Automatic Parallel Parking Algorithm for a Carlike Robot using Fuzzy PD+I Control

Article *in* Engineering Letters · November 2018

CITATION
1

READS
989

5 authors, including:

Yoshio Josué Rubio
Instituto Politécnico Nacional
10 PUBLICATIONS **10** CITATIONS
SEE PROFILE

Enrique Ballinas
Instituto Politécnico Nacional
2 PUBLICATIONS **3** CITATIONS
SEE PROFILE

Oscar Humberto Montiel Ross
Instituto Politécnico Nacional
157 PUBLICATIONS **1,756** CITATIONS
SEE PROFILE

Oscar Castillo
Tijuana Institute of Technology
1,007 PUBLICATIONS **13,398** CITATIONS
SEE PROFILE

Automatic Parallel Parking Algorithm for a Car-like Robot using Fuzzy PD+I Control

Enrique Ballinas, Oscar Montiel, Oscar Castillo, Yoshio Rubio, Luis T. Aguilar

***Abstract*—In this work, the design, analysis, and implementation of an algorithm for automatic parallel parking for a nonholonomic mobile robot is presented. The mobile robot is a four-wheeled scaled vehicle and it is assumed that there is space limitation for the parking maneuver. The main objective was to design a parallel parking path trajectory avoiding collisions. We designed a fuzzy PD+I controller for driving the error generated between the real position and the previously generated objective position to the origin. We presented simulations results to validate the analysis and demonstrating how the fuzzy controller solved the tracking problem for the derived path trajectory to follow.**

***Index Terms*—Automatic parking, fuzzy-PID control, mobile robots, nonholonomic system**

I. Introduction

In recent years, many car manufacturers have implemented an intelligent driver assistance system to its vehicles. These systems consist of an array of sensors to get information of the environment supporting the driver in different tasks. Some of these applications are lane keeping, blind spot detection, proximity indicators, driver drowsiness detectors, and assisted parallel parking systems, among others [1-4].

With the increase of vehicles in big cities and urban development, the availability of parking spaces is becoming a day to day issue, and parallel parking in confined areas can be a difficult task [5]. When trying to perform parking maneuvers, drivers can cause accidental damage to their cars such as scratches or slight dents to the vehicle.

In the last decade, much interest has been put by researchers and car companies to develop assisted or automatic parking systems. The system aims to improve the security and comfort of inexperienced and disable drivers, in this challenging operation [6,7].

Assisted parking systems facilitate the parking operation for the driver, the level of assistance varies, and commonly the driver has some control of the procedure. Some companies have introduced parallel parking assistance in their vehicles. For example, Toyota Motor Company launched the Toyota Prius in the early 2000's while Lexus, with their model LS, also includes an option of parking assistance, and Renault is researching Autonomous Valet Parking [7,8].

Motivation: The fact is that parallel parking of autonomous cars is a challenging task that has allowed neither the complete welcome nor the confidence of such vehicles in the worldwide market. Most advanced systems are automatic parallel parking systems, where there is no need for driver input. Such algorithm consists in, once a free space is validated, the parking routine is performed by controlling the steering angle of the vehicle and following a collision-free trajectory from the origin point to a desired point in the parking spot.

There exist two main approaches reported in the literature for automatic parking. The first one is based on the stabilization of the vehicle to a target point, where the vehicle travels without a planned path and the objective is to reduce the distance between the reference point of the vehicle and the goal. The second approach consists of performing a path planning that takes the vehicle from an initial point to the target point and then follows the path [9].

Existing path planning strategies for automatic parking in vehicles have three main steps. First, the vehicle sensors check for available space for parking. After that, a collision-free path is calculated. In the last step, a control strategy to follow the calculated path and to evade any unexpected obstacle is used. The existence of unexpected obstacles can lead to a path recalculation [1,9,10]. There have been reported different approaches for planning the collision-free path used by the parking maneuver, being the geometric approach the most common one [9-11]. This methodology is based on human driver's heuristics when parking: first the drivers turn all the steering wheel to maximum angle in the same direction as the parking space; at a middle point, the driver steers the maximum angle in the opposite direction till the vehicle is parallel to the parking spot. Hence, the whole a path consists of two identical curvatures that connect in the middle point.

The geometrical approach works when the length of the parking space is big enough. Under constrained parking spaces, modifications to the trajectory is needed. The identical curvatures remain the same, but in a third step, the steering wheel moves slightly to the opposite side and then moves forward until the vehicle is in the middle of the parallel spot or at a safe distance of the car front of us.

Manuscript received May 6, 2018; revised August 26, 2018.

E. Ballinas is with Centro de Investigación y Desarrollo de Tecnología Digital (CITEDI) del Instituto Politécnico Nacional (IPN), Ave. Instituto Politécnico Nacional No. 1310 Colonia Nueva Tijuana, Tijuana Baja California, México (e-mail: lballinas@ citedi.mx).

O. Montiel is with Centro de Investigación y Desarrollo de Tecnología Digital (CITEDI) del Instituto Politécnico Nacional (IPN), Ave. Instituto Politécnico Nacional No. 1310 Colonia Nueva Tijuana, Tijuana Baja California, México. Corresponding author phone: +52(664)6231344; fax: +52(664)6231344; (e-mail: oross@ipn.mx).

O. Castillo is with Tijuana Institute of Technology, Calzada Tecnológico s/n, Fracc. Tomas Aquino, CP 22379, Tijuana, México (email: ocastillo@tectijuana.mx).

Y. Rubio is with Centro de Investigación y Desarrollo de Tecnología Digital (CITEDI) del Instituto Politécnico Nacional (IPN), Ave. Instituto Politécnico Nacional No. 1310 Colonia Nueva Tijuana, Tijuana Baja California, México (e-mail: rrubio@citedi.mx).

L. Aguilar is with Centro de Investigación y Desarrollo de Tecnología Digital (CITEDI) del Instituto Politécnico Nacional (IPN), Ave. Instituto Politécnico Nacional No. 1310 Colonia Nueva Tijuana, Tijuana Baja California, México (email: laguilarb@ipn.mx)

Fuzzy controllers have been widely used for automatic parallel parking. Li and Tseng [10] implemented automatic parallel parking for an autonomous vehicle using a Fuzzy-PID. The system has the following stages: place searching, steering control, path tracking, and wireless communication. Ultrasonic sensors obtain the data from the surroundings. Then the information about the suitable parking spot is recorded and sent to the driver's smartphone. Finally, the parking routine using the Fuzzy-PID is executed.

Another work using a fuzzy controller was presented by Aye [12], who constructed the fuzzy rules based on the skills of an experienced human driver. The path was followed using camera information to obtain the position of the vehicle, and the fuzzy controller corrects the position based on the error of the actual position with respect to the desired position. Their membership functions were tuned using a genetic algorithm to minimize the position error.

Filatov *et al*. [6] used a Fuzzy controller to adjust the speed of a car-like robot when parking. The vehicle follows a path generated by a geometric approach, with the position of the robot tracked by a camera, the speed is estimated using distance sensors as the input for the fuzzy controller. In a similar approach, Lee et al. [13] used a Fuzzy controller to adjust the acceleration and the speed of an autonomous vehicle in the parking operation, but they used a LiDAR distance sensor to get more information of the surroundings. Scicluna [14] also used a LiDAR and Fuzzy Logic implemented into an FPGA to enhance the processing speed.

Fuzzy logic controllers have been successfully applied to different tasks and architectures of mobile robots. For example in Castillo *et al*. [15] a fuzzy logic controller was developed for a nonholonomic unicycle mobile robot based on a backstepping approach to ensure the stabilization of the robot's position and orientation around a desired path. For the same kind of mobile robots, Astudillo *et al*. [16] used a type-2 fuzzy logic controller to control the torque of an autonomous mobile robot when following the desired path.

Contribution of the paper: This work uses the path-following approach for parallel parking controlled by a Fuzzy PD+I controller, with the fuzzy rules and membership functions based on the experience of human drivers. We are using the geometrical approach to calculate the desired path, which is based on the three-step movement heuristics for parallel parking. The contribution of this work is the design of a Fuzzy PD+I controller for automatic parallel parking and the mathematical formulation to find the parking route. The controller is tested experimentally in a laboratory scaled car-like robot.

This paper is organized as follows: In Section II, we introduce the kinematics of car-like robots, the theoretical framework of fuzzy controllers for path following, the geometrical approach to obtain the path that the robotic car must follows, the design specifications of the Fuzzy PD, and the prototype used for the test. In Section III, the results of the Fuzzy PD+I controller and the comparison using a classical PID control are presented and discussed. Finally, in Section IV we present the conclusions.

II. Materials and Methods

A. Kinematics of a car-like robot

A car-like robot, from a kinematic point of view, is a constrained driven system since it has less controllable degrees of freedom than the total amount of degrees in the system. Therefore, these robots are called underactuated [17] and can be explained by analyzing how the constraints of the system are expressed. A system is called *holonomic*, if its constraints are expressed without any time derivative, using the form $\boldsymbol{f}(\boldsymbol{q},\boldsymbol{t}) = \boldsymbol{0}$, where $\boldsymbol{q}$ are the constraints of the system. On the other hand, if the constraints of the system have derivatives in the form $f(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}},t) = 0$, the system is called *nonholonomic*. Since the equations of nonholonomic systems are non-integrable, the total amount of degrees-of-freedom in the system is greater than the degree of freedom that can be controlled [10,17].

The kinematics constraints of car-like robots arise from the wheels that control the steering of the robot. Wheeled robots operate in a two-dimensional plane, which gives them three degrees of freedom: *x* and *y*-axis, and orientation around one of the axes (typically the *x*-axis). The relation between the position of the car-like robot in the *x* and *y*-axis depends on the orientation given by the wheels for steering; this causes kinematics constraints with derivative relations, which classifies this type of robots as a nonholonomic system.

As mentioned above, the position of car-like robots is described by a point in a two-dimensional plane where the reference point is in the middle of the rear axle. The orientation angle concerning the two-dimensional plane (x,y) is denoted by θ. At any given time and point *P*, the system has an instantaneous linear velocity v and an angular velocity ω for the wheels when steering (see Fig. 1).

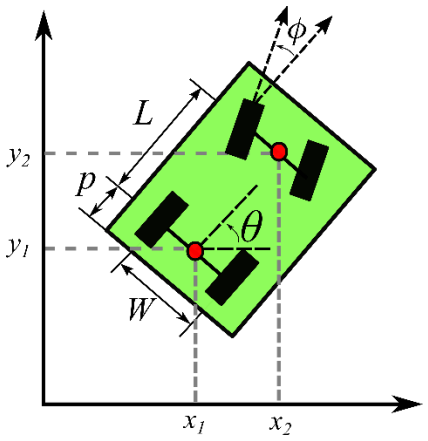

Fig. 1. Kinematic scheme of a front-steered car-like robot. *L* denotes the wheelbase of the model, *p* is the distance between the rear axle and the rear bumper and *W* the axle width.

As mentioned before, the nonholonomic constraints are imposed by the steering angle represented by ϕ, which constrains the position and orientation of the robot [18-20]. The restrictions of a front-steered robot are described as follows:

$$\begin{aligned} \dot{y}_1 \cos\theta - \dot{x}_1 \sin\theta = 0\,, \\ \dot{y}_2 \cos(\theta+\phi) - \dot{x}_2 \sin(\theta+\phi) = 0, \end{aligned} \tag{1}$$

where $x_2 = x_1 + L\cos\theta$ and $y_2 = y_1 + L\sin\theta$. From (1), the constrained variables are

$$\boldsymbol{q} = \begin{bmatrix} x_1 \\ y_1 \\ \theta \\ \phi \end{bmatrix}. \tag{2}$$

To neglect the influence of slippage, i.e., the wheels roll without slip and also cannot have side slip [21], the kinematic model considers a robot with low mass and low velocities [22]. After this consideration, the instantaneous curvature k of the robot is only affected by the steering angle of the front wheels ϕ, and the length L between axles. The instantaneous curvature is calculated as follows:

$$k = \frac{\tan\phi}{L}. \tag{3}$$

Once the instantaneous curvature is defined, the next step is to determine the relationship between the elements of $\boldsymbol{q}$ to obtain the kinematics equations of the system. Since the rate of change for the steering angle $\dot{\phi}$ only depends on the angular velocity of the front wheels, the rate of change of the steering angle is represented by the first order differential equation $\dot{\phi} = \omega$ [1,22,23]. Using the system described in Fig. 1, and the provided instantaneous curvature and linear velocity equations (1)-(3), the following kinematic model of the car-like robot is obtained:

$$\dot{q} = \begin{bmatrix} \dot{x}_1 \\ \dot{y}_1 \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ \frac{\tan\phi}{L} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}. \tag{4}$$

We assume that the linear and angular velocities and the steering angle are bounded, that is, there are apriori known positive constants v_{max}, ϕ_{max}, w_{max}, such that

$$\begin{cases} |v| \le v_{max}, \\ |\phi| \le \phi_{max}, \\ |\omega| \le \omega_{\max}. \end{cases} \tag{5}$$

B. Fuzzy controllers

Fuzzy logic can emulate the approximate reasoning of the human brain based on natural language with uncertainty, which allows the handling of vague information and an imprecise set or rules. The main advantage of these systems is that they can use human knowledge in problems where it is difficult or sometimes impossible to build accurate mathematical models.

A fuzzy set is represented by:

$$A = \{(x, \mu_A(x) | x \in X)\}, \tag{6}$$

where $\mu_A(x)$ is called the membership function of the fuzzy set A. The membership function maps each element of $\boldsymbol{X}$ to a membership grade between 0 and 1. Set $\boldsymbol{X}$ is referred as the universe of discourse, and it can be a discrete or continuous space [24].

A Fuzzy inference system (FIS) can implement a non-linear mapping of its input to an output space. This mapping is achieved through the IF—THEN fuzzy rules, where each rule describes the local behavior of the mapping. A fuzzy IF—THEN rule is a sentence that is divided into two parts and has the next form: $if\ x\ is\ A\ then\ y\ is\ B$. The first part is the antecedent or premise (*x is A*), while the second part is the consequence or conclusion (*y is B*).

The Inference system (see Fig. 2) has a knowledge base that includes the information given by the expert in the form of linguistic control rules. The fuzzification interface transforms the crisp data (inputs) into fuzzy sets based on the membership functions (MF) and gives this data to the knowledge base. The last stage of the FIS is the defuzzification interface and translates the fuzzy control actions into a real control action [24].

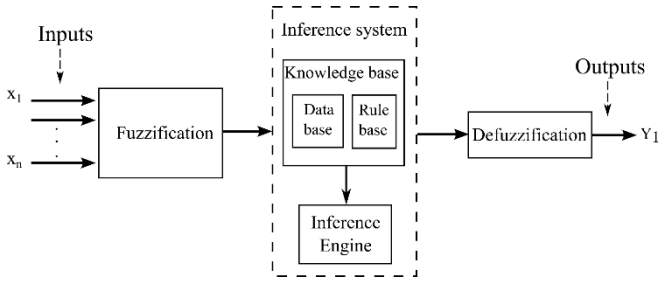

Fig. 2. Fuzzy Inference System.

FIS has been widely used as control systems [24, 25] in many applications, and because their properties, FISs are candidates to be mixed with classical control techniques such as the proportional-integral-derivative (PID) controller. The PID controller, in the discrete time domain, is given by [26]:

$$u(k) = \underbrace{K_P e(k)}_{P} + \underbrace{K_I \sum_{i=0}^{k} e(i)}_{I} + \underbrace{K_D \Delta e(k)}_{D} \tag{7}$$

where K_p, K_i, K_D are the system's gains and e is the error generated by the difference between the current value and the desired value. The main objective of the controller is to minimize the control error by adjusting the process controller output; then the system should reach a point of stability. Stability means the set point is held on the output without oscillations around it [27]. This controller offers fast response proportional to the error, while it has an automatic reset from the integral part that eliminates the steady-state error. The derivative action allows the controller to respond fast to changes in the error [7, 28].

The fuzzy PD+I controller is in the discrete time domain (see Fig. 3). It mixes a fuzzy PD controller with two inputs — the error and the derivative of error (e,Δe) multiplied by their respective constants (K_P, K_D)— with the integral of the error. The output of the FD is added to the integral of the error multiplied by K_I, as shown by the I term of (7). The final output u is then multiplied by K_U to obtain U that is the input of the process to be controlled. The error given by the measurement of the desired state and the actual state is then the new error for the Fuzzy PD+I controller.

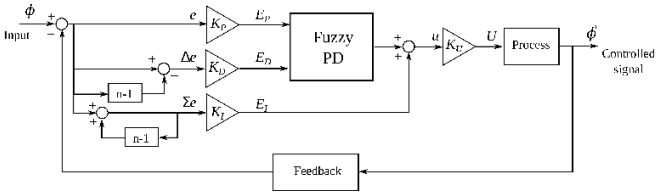

Fig. 3. Fuzzy PD+I controller in discrete domain time.

Opposite to the PID controller based on difference equations, in the PD+1 controller, the PD part is achieved by a FIS without including the integral controller to reduce the fuzzy controller complexity. This kind of controllers has been applied successfully to many control problems [25].

C. Path planning for parallel parking

For our implementation, the path planning is based on the geometric approach of Choi [9]. To assure that the moving robot will not impact the already parked front vehicle, a perfectly aligned robot at the minimum distance x_{min} and y_{min} in both axes is considered. The design of the full path consists of three segments: two identical curves that connect at a breaking point and a small straight segment to keep the vehicle in the middle of the parking spot (see Fig. 4). The steering angle only changes its direction only at the junction of pair segments, and depends on the arc segment α [7].

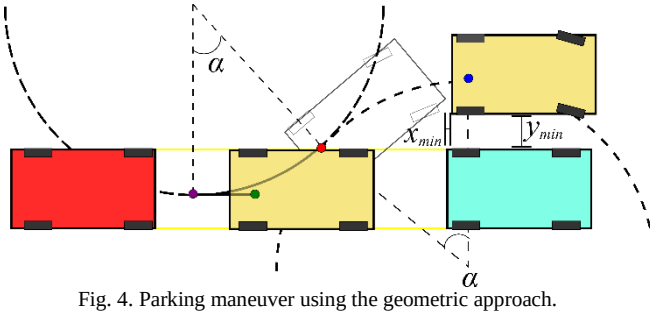

Fig. 4. Parking maneuver using the geometric approach.

The desired path is integrated by four points (*A*, *B*, *C*, *D*), that are the components of the three segments: curves $\overline{AB}$ and $\overline{BC}$, and a straight-line $\overline{CD}$ (see Fig. 5). Both curves, $\overline{AB}$ and $\overline{BC}$, have identical length S and angle α. Therefore, the *path planning stage aim*s to calculate $\bar{P}$ which is the full path that the controlled car must follow to get parked correctly. The full path is given by (8).

$$\bar{P} = \overline{AB} + \overline{BC} + \overline{CD} \tag{8}$$

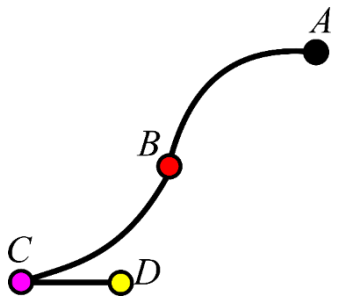

Fig. 5. The segments $\overline{AB}$, $\overline{BC}$, and $\overline{CD}$ conform the full path for parallel parking of the autonomous car. The first two segments are driven in reverse whereas the last segment is driven forward.

A geometrical analysis is needed to get the relations between the dimensions and the kinematic model of the car-like robot with respect to the desired paths. To obtain these relations, a steady steering angle should be set. If the steering angle is different of zero, then a circle of radius R is formed, getting bigger at lower angles and smaller at bigger steering angles.

The minimum arcs for the curves $\overline{AB}$ and $\overline{BC}$ (see Fig. 5) depend strongly on the dimensions of the robot. If the distance of the arc radius is too small, the inner part of the car being parking can hit with the front parked car. On the other hand, if the radius is too big, then the vehicle will hit the back car.

Two circles can be obtained from arcs $\overline{AB}$ and $\overline{BC}$, with center C_2 and C_1, respectively (see Fig. 6), which are tangent to the point (X_T, Y_T). The origin is the center C_2 with respect to our coordinate system that is aligned with the initial point of the system, where $X_S = 0$ and $Y_S = R$. The center C_2 is the vertex of the first arc of the path $\bar{P}$, whereas C_1 is the vertex of the second arc of the path $\bar{P}$.The distance from the intersection point to both centers is denoted by R, which is the same distance of the point (X_S, Y_S) to C_2, and the distance from the point (X_F, Y_F) to C_1. The distance R is defined as:

$$R = \frac{L}{\tan\phi} \tag{9}$$

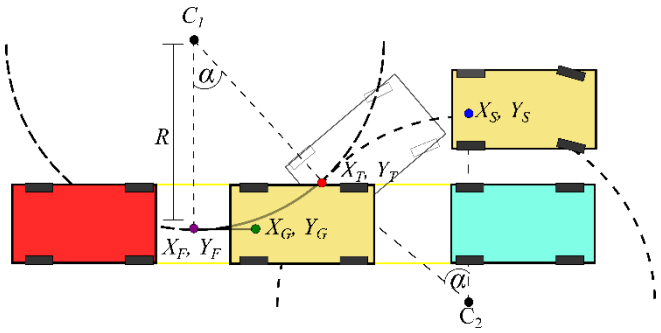

Fig. 6. Parking path components.

Since both arcs are identical, and they share a common point, it is possible to calculate their aperture angle α of the arc if the coordinates of the common point are known. If we use as a restriction that point (X_T, Y_T) must be at the same height that the parking spot (see Fig. 6), the reference point of the system is C_2, and using:

$$X_T = R\cos\alpha, \tag{10}$$
$$Y_T = R\sin\alpha, \tag{11}$$

the height of the intersection can be calculated as follows $Y_T = R - \left(y_{min} + \frac{W}{2}\right)$. Because the height of the point is known, the angle α can be found using the following relation:

$$\alpha = \left|\sin^{-1}\left(\frac{R-\left(y_{min}+\frac{W}{2}\right)}{R}\right)\right|. \tag{12}$$

The last two values to calculate are the minimum length of the parking space M_{min} and the location of (X_G, Y_G). The length of the parking space must be long enough to allow the vehicle to perform the parking operation without impacting the rear vehicle. Using Fig. 4 and Fig. 6, the length M_{min} can be calculated as follows, where p is the distance between the rear axle and the rear bumper as is illustrated in Fig. 1,

$$M_{min} = 2X_T + p - x_{min}\,. \tag{13}$$

The height of the goal point Y_G is the same Y_F, or the position in the circle with center at C_1 at 3π/2 radians. An easier way to calculate this position is with the geometric information that is already known (see Fig. 4 and Fig. 6), and the algebraic result is:

$$Y_G = R - (W + y_{min}). \quad (14)$$

Using the same idea, and the minimum length for the parking space, the X_G is obtained from the following relation:

$$X_G = \frac{M_{min}}{2} + x_{min} + \frac{L}{2} + p. \quad (15)$$

Algorithm 1 shows step-by-step the calculus of the desired path using the geometrical approach.

Algorithm 1. Steps to obtain the full path $\bar{P}$

Input: Steering angle ϕ, wheelbase L, the width of the robot W, the distance between the rear axle and the rear bumper p, the initial points X_S, Y_S and the minimum distance x_{min} and y_{min}

Output: The whole path $\bar{P}$

1 Calculate the distance R

$$R = \frac{L}{\tan(\phi)}$$

2 Find the intersection point Y_T using

$$Y_T = R - \left(y_{min} + \frac{W}{2}\right)$$

3 Calculate the aperture angle α using (10)

4 Find the intersection point X_T using (8)

5 Calculate the parking length space M_{min} using (11)

6 Find the goal point X_G using (13)

7 Calculate the goal point Y_G using (12)

8 $Y_F = Y_G$

Calculate the X_F point using

$$X_F = M_{min} - p$$

9 Calculate the full path $\bar{P}$

$$\bar{P} = \overline{(X_S, Y_S)(X_T, Y_T)} + \overline{(X_T, Y_T)(X_F, Y_F)} + \overline{(X_F, Y_F)(X_G, Y_G)}$$

10 **end**

D. Control strategy

The control strategy to solve the path tracking problem is a Fuzzy PD+I. The aim of the Fuzzy PD+I system is to reduce the difference between the desired path $\bar{P}$ and the actual position modifying the steering angle of the robot (see Fig. 7). The sensor to measure the position of the vehicle is an odometer, which calculates the position based on the orientation and velocity of the vehicle.

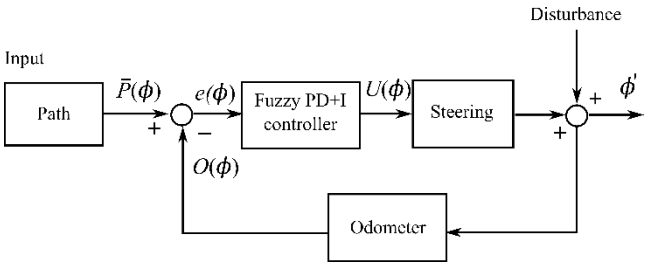

Fig. 7. Proposed Fuzzy PD+I controller system for path following.

The path created with the designated steering angle is discretized, and it gives the desired position $\bar{P}(\phi)$ at a given steering angle ϕ. The odometer measures the actual position $O(\phi)$, and then, the error between both is calculated $e(\phi) = p(\phi) - O(\phi)$. The objective control of the system considering the time can be expressed as $\lim_{t\to\infty} \|e(\phi, t)\| = 0$.

The error $e(\phi)$ is multiplied by K_P and the derivative of the error by K_D, the results are the inputs for the FIS. The input for the Fuzzy PD+I controller are E_p, E_D, and E_I; and the output is the correction of the steering angle $U(\phi)$ (see Fig. 3). Fig. 8 gives more detail of the Fuzzy PD controller.

Since the output of the FIS is the raw correction for the steering angle $u(\phi)$, then the integral of the error $\Sigma e(\phi)$ is added and multiplied by the variable scaling factor K_U, that is,

$$U(\phi) = K_U\big(u(\phi) + K_I \Sigma e(\phi)\big). \quad (16)$$

The new value of the steering angle would be

$$\phi' = \phi + U(\phi), \quad (17)$$

that includes the correction of the current direction considering any unwanted displacement of the desired trajectory.

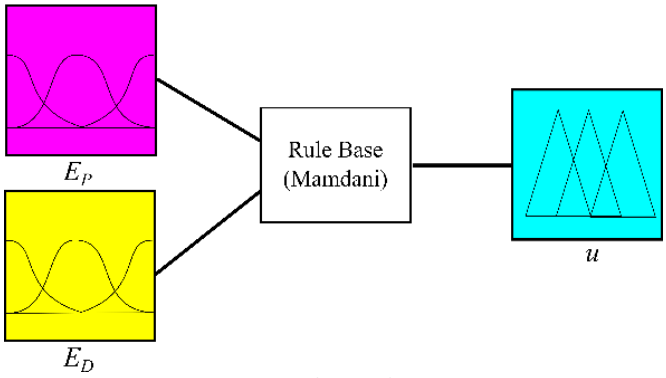

Fig. 8. Fuzzy PD for path following.

E. Fuzzy Inference System design

The membership functions (MFs) of the FIS are shown in Figs. 9, 10, and 11. Table I shows the type and parameter values of the MFs. Table II shows the rule matrix.

As mentioned before, there are two input variables: the first one is the error (e) multiplied by K_P and the second is the derivative of the error by K_D (E_p, E_D respectively). For each input variable, there are four linguistic terms: NB, N, Z, P, and PB. The input variables are measured in centimeters.

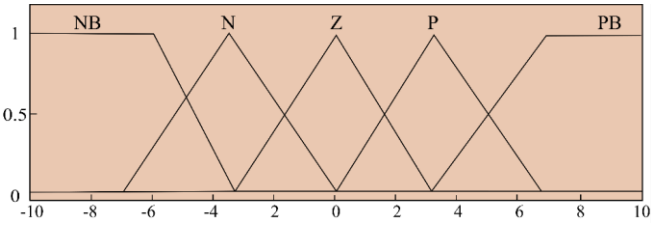

Fig. 9. Input membership functions for the error E_p.

For the output variable u, there are four membership functions: BI, I, Z, D, and BD (see Fig. 11). The aim of these membership functions is to correct the position of the robot based on the actual position of the reference point and the steering wheels. The universe of discourse of the output is in

radians, the maximum value was based on the physical limitations of the model.

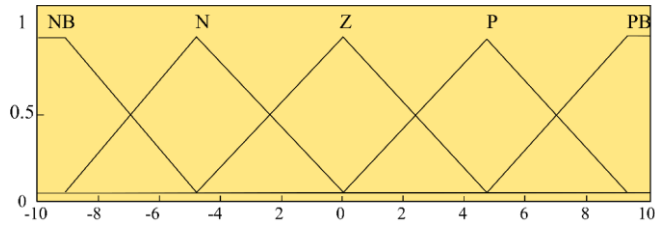

Fig. 10. Input membership functions for the derivative error E_D.

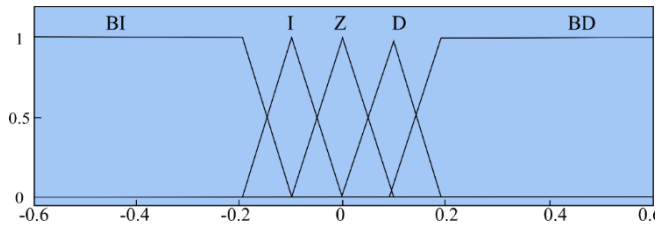

Fig. 11. Output membership functions of the variable u.

TABLE I
LINGUISTIC INPUT VARIABLES AND TERMS FOR THE FIS

Variable name	Term name	Type	Parameters
E_P	NB	Trapezoidal	[-20, -20, -6.7, -3.35]
E_P	N	Triangular	[-6.7, -3.35,0]
E_P	Z	Triangular	[-3.35, 0, 3.35]
E_P	P	Triangular	[0, 3.35, 6.7]
E_P	PB	Trapezoidal	[3.35, 6.7, 20, 20]
E_D	NB	Trapezoidal	[-20, -20, -8.75, -5]
E_D	N	Triangular	[-8.75, -5, 0]
E_D	Z	Triangular	[-5, 0, 5]
E_D	P	Triangular	[0, 5, 8.75]
E_D	PB	Trapezoidal	[5, 8.75, 20, 20]
u	BI	Trapezoidal	[-1, -1, -0.1833, -0.06109]
u	I	Triangular	[-0.1833, -0.09163, 0]
u	Z	Triangular	[-0.06109, 0, 0.06109]
u	D	Triangular	[0, 0.09163, 0.1833]
u	BD	Trapezoidal	[0.06109, 0.1833, 1,1]

The fuzzy rules summarized in Table II were created to ensure the correct amount of correction is computed depending on the behavior of the error through the parking operation.

TABLE II
FUZZY RULE MATRIX OF THE FIS

	NB	**N**	**Z**	**P**	**PB**
NB	BI	BI	BI	I	Z
N	BI	I	I	Z	D
Z	BI	I	Z	D	BD
P	I	Z	D	D	BD
PB	Z	D	BD	BD	BD

F. Materials

The mobile robot used for this paper is a 1:10 scale vehicle (see Fig. 12). The dimensions and physical limitations are shown in Table III. The main computer is an Odroid board (XU4 64 GB) controlled using ROS (*Robotic Operating System*). The vehicle has a 15-volt motor for traction and a servomotor for steering control. The vehicle has a laser-type sensor (RPLidar 360), which detects the available parking spaces and possible obstacles when the parking process is performed. An Intel SR300 camera module is part of the system that allows the detection of obstacles on the front. This camera module has three cameras: an RGB, an infrared, and a depth camera.

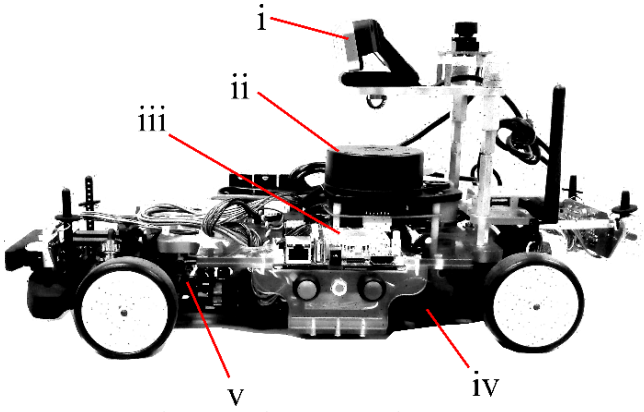

Fig. 12. Some of the main of components of the prototype: (i) Frontal camera, (ii) Lidar, (iii) CPU, (iv) traction, (v) and steering.

TABLE III
CAR-LIKE ROBOT DIMENSIONS

L[m]	W[m]	LF[m]	p[m]	ϕ_{max}[°]	ϕ_{min}[°]	v_{max}[m/s]
0.26	0.20	0.43	0.085	38	-38	0.58

III. Results and Discussion

Through experimentation x_{min} and y_{min} are estimated, obtaining $x_{min} = 0.02$ m and $y_{min} = 0.1$ m. Based on the dimensions of the robot and its physical limitations (see Table III), a steering angle of 0.5235 rad (30 degrees) was chosen. For the simulations, the starting angle was 0.6108 rad (35 degrees). The values for K_P, K_D, K_I, and K_U were found experimentally for both arcs. For the first arc we obtained $K_P = 0.3$, $K_D = 1$, $K_I = 0.01$, and $K_U = 0.016$. For the second arc, the values were $K_P = 0.3$, $K_D = 1$, $K_I = 0.001$, and $K_U = 0.001$.

To compare the proposed controller, a classical PID controller manually tuned was implemented. The values for the constants for the first arc are $K_P = 1.4$, $K_D = 1.4$, $K_I = 0.001$, and $K_U = 0.012$; and for the second arc are $K_P = 1.16$, $K_D = 1.4$, $K_I = 0.001$, and $K_U = 0.001$.

The results of our proposed controller and the PID controller can be seen in Table IV and Table V, the results show that both controllers lasted the same time to perform the parking task, they took 4.646 seconds for the first arc, and 4.364 for the second arc. However, Table V shows that the parking task with less error in both arcs was achieved by the Fuzzy PD+I controller.

The performance indices, provided in Table V, were obtained using the ℓ_2-norm defined as $\|e\|_2 = \sqrt{\sum_{i=1}^{n}|e_i|^2}$ to calculate the mean squared error (MSE) of the steady-state error of the first and second curvatures, whereas we used the $\|\ell\|_\infty$ norm defined as $\|e\|_\infty = \max_{1\le i\le t}|e_i|$ to report the maximum error shown in the fourth column.

TABLE IV
COMPARISON OF RESPONSE TIME BETWEEN FUZZY PD+I CONTROL AND PID CONTROL

Control	Time to reach stable state first curvature (s)	Time to reach stable state second curvature (s)
PID	4.626	4.364
Fuzzy PD+I	4.626	4.364

TABLE V
ERROR COMPARISON BETWEEN FUZZY PD+I CONTROL AND PID CONTROL

Control	Steady-state error for first curvature (cm)	Steady-state error for second curvature (cm)	Maximum error (cm)	Mean square error (cm)
PID	0.0195	0.0112	0.4565	0.0420
Fuzzy PD+I	0.0232	0.0068	0.4224	0.0266

The behavior of the fuzzy PD+I controller and the PID controller versus the desired parking trajectory is shown in Fig 13 and Fig 14, respectively. The upper part of both figures represents the first curve while the lower part is the second curve. In these figures, we show only the path of the two arcs since the last segment is just driving forward to park the car in the center of the parking space. Both controllers decreased the path error satisfactorily preventing the vehicle from crashing during the parking process.

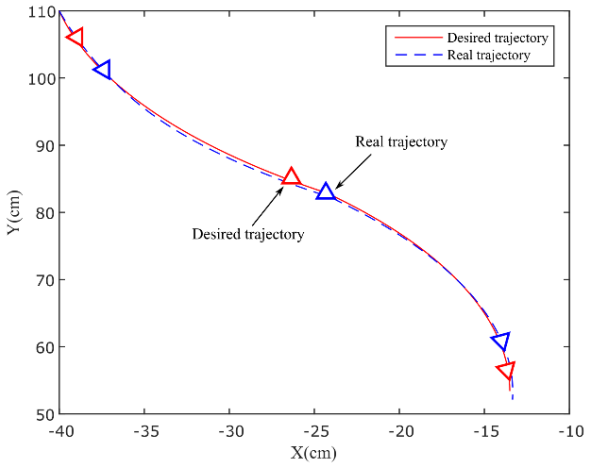

Fig. 13. Comparison between the real trajectory vs. the desired trajectory with the Fuzzy PD+I control.

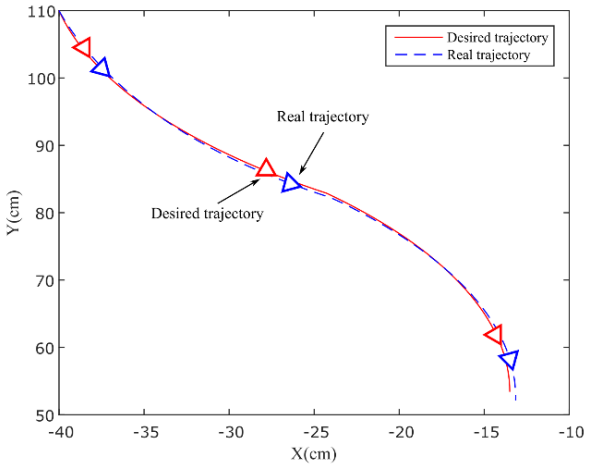

Fig. 14. Comparison between the real trajectory vs. the desired trajectory with the PID control.

Comparative results of the transient response and steady-state error can be seen in Fig 15, where the fuzzy PD+I controller has an underdamped response hence it has the faster rising time with a very small overshoot with no oscillations; on the other hand, the PID controller has an overdamped response with a low rising time. The fuzzy PD+I get very close around the steady-state almost two seconds.

Table V shows that the maximal error occurred at the beginning of the path when the transient response took place, the Fuzzy PD+I had an undershoot of 0.4224 cm, and the undershoot of the PID was 0.4565 cm. In the first curvature, the MSE of the steady-state is approximately 19% higher in the fuzzy PD+I controller. However, in the second curvature, in the fuzzy PD+I controller the MSE of the steady-state error is approximately 83% smaller than in the PID controller. The overall MSE error for the fuzzy PD+I is 0.4224 and for the PDI is 0.4565.

To better understand the difference in the steady-state error of both controllers, we plotted the steering error (E_P) vs. the error change (E_D); we show the results in Figures 16 and 17. In the fuzzy PD+I the plot of the error radius is smaller than in the PID controller, and the plot of the fuzzy PD+1 finish closer to zero than in the PID.

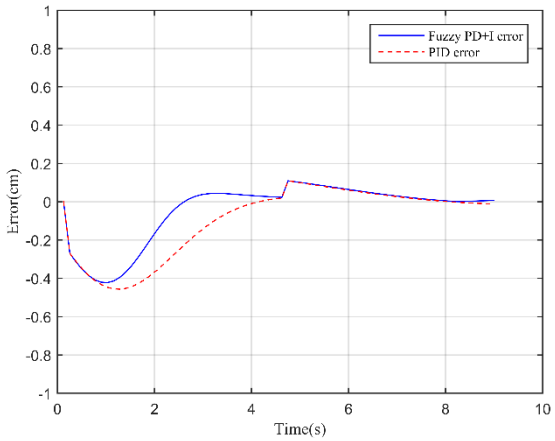

Fig. 15. Fuzzy PD+I error vs PID error.

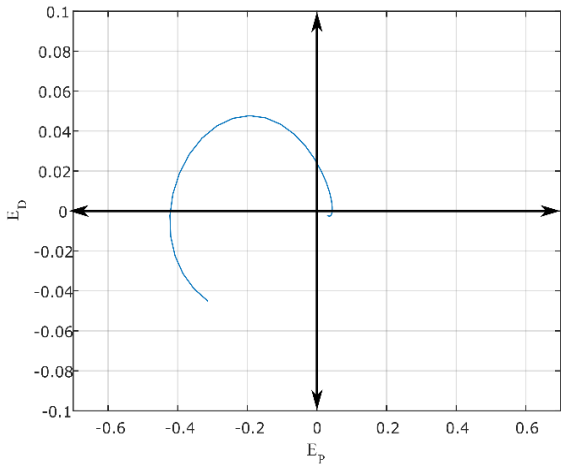

Fig. 16. Error and error change behavior through time with a Fuzzy PD+I control.

IV. Conclusions

In this paper, we have developed the mathematical basis for the route tracking for the parallel parking of an autonomous vehicle as well as the Fuzzy PD+I controller. Simulation results of the closed-loop system demonstrate the effectiveness of the proposed methodology. Mathematical tools such as geometry were used to calculate the parking

arches of the vehicle, while the steering angle was calculated using heuristic methods as well as the parking start position.

In the literature there are previous works on parallel parking, even already implemented in real vehicles. The experiments of this work are based in a 1:10 scaled prototype car. We have presented a direct comparison of the fuzzy PD+I controller with a digital PID controller, where the experiments have shown that the fuzzy PD+I controller has the better performance to achieve the parking maneuver in parallel. In general, the experiments demonstrated that the fuzzy PD+I controller can track the entire path with a maximal error around 7.5% and an overall MSE around 37% smaller than using the PID controller. This results will have more impact when the proposed controller be implemented in a real car, where for example, an error of 0.5 cm will represent 5 cm, which is a considerable amount for many cars and parking spaces.

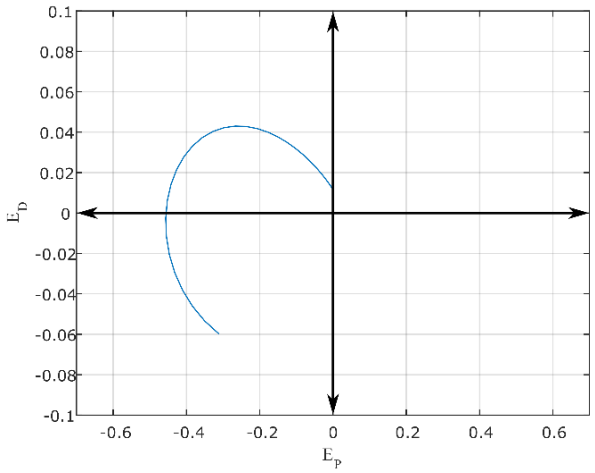

Fig. 17. Error E_P and error change E_D behavior through time with a PID control.

To properly adjust the fuzzy controller parameters we had to perform several tests, hence as future work, we are planning to use evolutionary computation embedding the system into a GPU to adapt online different membership functions for a real vehicle, in this way more accidents could be avoided due to the optimal tuning of the fuzzy controller.

REFERENCES

[1] N. Ghita, M. Kloetzer, "Trajectory planning for a car-like robot by environment abstraction", Robotics and Autonomous Systems, vol. 60, pp609 – 619, 2012

[2] D. G. Kidd, A. T. McCartt, "Differences in glance behavior between drivers using a rearview camera, parking sensor system, both technologies, or no technology during low-speed parking maneuvers", Accident Analysis and Prevention, no. 87, pp92-101, 2015

[3] D. G. Kidd, A. Brethwaite, "Visibility of children behind 2010–2013 model year passenger vehicles using glances, mirrors, and backup cameras and parking sensors", Accident Analysis and Prevention, no. 66, pp158-167, 2014

[4] M. D. Keall, B. Fildes, S. Newstad, "Real-world evaluation of the effectiveness of reversing camera and parking sensor technologies in preventing backover pedestrian injuries", Accident Analysis and Prevention, no. 99, pp39-43, 2017

[5] J. Shin, H. Jun, "A study on smart parking guidance algorithms", Transportation Research Part C, no. 44, pp299-317, 2014

[6] D.M. Filatov, E.V. Serykh, M.M. Kopichev, A.V. Weinmesiter, "Autonomous parking control system of four-wheeled vehicle", IEEE, pp102-107, 2016

[7] A. Gupta, R. Divekar. "Autonomous Parallel Parking Methodology for Ackerman Configured Vehicles". ACEEE International Journal on Communication, vol. 1, no. 2, pp22-27, 2010

[8] M. Chirca, M. Guillaume, C. Roland, C. Debain, R. Lenain, "Autonomous valet parking system architecture", IEEE International Conference of Intelligent Transportation Systems, vol. 18, pp2619-2624, 2015

[9] S. Choi, C. Boussard, B. d' Andréa-Novel, "Easy path planning and robust control for automatic parallel parking", 18th IFAC world congress, pp656-661, 2011

[10] B. Li, Z. Shao, "Simultaneous dynamic optimization: A trajectory planning method for nonholonomic car-like robots", Advances in Engineering Software, vol. 87, pp30-42, 2015

[11] Z. Liang, G. Zheng, J. Li, "Automatic parking path optimization based on Bezier curve fitting", IEEE International Conference on Automation and Logistic, pp583-587, 2012

[12] Y. Y. Aye, K. Watanabe, S. Maeyama, I. Nagal, "An automatic parking system using an optimized image-based fuzzy controller by genetic algorithms", Artificial Life and Robotics, no. 22, pp139-144, 2017

[13] B. Lee, Y. Wei, I. Yuan Guo, "Automatic parking of self-driving car based on lidar",The International Archives of the photogrammetry, remote sensing and spatial information sciences, vol. 42, pp241-246, 2017

[14] N. Scicluna, E. Gatt., O. Casha, I. Grech, J. Micallef, "FPGA-based autonomous parking of a car-like robot using fuzzy logic control", IEEE, pp229-232, 2012

[15] O. Castillo, L.T. Aguilar, S. Cardenas, "Fuzzy Logic Tracking Control for Unicycle Mobile Robots", Engineering Letters, vol. 13, no. 2, pp73-77, 2006

[16] L. Astudillo, O. Castillo, P. Melin, A. Alanis, J. Soria, L.T. Aguilar, "Intelligent Control of an Autonomous Mobile Robot using Type-2 Fuzzy Logic", Engineering Letters, vol. 13, no. 2, pp93-97, 2006

[17] A. Prasad, Sharma B, Vanualailai J, "A solution to the motion planning and control problem of a car-like robot via a single-layer perceptron". Robotica, vol. 32, no. 6, pp935–952, 2014

[18] E. Masehian, H. Kakahaji, "A nonholonomic random replanner for navigation of car-like robots in unknown environments", Robotica, vol. 32, no. 7, pp1101–1123, 2014

[19] J. Ni, J.B. Hu, "Dynamic control of autonomous vehicle at driving limits and experiment on an autonomous formula racing car", Mechanical Systems and Signal Processing, vol. 90, pp154-174, 2017

[20] Nguyen, H. Trong, D. H. Kim, C. H. Lee, H. K. Kim, "Mobile Robot Localization and Path Planning in a Picking Robot System Using Kinect Camera in Partially Known Environment", Lecture Notes in Electrical Engineering, vol. 415, pp686-701, 2017

[21] X. Bai, J. Davis, J. Doebbler, J. Turner, J. L. Junkins, "Modeling, Control and Simulation of a Novel Mobile Robotic System", Engineering Letters, vol. 16, no. 2, pp266-273, 2008

[22] U. Rosolia, S. De Bruyne, A. G. Alleyne, "Autonomous Vehicle Control: A Nonconvex Approach for Obstacle Avoidance". IEEE Transactions on Control System Technology, vol. 25, no. 2, pp469-484, 2017

[23] A. V. Pesterev, "A linearizing feedback for stabilizing a car-like robot following a curvilinear path", Journal of Computer and Systems Sciences International, vol. 52, no. 5, pp819-830, 2013

[24] J.-S. R. Jang, C.-T. Sun, E. Mizutani, "Neuro-Fuzzy and Soft Computing, A computational approach to learning and Machine Intelligence", Ed. Prentice Hall, pp11-90 1997

[25] B. Bhushan, N. Jha, S. Devra, S. S. Pillai, "Performance analysis of PID and Fuzzy PD+I controller on nonlinear systems," 2014 IEEE International Advance Computing Conference (IACC), pp1195-1200, 2014

[26] Z. Kovacic, S. Bogdan "Fuzzy Controller Design Theory and Applications", Taylor & Francis Group, pp75-79, 2006

[27] R. Matousek, P. Minar, S. Lang, M. Seda "HC12: Efficient PID Controller Design" Engineering Letters, vol. 20, no. 1, pp42-48, 2012

[28] R. Dorf, R. Bishop, "Modern Control Theory: 13th Edition. Pearson: England, 2016