order Takagi–Sugeno inference method.

X / φ	LE	LC	CE	RC	RI
RL	NL	NL	NM	NM	NS
RU	NL	NL	NM	NS	PS
RV	NL	NM	NS	PS	PM
VE	NM	NM	ZE	PM	PM
LV	NM	NS	PS	PM	PL
LU	NS	PS	PM	PL	PL
LL	PS	PM	PM	PL	PL

Table 1. The learned rules for the x and φ variables

The consequents of the rules are the following: ZE=0°, NL=-40°, NM=-30°, NS=-20°, PL=40°, PM=30°, PS=20°.

4.2 Combined approach

As shown in Fig. 7, structure of the controller consists of a fuzzy module and three blocks (Dis, Controller1 and Controller2).

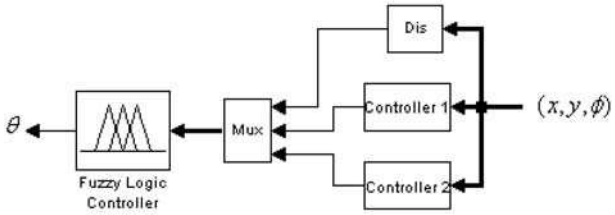


Fig. 7. Structure of the control system

In fact, if we try to find a mathematical expression for the steering angle θ which produces curvature of these short paths, we can recommend equations (7),(8). The angle of wheels (θ1, θ2) is computed in Controller1 and Controller2 based on the equation (7), (8).

$$\begin{aligned} \gamma &= ar\cos(x\sqrt{x^2+y^2}) \\ rel &= \phi - \gamma \\ \beta &= rel - round(\frac{rel}{2\pi})\times 2\pi \\ \theta 1 &= a\gamma + b\beta + c \end{aligned} \tag{7}$$

$$\theta_1 = \begin{cases} \theta_1 & if \quad -\pi/4 < \theta_1 < \pi/4 \\ \pi/4 & if \quad \theta_1 \geq \pi/4 \\ -\pi/4 & if \quad \theta_1 \leq -\pi/4 \end{cases}$$

$$\gamma = x$$

$$rel = \phi - \frac{\pi}{2}$$

$$\beta = rel - round(\frac{rel}{2\pi}) \times 2\pi \qquad (8)$$

$$\theta 2 = m\gamma + n\beta + p$$

$$\theta_2 = \begin{cases} \theta_2 & if \quad -\pi/4 < \theta_2 < \pi/4 \\ \pi/4 & if \quad \theta_2 \geq \pi/4 \\ -\pi/4 & if \quad \theta_2 \leq -\pi/4 \end{cases}$$

where a, b, c, m, n and p are constant values.
The distance between the vehicle rear axle midpoint and constrained domain is computed by:

$$\text{distance} = \sqrt{(x - x_0)^2 + (y - y_0)^2} \qquad (9)$$

Two S-shaped and Z-shaped membership functions (far, near) are selected to cover the distance universe of discourse (Fig. 8). The consequent of the rules (out1, out2) are singletons. The rules are the following:

1. If distance = near → θ = out1
2. If distance = far → θ = out2

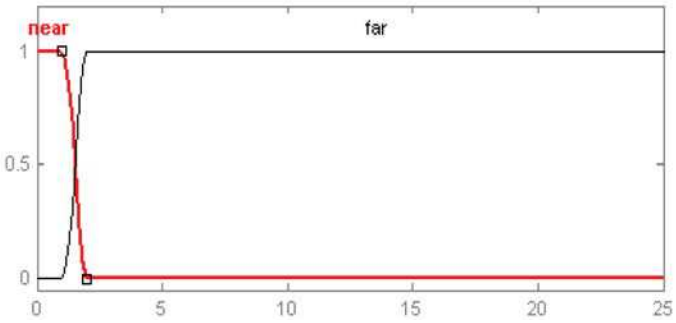
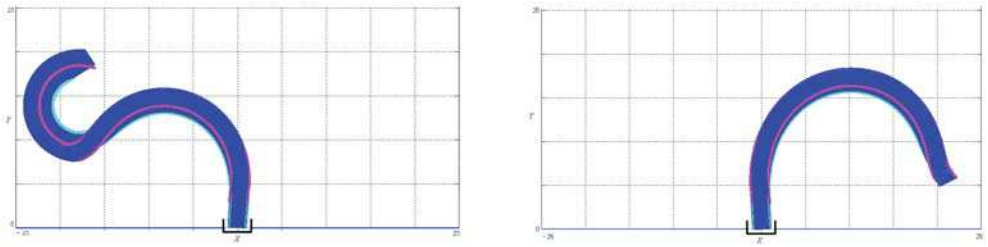


Fig. 8. Membership functions for distance variables

Fig. 9 shows three examples of the generated paths by combined approach.

4.3 Integrated approach

In this section the hierarchical structure is introduced .The scheme is basically made up of two rule bases (Fig. 10).



(a) (b)

Fig. 9. Simulated results of the parking maneuvers corresponding to the initial configurations (a) x=-15, y=18, φ =180. (b) x=20, y=9, φ =228

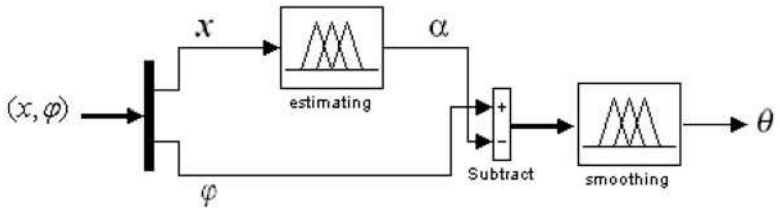


Fig. 10. Structure of hierarchical approach

The first one ("estimating") provides approximately the value of the angle α depending on the input variables x. The second one ("smoothing") provides the desired value for steering angle (θ) depending on the value of difference φ-α. Two triangular and two trapezoidal membership functions (LB, LS, RS and RB) are selected to cover the x universe of discourse and four rules are included in the rule base "estimating". The consequents of the rules (mf1, mf2, mf3, and mf4) are singletons. Also the "center of gravity" method is used for defuzzification.

The rule bases (estimating and smoothing) implements a zero-order Takagi–Sugeno inference method.

The rules are:

1. if (x = LB) → α= mf1
2. if (x = LS) → α= mf2
3. if (x = RS) → α= mf3
4. if (x = RB) → α= mf4,

The rule base "smoothing" also contains two triangular and two trapezoidal membership functions and four rules. The rules are:

1. if (diff = MZ) → θ= nf1
2. if (diff = NZ) → θ= nf2
3. if (diff = PZ) → θ= nf3
4. if (diff = RZ) → θ= nf4,

where MZ, NZ, PZ and RZ are fuzzy sets represented by triangular and trapezoidal membership functions (they cause the smooth switching in the steering angle θ when φ is

around α). nf1, nf2, nf3, nf4 are singleton values associated with the angle front wheels. The membership functions for the variables x and diff are shown in Fig. 11.

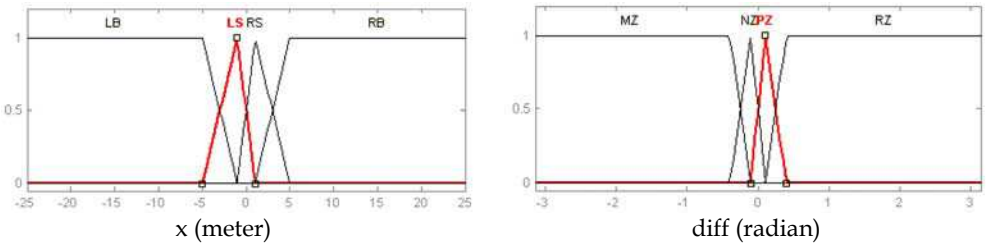


Fig. 11. Membership functions for x and diff variables

The "estimating" module for the hierarchical approach provides a fuzzy approximation for the angle α. The advantage of using this module instead of giving α analytically is that the required computational cost is reduced. Using normalized triangular and trapezoidal membership functions for the antecedents of the rules and a zero-order Takagi–Sugeno inference engine makes this approximation piecewise linear, which means that only several additions and products need to be implemented. The computational cost of additions and products is less than that of a nonlinear function such as Arcos (.) in (6).

Fig. 12 shows the variations of θ versus x and φ corresponding to (5) and (6). These equations are associated with an on–off control because the θ value presents abrupt changes, and would require stopping the robot to perform this switching.

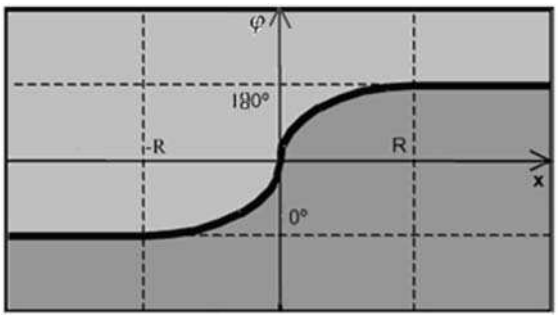
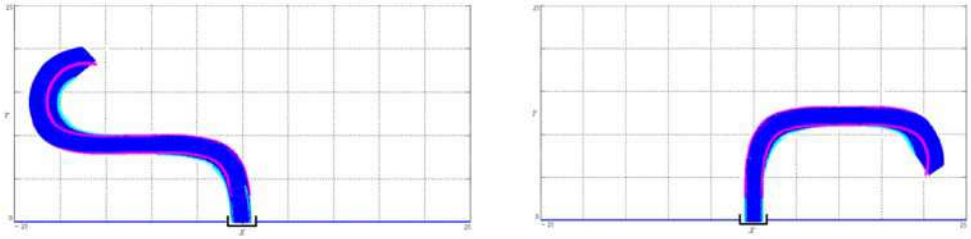


Fig. 12. Steering angle θ versus x and φ for short paths. The dark color presents the θ=-40°and the light one presents θ=40°

Three fuzzy modules (integrated, combined and hierarchical) described previously are zero-order Takagi–Sugeno systems whose input membership functions always overlap each other. Hence, the subgoal of providing continuous-curvature and short paths is achieved.

Comparing the three approaches for designing the controller the hierarchical one is more efficient since it generates paths but with small number of rules. Besides it provides the higher smoothness near the target configuration (x=0). As a result, the hierarchical module was selected as the control system.

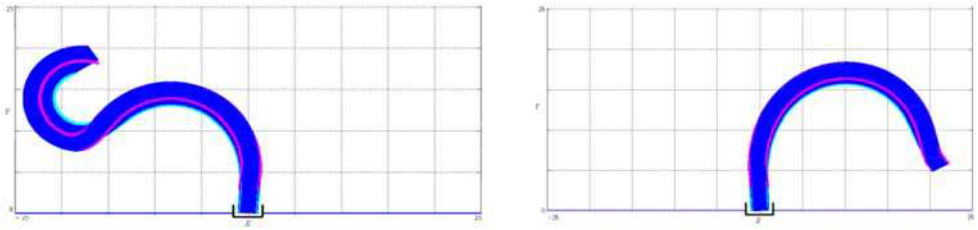
Simulated results using the present hierarchical scheme for the different initial positions are shown in Fig. 13. In this figure, t indicates the parking duration. It can be seen how the generated paths (Fig. 13) are very close to the ideal paths (Fig. 4) made up of circular arcs and straight lines.



(a) (b)

Fig. 13. Results of the parking maneuver corresponding to the initial configurations (a) x=-20, y=18.4, φ=120°, t=78 steps, (b) x=17.5, y=8, φ=252°, t=72 steps

Further, according to the robot kinematics equations, the work of Li and Li (Li & Li, 2007) has been used for comparison. Fig.14 shows simulated results of Li and Li (Li & Li, 2007) for the same initial conditions of Fig.13.



(a) (b)

Fig. 14. Results of the parking maneuver corresponding to the initial configurations (a) x=-20, y=18.4, φ=120°, t=93 steps, (b) x=17.5, y=8, φ=252°, t=86 steps, (Li & Li, 2007)

An advantage of this approach is that the rules are linguistically interpretable and the controller generates paths with 8 rules compared with 35 used by (Riid & Rustern, 2002). Besides it provides the higher smoothness near the target configuration (x=0). Also, parking durations are shorter than those obtained by (Li & Li, 2007) under the same initial conditions. In this work, trajectories are composed of circular arcs and straight segments but in other methods, trajectories are composed of circular arcs.

5. Real time experimental studies

As shown in Fig. 15(a), the designed mobile robot has a 30cm×20cm×10cm, aluminium body with four 7cm diameter tires. It contains an AVR-ATMGEA64 micro controller, running at 16 MHz clock. The robot is equipped with three 0.9 degree stepper motors, two for the back wheels and one guides the steering through a gear box. The control of the mobile robot

motion is performed on two levels, as demonstrated in Fig. 15(b). This two-layer architecture is very common in practice because most mobile robots and manipulators usually do not allow the user to impose accelerations or torques at the inputs. It can also be viewed as a simplification to the problem as well as a more modular design approach. The high level control (Hierarchical Fuzzy Controller) determines the steering angle θ of the robot considering the position (x,y) and angle (φ) of the robot which is received from the vision system. While the low level controller receives the output of high level control and determines steering angle of the front wheel and the speed of two rear wheels differentially.

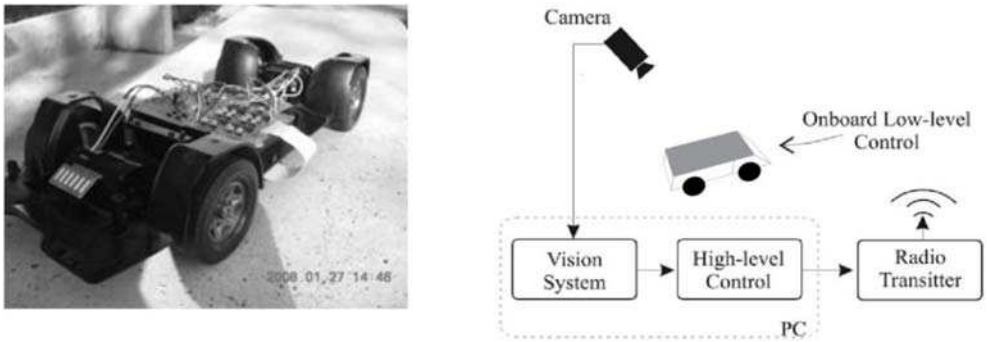


Fig. 15. (a) Designed mobile robot. (b) The control architecture of the mobile robot

The structure of real control system is shown in Fig. 16.

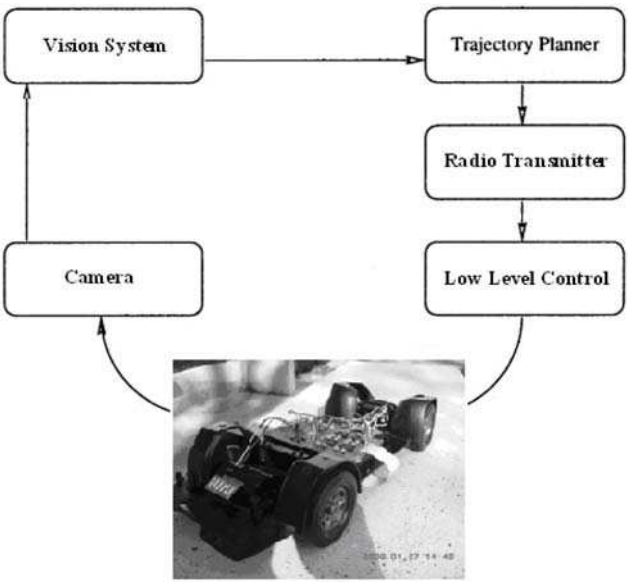


Fig. 16. The structure of real control system

5.1 Vision subsystem

For the backer-upper system to work in a real environment it is necessary to obtain the car position and orientation parameters. For this task different sensing and measuring instruments have been used in the literature. Some authors (Demilri & Turksen, 2000) have used sonar to identify the location of the mobile robots in a global map. This is achieved by using fuzzy sets to model the sonar data and by using the fuzzy triangulation to identify the robots position and orientation. Other authors have used analogue features of RFID tags system (Miah & Gueaieb, 2007) to locate the car-like mobile robot. Vision based position estimation has been also used for this task. In (Chen & Feng, 2009) a hardware implemented vision based method is used to estimate the robot position and direction. They use a camera mounted on the mobile robot and estimate the car-like robot position and direction using profiles of wavelet coefficients of the captured images and using of a self organizing map neural network. Each neuron categorizes measurements of a location and direction bin. This method is limited in that it works based on recognizing the part of parking that is in the view field of robot's camera. This parking view classification based approach, requires new training if the parking space is changed. Also it has not the potential for localizing free parking lots and other robots or obstacles which may be required in real applications.

A ceiling mounted camera can provide a holistic view to the location. Using a CCD camera as measuring device to capture images from parking area, and using image processing and tracking algorithms, we can estimate position and direction of the object of interest. This approach can be used in multi-agent environments to localize other objects and obstacles and even free parking lot positions. Here we assume just one robot and no obstacles. Also, we assume that the camera has been installed on the ceiling in the center of parking zone and at a proper height such that we can ignore perspective effects at corners of the captured images. Thus a linear calibration can be used for conversion between the (i, j) pixel indices in the image and the (x, y) coordinates of the parking zone. This assumption can introduce some approximation errors. As will be described here, using a prior knowledge of the car kinematic in an extended Kalman filtering framework can correct these measurement errors. With this configuration and assumptions a simple non realistic solution for position and direction estimation can be used as follows. Set two different color marks on top of the car in middle front and rear wheels position. Then from the captured image extract the two colored marks and find their center. Assume (x_r, y_r) and (x_f, y_f) be coordinates of middle rear and front points then (x, y) input variables of the fuzzy controller can be estimated from (x_r, y_r) after some calibration. The direction φ of the car-like robot relative to x-axis can also be determined using:

$$\varphi = \tan^{-1}\frac{(y_f - y_r)}{(x_f - x_r)} \tag{10}$$

Note that the $tan^{-1}(.)$ function used here should consider signs of y_f-y_r and x_f-x_r terms so that it can calculate the direction in the range $[0,2\pi]$ or equivalently $[-\pi, \pi]$. Such a function in most programming environments is commonly named $atan2(.,.)$ which perceives y_f -y_r and x_f - x_r separately and calculates the true direction accordingly.

This is a simple solution for non-realistic experimental conditions. However it is necessary to consider more realistic applications of the backer-upper system. So we should eliminate strong non-realistic constraints like hand marking the car with two different color marks.

Here we propose a method based on Hough transform for extracting measurements to estimate car position and orientation parameters. Using Hough transform we can just extract the orientation from the border lines of the car, but the controller subsystem needs the direction φ in range $[-\pi, \pi]$ to calculate correct steering angle. To find the true direction we use a simple pattern classification based method to discriminate between front and rear sides of the car-like robot from its pixel gray values. This classifier trains the robots image and is independent of the parking background. Also it can be trained to work for different moving objects.
We can use extracted measurements of each frame to directly estimate (x, y, φ) state variables. But since extracted measurements are not accurate enough, we use these measurement parameters together with kinematic equations (1) of the plant as a state transition model in an extended Kalman filter to estimate the state variables (x, y, φ) of the robot more accurately.

5.2 Car position extraction using Hough transform

Hough transform (HT) first proposed by Hough (Hough, 1962) and improved by Duda & Hart (Duda & Hart, 1972) is a feature extraction method which is widely used in computer vision and image processing. It converts edge map of an image into a parametric space of a given geometric shape. Edge map can be extracted using edge extraction methods which filter the image to extract high frequency parts (edges) and then apply a threshold to get a binary matrix. HT tries to find noisy and imperfect examples for a given shape class within an image. There exists HTs for lines, circles and ellipses.
For example classic Hough transform, finds lines in a given image. A line can be parameterized in the Cartesian coordinate by slope (m) and interception (b) parameters (Hough, 1962). Each point (x, y) of the line can be constrained by the equation $y = mx + b$. However this representation is not well-formed for computational reasons. The slope of near vertical lines, go to infinity hence it is not a good representation for all possible lines. The classic Hough transform proposed by Duda and Haart (Duda & Hart, 1972) uses a polar representation in which lines are shown by two parameters r and θ in the polar coordinate. Parameter r is length of the vector started from origin and perpendicularly connected to the line (distance of line to the origin) and θ is the angle between that vector and x axis.
Classic Hough transform calculates a 2D parameter map matrix for quantized values of (r,θ) parameters. An algorithm determines lines with (r,θ) values that pass through each edge point of the image and increases votes of those (r,θ) bins in the matrix. For each edge point this accumulation is carried out. Finally the peaks in the parameter map show the most perfect lines that exist in the image. The following equation relates the (x,y) Cartesian coordinate of line points with the r,θ polar line parameters, as previously defined.

$$\begin{cases} y = -\dfrac{\cos\theta}{\sin\theta}\,x + \dfrac{r}{\sin\theta} \\ r(\theta) = x\cos\theta + y\sin\theta \end{cases} \tag{11}$$

For any edge point (x_i,y_i), equation (11) provides a sinusoidal curve in terms of r and θ parameters. Points on this curve determine all lines (r_j,θ_j) that pass through the edge point (x_i,y_i). For each edge point votes of all cells of the parameter matrix that fall on the corresponding sinusoidal curve are increased.

The external boundary of the car-like robot is approximated by a rectangle. To extract four lines of this rectangle in each input image frame, first calculate the edge map of the image using an edge extraction algorithm. Then apply Hough transform and extract dominant peaks of the parameter map. Then among these peaks we search to select four lines that satisfy the constraints of being edges of a rectangle corresponding to car-like robot size. Four selected lines should approximately form a a×b rectangle where a and b are width and length of the car-like robot.

Let the four selected lines have parameters (r_i,θ_i), i = 1,2,3,4. In order to extract the rectangle formed by these four lines, four intersection points (x_j,y_j), j = 1,2,3,4 of perpendicular pairs should be calculated. Solving for the linear system in equation (12), intersection point (x_0,y_0) of two sample lines (r_1,θ_1) and (r_2,θ_2) can be determined.

$$\begin{cases} x_0 \cos\theta_1 + y_0 \sin\theta_1 = r_1 \\ x_0 \cos\theta_2 + y_0 \sin\theta_2 = r_2 \end{cases} \tag{12}$$

If the lines are not parallel, the unique solution is given by equation (13).

$$\begin{cases} x_0 = \dfrac{r_1 \sin\theta_2 - r_2 \sin\theta_1}{\sin(\theta_2 - \theta_1)} \\ y_0 = \dfrac{-r_1 \cos\theta_2 + r_2 \cos\theta_1}{\sin(\theta_2 - \theta_1)} \end{cases} \tag{13}$$

A problem with HT is that it is computationally expensive. However its complexity can be reduced since position and orientation of the robot is approximately known in the tracking procedure. Thus HT just should be calculated for a part of the image and a range of (r,θ) around current point. Also the level of quantization of (r,θ) can be set as large as possible to reduce the time complexity. Relative coarse bin sizes for (r,θ) also help to cope with little curvatures in the border lines of the car-like robot. This is at the expense of reducing the estimated position and direction resolution. The relative degraded resolution of (r,θ) due to coarse bin sizes can be restored by the correction and denoising property of Kalman filter. Note that the computation complexity of Kalman filter is very low relative to HT, since the former manipulates very low dimensional extracted measurements while the latter manipulates high dimensional image data.

5.3 Determining car direction using classification

Using equation (13), four corners of the approximately rectangular car border can be estimated. Now it is necessary to specify which pair of these four points belongs to the rear and which pair belongs to the front side of the car. We can not extract any information from Hough transform about the rear-front points assignment. But this assignment is required to determine middle rear wheels points (x_r,y_r) and also the signed direction φ of the car.

To solve this problem we adopt a classification-based approach. For each frame, using the four estimated corner points of the car, a rectangular area of $n_a \times n_b$ pixels of the car-like object is extracted. Then extracted pixels are stacked in a predefined order to get a $n_a \times n_b$ feature vector. A classifier that is trained using training data, is used to determine the direction using these feature vectors. However, due to large number of features, it is necessary to apply a feature reduction transformation like principle component analysis

(PCA) or linear discriminant analysis (LDA) before the classification (Duda et al, 2000). These linear feature transforms reduce the size of feature vectors by selecting most informative or discriminative linear combinations of all features. Feature reduction, reduces the classifier complexity hence the amount of labeled data that is required for training the classifier. Different feature reduction and classifier structures can be adopted for this binary classification task. Here we apply PCA for feature reduction and a linear support vector machine for classification task. Supprot Vector Machine (SVM) proposed by Vapnik (Vapnik, 1995) is a large margin classifier based on the concept of structural risk minimization. SVM provides good generalization capability. Its training, using large number of data, is time consuming to some extent, but for classification it is as fast as a simple linear transform. Here we use SVM because we want to create a classifier with good generalization and accuracy, using small number of training data.

LDA is a supervised feature transform and provides more discriminative features relative to PCA hence it is commonly preferred to PCA. But the simple LDA reduces the number of features to at most $C - 1$ features where C is number of classes. Since our task is a binary classification, hence using LDA we just would get one feature that is not enough for accurate direction classification. Thus we use PCA to have enough features after feature reduction. To create our binary direction sign classifier, first we train the PCA transform. To calculate principle components, mean and covariance of feature vectors are estimated then eigen value decomposition is applied on the covariance matrix. Finally N eigen vectors with greater corresponding eigen values, are selected to form the transformation matrix W. This linear transformation reduces dimension of feature vectors from $n_a \times n_b$ to N elements. Here in experiments $N = 10$ eigen values provides good results.

To train a binary SVM, reduced feature vectors with their corresponding labels are first normalized along each feature by subtracting the mean and dividing by the standard deviation of that feature. About 100 training images are sufficient. These examples should be captured in different points and directions in the view field of the camera. The car pixels extracted from each training image, can be resorted in two feature vectors one from front to rear which takes the label -1 and one from rear to front which takes the label +1. In the training examples position of the car and its pixel values are extracted automatically using Hough transform method described in previous section. But the rear-front labeling should be assigned by a human operator. This binary classification approach provides accuracy higher than 97% which is completely reliable. Because the car motion is continuous, we can correct possible wrong classified frames using previous frames history.

Using this classification method the front-rear assignment of the four corner points of the car is determined. Now Corner points are sorted in the following defined order to form an 8 dimensional measurement vector $Y^I = [x_{r_1}, y_{r_1}, x_{r_2}, y_{r_2}, x_{f_1}, y_{f_1}, x_{f_2}, y_{f_2}]^T$. The r_1, r_2, f_1, f_2 subscripts denote in order, the rear-left, rear-right, front-left and the front-right corners of the car.

From the four ordered corner points in the measurement vector Y^I, we can also directly calculate an estimate of the car position state vector to form another measurement vector $Y^D = [x_r, y_r, \varphi_{rf}]^T$ where (x_r, y_r) is the middle rear point coordinate and φ_{rf} is the signed direction of rear to front vector of the car-like robot relative to the x-axis. The superscripts D and I in these two measurement vectors show that they are directly or indirectly related to the state variables of the car-like robot that is required in the fuzzy controller. The measurement vector Y^D can be determined from measurement vector Y^I using equation (14).

$$Y^D = \begin{pmatrix} x_r \\ y_r \\ \varphi_{rf} \end{pmatrix} = \begin{pmatrix} (x_{r_1} + x_{r_2})/2 \\ (y_{r_1} + y_{r_2})/2 \\ \tan^{-1} \dfrac{(y_{f_1} + y_{f_2}) - (y_{r_1} + y_{r_2})}{(x_{f_1} + x_{f_2}) - (x_{r_1} + x_{r_2})} \end{pmatrix} \tag{14}$$

In the next section we will illustrate a method for more accurate estimation of state parameters by filtering these inaccurate measurements in an extended Kalman filtering framework.

5.4 Tracking the car state parameters with extended Kalman filter

Here we illustrate the simple and extended Kalman filters and their terminology and then describe our problem formulation in terms of an extended Kalman filtering framework.

5.4.1 Kalman filter

The Kalman filter (Kalman, 1960) is an efficient Bayesian optimal recursive linear filter that estimates the state of a time discrete linear dynamic system from a sequence of measurements which are perturbed by Gaussian noise. It is mostly used for tracking objects in computer vision and for identification and regulation of linear dynamic systems in control theory. Kalman filter considers a linear relation between measurements Y and state variables X of the system that is commonly named as the observation model of the system. Another linear relation is considered for state transition, between state variables in time step t, X_t and in time step t-1, X_{t-1} and the control inputs u_t of the system. These linear models are formulated as follows:

$$\begin{cases} X_t = F_t X_{t-1} + B_t u_t + w_t, & w_t \sim N(0, Q) \\ Y_t = H_t X_t + v_t, & v_t \sim N(0, R) \end{cases} \tag{15}$$

In equation (15), F_t is the dynamic model, B_t is the control model, w_t is the stochastic process noise model, H_t is the observation model, v_t is the stochastic observation noise model and u_t is the control input of the system. Kalman filter considers the estimated state $\widehat{X}$ as a random vector with Gaussian distribution and a covariance matrix P. In following equations the notation $\hat{X}_{i|j}$ is used for the estimated state vector in time step i by using measurement vectors up to time step j.

The prediction estimates of state are given in equation (16), where $\hat{X}_{t|t-1}$ is the predicted state and $P_{t|t-1}$ is the predicted state covariance matrix. Note that in the prediction step just the dynamic model of the system is used to predict what would be the next state of the system. The prediction result is a random vector so it has its covariance matrix with itself.

$$\begin{cases} \hat{X}_{t|t-1} = F_t \hat{X}_{t-1|t-1} + B_{t-1} u_{t-1} \\ P_{t|t-1} = F_t P_{t-1|t-1} F_t^T + Q_{t-1} \end{cases} \tag{16}$$

In each time step before the current measurement is prepared we can estimate the predicted state then we use the acquired measurements from the sensors to update our predicted belief according to the error. The updated estimates using the measurements are given in

equation (17). In this equation, Z_t is the innovation or prediction error, S_t is the innovation covariance, K_t is the optimal Kalman gain, $\hat{X}_{t|t}$ is the updated estimate of system state and $P_{t|t}$ is the updated or posterior covariance of the state estimation in time step t. The Kalman gain balances the amount of contribution of dynamic model and the measurement to the state estimation, according to their accuracy and confidence.

$$\begin{cases} Z_t = Y_t - H_t \hat{X}_{t|t-1} \\ S_t = H_t P_{t|t-1} H_t^T + R_t \\ K_t = P_{t|t-1} H_t^T S_t^{-1} \\ \hat{X}_{t|t} = \hat{X}_{t|t-1} + K_t Z_t \\ P_{t|t} = (I - K_t H_t) P_{t|t-1} \end{cases} \tag{17}$$

In order to use Kalman filter in a recursive estimation task we should specify dynamic and observation models F_t, H_t and some times the control model B_t. Also we should set initial state $\hat{X}_{0|0}$ and its covariance $P_{0|0}$ and prior process noise and measurement noise covariance matrices Q_0, R_0.

5.4.2 Extended Kalman filter

Kalman filter proposed in (Kalman, 1960) has been derived for linear state transition and observation models. These linear functions can be time variant that result in different F_t and H_t matrices in different time steps t. In extended Kalman filter (Bar-Shalom & Fortmann, 1988), the dynamic and observation models are not required to be linear necessarily. The models just should be differentiable functions.

$$\begin{cases} X_t = f(X_{t-1}, u_t) + w_t, & w_t \sim N(0, Q) \\ Y_t = h(X_t) + v_t, & v_t \sim N(0, R) \end{cases} \tag{18}$$

Again w_t and v_t are process and measurement noises which are Gaussian distributions with zero mean and Q, R covariance matrices.
In extended Kalman filter functions f (.) and h (.) can be used to perform prediction step for state vector $\hat{X}_{t|t-1}$ but for prediction of covariance matrix $P_{t|t-1}$and also in the update step for updating state and covariance matrix we can not use this non-linear functions. However, we can use a linear approximation of these non linear functions using the first partial derivatives around the predicted point $\hat{X}_{t|t-1}$. So for each time step t, Jacobian matrices of functions f (.) and h (.), should be calculated and used as linear approximations for dynamic and observation models in that time step.

5.5 Applying extended Kalman filter for car position estimation

Now we illustrate the dynamic and observation models to be used in the extended Kalman filtering framework. The dynamic model should predict the state vector $X_t = [x_t, y_t, \varphi_t]^T$ from existing state vector $X_{t-1} = [x_{t-1}, y_{t-1}, \varphi_{t-1}]^T$ and the control input to the car-like robot which is the steering angle θ_{t-1}. This is just the kinematic equations of the car-like robot that is given in equation (1). This equation considers unit transition velocity between time steps. This should be replaced with a translation velocity parameter V that is unknown. It can be embedded as an extra state variable to X to form the new state vector $X^v = [X; V]$ or may be

left as a constant. The state transition function for the new state vector used here is given in equation (19).

$$X_t^V = \begin{bmatrix} x_t \\ y_t \\ \varphi_t \\ V_t \end{bmatrix} = f(X_{t-1}^V, \theta_{t-1}) = \begin{bmatrix} x_{t-1} - V_{t-1}[\cos(\varphi_{t-1} + \theta_{t-1}) - sin(\theta_{t-1})\cos(\varphi_{t-1})] \\ y_{t-1} - V_{t-1}[\sin(\varphi_{t-1} + \theta_{t-1}) + cos(\varphi_{t-1})\sin(\theta_{t-1})] \\ \varphi_{t-1} - V_{t-1} \arcsin(2sin(\theta_{t-1})/b) \\ V_{t-1} \end{bmatrix} \quad (19)$$

The observation model should calculate measurements from current state vector. As we have considered two measurements $Y^I = [x_{r_1}, y_{r_1}, x_{r_2}, y_{r_2}, x_{f_1}, y_{f_1}, x_{f_2}, y_{f_2}]^T$ and $Y^D = [x_r, y_r, \varphi]^T$, we would have two observation models correspondingly. First observation model is a nonlinear function $Y^I = h_I\,(X_t^V)$ since its calculation of it requires some $cos(\varphi)$ and $sin(\varphi)$ terms. The second observation model is an identity function $Y_t^D = h_D\,(X_t^V) = X_t$ that is $H_t = I_{3\times4}$. To prevent complexity we used the direct measurement vector hence identity observation model. Now the extended Kalman filter can be set up. Initial state vector can be determined from Y_0^D that is extracted from first frame the velocity can be set to 1 for initial step. Update steps of the filtering will correct the speed. The Initial state covariance matrix and process and measurement noise covariance matrices are initialized with diagonal matrices that contain estimations of variance of corresponding variables.

For each input frame first the predicted state is calculated using prediction equations and state transition function (19), then HT is computed around current position and direction and best border rectangle is determined from extracted lines, then signed direction is determined using the classification. Then measurement Y_t^D is calculated. Finally we use this measurement vector to update the state according to extended Kalman filter update equations. Then x_t, y_t, φ_t values of the updated state parameters are passed to the high level fuzzy control to calculate the steering angle θ which is passed to the robot and also is used in the state transition equation (19) in the next step.

6. Results

In order to test the designed controller, the truck is backed to the loading dock from two different initial positions (Fig. 17). Hierarchical control system is very suitable for the implementation of the multi-level control principle and bringing it back together into one functional block. Experimental and simulation results using the present hierarchical scheme for different initial positions are shown in Fig. 17. In this figures, t indicates the parking

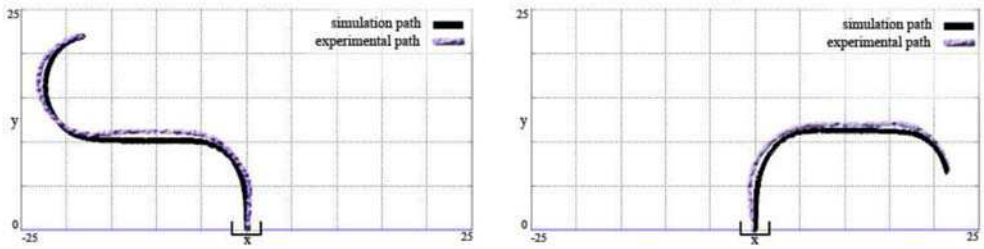
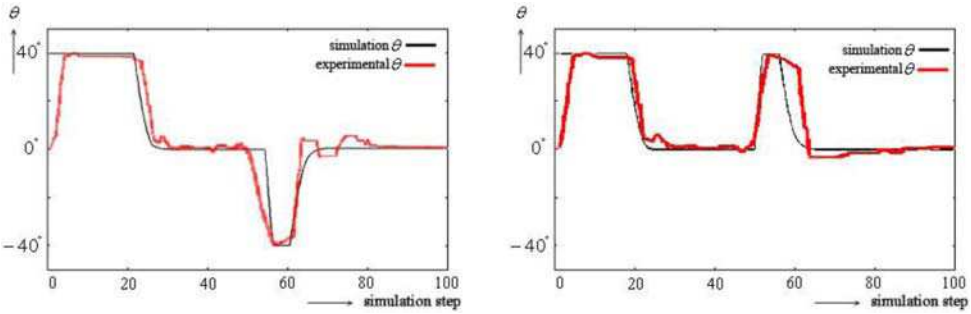


Fig. 17. Experimental and simulation results of the parking maneuver corresponding to the initial configurations (a) x=-20, y=18.4, φ =60, t=78 steps, (b) X=17.5, y=4, φ =162, t=69 steps

duration. It can be seen how the generated paths (Fig. 17) are very close to the ideal paths (Fig. 4) made up of circular arcs and straight lines.
Fig.18 illustrates how the steering angle "given by the hierarchical fuzzy controller" in short paths of Fig.17 is continuous, so the robot can move continuously without stopping.
The difference between generated paths (Fig. 17) is attributed to error of the vision subsystem, in estimating x,y,φ position variables. This error is propagated to the output of the controller and finally to the position of robot in the real environment.



(a) (b)

Fig. 18. (a) Experimental and simulation steering angle transitions for the paths in Fig. 17(a), (b) Experimental and simulation steering angle transitions for the paths in Fig. 17(b)

7. Conclusion

A fuzzy control system has been described to solve the truck backer-upper problem which is a typical problem in motion planning of nonholonomic systems. As hierarchy is an indispensable part of human reasoning, its reflection in the control structure can be expected to improve the performance of the overall control system. The main benefit from problem decomposition is that it allows dealing with problems serially rather than in parallel. This is especially important in fuzzy logic where large number of system variables leads to exponential explosion of rules (curse of dimensionality) that makes controller design extremely difficult or even impossible. The "divide and rule" principle implemented through hierarchical control system makes it possible to deal with complex problems without loss of functionality. It has also been shown that problem decomposition is vital for successful implementation of linguistic analysis and synthesis techniques in fuzzy modelling and controlling because a hierarchy of fuzzy logic controllers simulates an existing hierarchy in the human decision process and keeps the linguistic analysis less complicated so that it is manageable. In this work the proposed controller has a hierarchical structure composed of two modules which adjust the proper steering angle of front wheels similar to what a professional driver does. The computational cost is also less because we don't have to work with nonlinear function such as "Arccos (.)". Compared with traditional controller, this fuzzy controller demonstrates advantages on the control performance, robustness, smoothness, rapid design, convenience and feasibility. Trajectories are composed of circular arcs and straight segments and as a result the hierarchical approach produces shorter trajectories in comparison with other methods. The control system has been simulated with a model of a mobile robot containing kinematics constraints. 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 and short paths with continuous-curvature are generated during backward maneuver. The vision system utilizes measurements extracted from a ceiling mounted camera and estimates the mobile robot position using an extended Kalman filtering scheme. This results in correction and denoising of the measured position by exploiting the kinematic equations of the robot's motion.

8. References

Paromtichk, I.; Laugier, C.; Gusev, S. V. & Sekhavat, S. (1998). Motion control for parking an autonomous vehicle, *Proc. Int. Conf. Control Automation, Robotics and Vision*, vol. 1, pp. 136–140

Latombe, J. C. (1991). Robot motion planning, Norwell, MA: Kluwer Murray, R. M. & Sastry, S. S. (1993). Nonholonomic motion planning: Steering using sinusoids, *IEEE Trans. Automat. Contr.*, vol. 38, pp. 700–715

Lamiraux, F. & Laumond, J.-P. (2001). Smooth motion planning for car-like vehicles, *IEEE Trans. Robot. Automat.*, vol. 17, pp. 498–502

Scheuer, A. & Fraichard, T. (1996). Planning continuous-curvature paths for car-like robots, *Proc. IEEE Int. Conf. Intelligent Robots and Systems*, vol. 3, Osaka, Japan, pp. 1304–1311

Walsh, G.; Tylbury, D.; Sastry, S.; Murray, R. & Laumond, J. P. (1994). Stabilization of trajectories for systems with nonholonomic constraints, *IEEE Trans. Automat. Contr.*, vol. 39, pp. 216–222

Tayebi, A. & Rachid, A. (1996). A time-varying-based robust control for the parking problem of a wheeled mobile robot, *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 3099–3104

Jiang, K. & Seneviratne, L. D. (1999). A sensor guided autonomous parking System for nonholonomic mobile robots, *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 311–316

Gomez-Bravo, F.; Cuesta, F. & Ollero, A. (2001). Parallel and diagonal parking in nonholonomic autonomous vehicles, *Engineering Applications of Artificial Intelligence*, New York: Pergamon, vol. 14, pp. 419–434

Cuesta, F.; Bravo, F. G. & Ollero, A. (2004). Parking maneuvers of industrial-like electrical vehicles with and without trailer, *IEEE Trans. Ind. Electron.*, vol. 51, pp. 257–269

Reeds, J. A. & Shepp, R. A. (1990). Optimal path for a car that goes both forward and backward, *Pacific J. Math.*, vol. 145, no. 2, pp. 367–393.

Nguyen, D. & Widrow, B. (1989). The truck backer-upper: An example of self learning in neural network, *Proc. of the International Joint Conference on Neural Networks*, Washington DC, pp. 357-363

Kong, S. & Kosko, B. (1990). Comparison of fuzzy and neural truck backer-upper control systems, *Proc. IJCNN*, vol. 3, pp. 349-358

Koza, J.R. (1992). A genetic approach to the truck backer upper problem and the inter-twined spirals problem, *Proc. Int. Joint Conf. Neural Networks*, Piscataway, NJ, vol. 4, pp. 310-318

Schoenauer, M. & Ronald, E. (1994). Neuro-genetic truck backer-upper controller, *Proc. First Int. Conf. Evolutionary Comp.*, pages 720-723. Orlando, FL, USA

Jenkins, R.E. & Yuhas, B.P. (1993). A Simplified Neural Network Solution Through Problem Decomposition: The Case of the Truck Backer-Upper, *IEEE Trans. Neural Networks*, vol. 4, no. 4, pp. 718-720

Tanaka, K.; Kosaki, T. & Wang, H.O. (1998). Backing Control Problem of a Mobile Robot with Multiple Trailers: Fuzzy Modelling and LMI-Based Design, IEEE *Trans. Syst., Man, Cybern.*, Part C, vol. 28, no. 3, pp. 329-337

Ramamoorthy, P.A. & Huang, S. (1991). Fuzzy expert systems vs. neural networks – truck backer-upper control revisited, *Proc. IEEE Int. Conf. Systems Engineering*, pp. 221-224

Wang, L.-X. & Mendel, J.M. (1992). Generating fuzzy rules by learning from examples. IEEE *Trans. Systems, Man, and Cybernetics*, vol. 22, no. 6, pp. 1414-1427

Ismail, A. & Abu-Khousa, E.A.G. (1996). A Comparative Study of Fuzzy Logic and Neural Network Control of the Truck Backer-Upper System, *Proc. IEEE Int. Symp. Intelligent Control*, pp. 520-523

Kim, D. (1998). Improving the fuzzy system performance by fuzzy system ensemble, *Fuzzy Sets and Systems*, vol. 98, pp. 43-56

Dumitrache, I. & Buiu, C. (1999). Genetic learning of fuzzy controllers, *Mathematics and Computers in Simulation*, vol. 49, pp. 13-26

Chang, J.S.; Lin, J.H. & Chiueh, T.D. (1995). Neural networks for truck backer upper control system, *Proc. International IEEE/IAS Conference on Industrial Automation and Control*, Taipei, pp. 328–334.

Schoenauer, M. & Ronald, E. (1994). Neuro-genetic truck backer-upper controller, *Proc. of the First IEEE Conference on Evolutionary Computation*, Part 2(of 2), Orlando, pp. 720–723

Wang, L.X. & Mendel, M. (1992). Fuzzy basis function, universal approximation, and orthogonal least-squares learning, *IEEE Trans. Neural Networks*, 3 (5), 807–814

Li,Y. & Li, Y. (2007). Neural-fuzzy control of truck backer-upper system using a clustering method, *NeuroComputing*, 70, 680-688

Riid, A. & Rustern, E. (2001). Fuzzy logic in control: truck backer-upper problem revisited, *The 10th IEEE International Conference on Fuzzy Systems*, Melbourne, pp. 513–516.

Riid, A. & Rustern, E. (2002). Fuzzy hierarchical control of truck and trailer, *The 8th Biennal Baltic Electronic Conference,* dcc,ttu,ee, Tallinn, pp. 343–375

Li, T. - H. S. & Chang, S.-J. (2003). Autonomous fuzzy parking control of a car-like mobile robot, *IEEE Trans. Syst., Man,Cybern.,* A, vol.3, pp. 451–465

Chen G. & Zhang, D. (1997). Back-driving a truck with suboptimal distance trajectories: A fuzzy logic control approach, *IEEE Trans. Fuzzy Syst.*, vol. 5, pp. 369–380

Shahmaleki, P. & Mahzoon, M. (2008). Designing a Hierarchical Fuzzy Controller for Backing-up a Four Wheel Autonomous Robot, *American Control Conference*, Seattle, 10.1109/ACC.2008.4587269

Shahmaleki, P.; Mahzoon, M. & Ranjbar, B. (2008). Real time experimental study of truck backer upper problem with fuzzy controller, *World Automation Congress, WAC 2008*, Page(s):1 – 7

Sugeno, M. & Murakami, K. (1985). An experimental study on fuzzy parking control using a model car, *Industrial Applications of Fuzzy Control*, M. Sugeno, Ed. North-Holland, The Netherlands, pp. 105–124

Sugeno, M.; Murofushi, T.; Mori, T.; Tatematsu, T. & Tanaka, J. (1989). Fuzzy algorithmic control of a model car by oral instructions, *Fuzzy Sets Syst.*, vol. 32, pp. 207–219

Yasunobu, S. & Murai, Y. (1994). Parking control based on predictive fuzzy control, *Proc. IEEE Int. Conf. Fuzzy Systems*, vol. 2, pp. 1338–1341

Daxwanger, W. A. & Schmidt, G. K. (1995). Skill-based visual parking control using neural and fuzzy networks, Proc. IEEE Int. Conf. System, Man, Cybernetics, vol. 2, pp. 1659–1664

Tayebi, A. & Rachid, A. (1996). A time-varying-based robust control for the parking problem of a wheeled mobile robot, *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 3099–3104

Leu, M. C. & Kim, T. Q. (1998). Cell mapping based fuzzy control of car parking, *Proc. IEEE Int. Conf. Robotics Automation*, pp. 2494–2499

An, H.; Yoshino, T.; Kashimoto, D.; Okubo, M.; Sakai, Y. & Hamamoto, T. (1999). Improvement of convergence to goal for wheeled mobile robot using parking motion, *Proc. IEEE Int. Conf. Intelligent Robots Systems*, pp. 1693–1698

Shirazi, B. & Yih, S. (1989). Learning to control: a heterogeneous approach, *Proc. IEEE Int. Symp. Intelligent Control*, pp. 320–325 O

hkita, M.; Mitita, H.; Miura, M. & Kuono, H. (1993). Traveling experiment of an autonomous mobile robot for a flush parking, *Proc. 2nd IEEE Conf. Fuzzy System*, vol. 2, Francisco, CA, pp. 327–332

Laumond, J. P.; Jacobs, P. E.; Taix, M. & Murray, R. M. (1994). A motion planner for nonholonomic mobile robots, *IEEE Trans. Robot. Automat.*, vol. 10, pp. 577–593

Demilri, K. & Turksen, I.B. (2000). Sonar based mobile robot localization by using fuzzy triangulation, *Robotics and Autonomous Systems*, vol. 33, pp. 109-123

Miah, S. & Gueaieb, W. (2007). Intelligent Parallel Parking of a Car-like Mobile Robot Using RFID Technology, *Robotics and Sensor Environments*, IEEE International Workshop On, pp. 1-6

Chen, Ch.Y. & Feng, H.M. (2009). Hybrid intelligent vision-based car-like vehicle backing systems design, *Expert Systems with Applications*, vol. 36, issue 4, pp. 7500-7509

Hough, P. (1962). Methods and means for recognizing complex patterns, U.S. Patent 3069654

Duda, R. & Hart, P. (1972). Use of the Hough Transformation to Detect Lines and Curves in Pictures, *Comm. ACM*, vol. 15, pp. 11–15

Duda, R.; Hart, P. & Stork, D. (2000). *Pattern Classification* (2nd ed.), Wiley Interscience

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*, Springer-Verlag, New York

Kalman, R. (1960). A new approach to linear filtering and prediction problems, *Transactions of ASME – Journal of Basic Engineering*, 82, pp. 32-45

Bar-Shalom, Y. & Fortmann, T. E. (1988). *Tracking and data association*, San Diego, California: Academic Press, Inc.

Motion Control

Edited by Federico Casolo

ISBN 978-953-7619-55-8
Hard cover, 590 pages
Publisher InTech
Published online 01, January, 2010
Published in print edition January, 2010

The book reveals many different aspects of motion control and a wide multiplicity of approaches to the problem as well. Despite the number of examples, however, this volume is not meant to be exhaustive: it intends to offer some original insights for all researchers who will hopefully make their experience available for a forthcoming publication on the subject.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Pourya Shahmaleki, Mojtaba Mahzoon, Alireza Kazemi and Mohammad Basiri (2010). Vision-Based Hierarchical Fuzzy Controller and Real Time Results for a Wheeled Autonomous Robot, Motion Control, Federico Casolo (Ed.), ISBN: 978-953-7619-55-8, InTech, Available from: http://www.intechopen.com/books/motion-control/vision-based-hierarchical-fuzzy-controller-and-real-time-results-for-a-wheeled-autonomous-robot

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821