2019 IEEE 4th Conference on Technology In Electrical and Computer Engineering (ETECH 2019)
Information and Communication Technology (ICT) Tehran, Iran

Designing a Fuzzy Logic Controller for Automated Backward Parking Car

Hamid Keshmiri Neghab
Faculty of computer engineering
Azad university of Mashhad
Hamid.keshmiri@gmail.com

Hamed Keshmiri Neghab
Faculty of control engineering
Ferdowsi University of Mashhad
Keshmirineghab.hamed@mail.um.ac.ir

Naser Pariz
Faculty of control engineering
Ferdowsi University of Mashhad
n-pariz@um.ac.ir

Abstract- **This paper describes a control system which automatically parks a scaled automobile inside a rectangular reduced space given certain conditions and making decisions based in fuzzy logic. The control is developed by the processing of entry variable data from simulated sensors of a specific scenario, and the run of three models in cascade to achieve a decision-action method. Finally, this paper shows a description of the way this project was achieved, and concludes with the acceptable results that a fuzzy control can provide in a management of this kind of mechanism by the use of taking decision models.**

Keywords—car parking, backward parking, fuzzy logic, fuzzy controller.

I. INTRODUCTION

Nowadays, the increase of using cars has highlighted in an urban area. However, most drivers get in troubles while trying to find an empty parking space. Some certain systematic complications can be time-consuming with inevitable risks like obstacles. This ability of vehicle to find a desire and vacant parking spot and park there by itself will be an evolution to the car industry. A research has conducted on some car manufactures which offer self-parking ability on their products [1]. Another project have been applied on parallel parking for autonomous vehicles with fuzzy logic [2],[3].

This project demonstrates the use of fuzzy logic to simulate parking a vehicle in a specific location. First we set the vehicle initial position and specify an initial vehicle angle. Then, we activated fuzzy controller to park the vehicle using fuzzy logic and verify the car moving to the garage.

II. METHODS

A fuzzy system consists of three main parts: linguistic variables, membership functions, and rules. In This chapter we describe the general process of designing a fuzzy system.

A. Creating Linguistic Variables

Linguistic variables represent, in words, the input variables and output variables of the system we want to control. When we create a linguistic variable to represent an input or output variable, decide how many linguistic terms, or categories of values of the linguistic variable, we want to create. Linguistic variables usually have an odd number of linguistic terms, with a middle linguistic term and symmetric linguistic terms at each extreme. Here, we want to automate a vehicle to park itself from an arbitrary starting position. A driver can control the vehicle by constantly evaluating the current status of the vehicle, such as the distance from the target position and the orientation of the vehicle, to derive the correct steering angle.

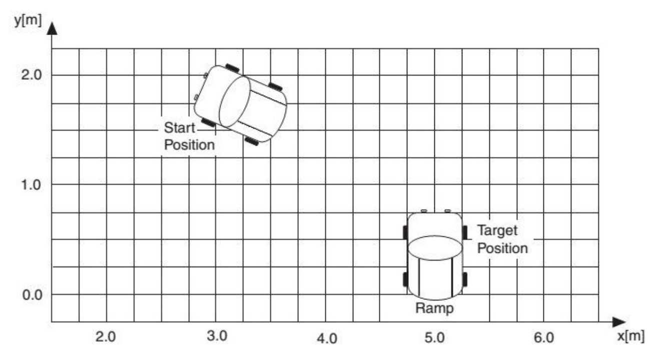

Fig 1. Automating Vehicle Parking

We define two input linguistic variables for this project. Vehicle Position x represents the vehicle position in relation to the destination. Vehicle Orientation β represents the orientation of the vehicle. We also define an output linguistic variable, Steering Angle ϕ, to represent the steering angle of the vehicle that we want to control.

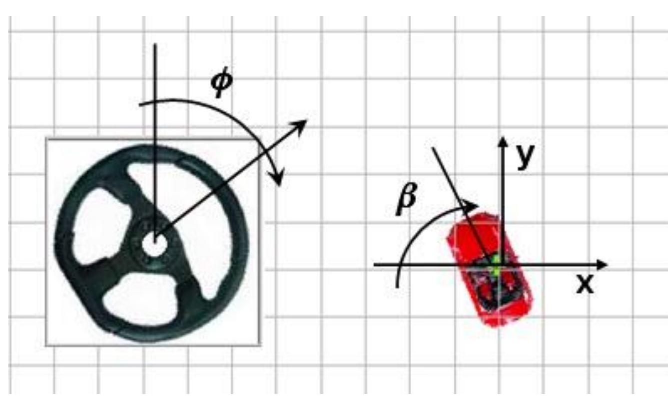

Fig 2. Vehicle Position and Orientation

By activating fuzzy controller, the vehicle starts to move and make curves to locating in correct position. Here is the equations to calculate the directions and finding the garage.

$$arc = (\, 2 * {}^{\pi}/_{360}\,) \tag{1}$$

$$R = a * \cot(\mathrm{arc} * \mathrm{Phi}) \tag{2}$$

$$Eta = Dir * 50 * \tan(arc * Phi) \tag{3}$$

$$Bo = \Big(\big((Bi - Eta) > -90\big) \&\& \big((Bi - Eta) < 270\big)\Big)\,?\, Bi - Eta : \big((Bi - Eta) < -90\big)\,?\, Bi - Eta + 360 : Bi - Eta - 360 \tag{4}$$

$$Xo = Xi + R * (\sin(arc * Bi) - \sin(arc * Bo)) \tag{5}$$

$$Yo = Yi + R * (\cos(arc * Bi) - \cos(arc * Bo)) \tag{6}$$

This equations are for making curves, where R is turning circle radius as function of the steering angle Phi. Eta is driving angle as function of the steering angle Phi. Here arc is constant for converting angle in degree to radian. We define Bi as current vehicle orientation (beta input) and Bo as new vehicle orientation (beta output).

$$aBeta = \left(2 * {}^{\pi}/_{360}\right) * Beta \tag{7}$$

$$Xo = Xi + Dir * s * \cos(Beta) \tag{8}$$

$$Yo = Yi - Dir * s * \sin(Beta) \tag{9}$$

For moving straight we use this equations. Here steering angle Phi is 0 and s is drive distance which is 0.26 and Dir is for forward moving -1 and for backward moving +1.

We can define linguistic terms of Left, Left Center, Center, Right Center, and Right for the Vehicle Position x input linguistic variable to describe the possible positions of the vehicle in relation to the destination. We also can define linguistic terms of Left Down, Left, Left Up, Up, Right Up, Right, and Right Down for the Vehicle Orientation β input linguistic variable to describe the possible orientations of the vehicle. The linguistic terms of the Steering Angle ϕ output linguistic variable must represent both the direction and magnitude that the steering angle changes. Therefore, we can use the linguistic terms Negative Large, Negative Medium, Negative Small, Zero, Positive Small, Positive Medium, and Positive Large for this output linguistic variable.

B. Creating Membership Functions

Membership functions are numerical functions corresponding to linguistic terms. A membership function represents the degree of membership of linguistic variables within their linguistic terms.

C. Creating a Rule Base

Rules describe, in words, the relationships between input and output linguistic variables based on their linguistic terms. A rule base is the set of rules for a fuzzy system.

To create a rule, we must specify the antecedents, or IF portions, and consequents, or THEN portions, of the rule. Associate an input linguistic variable with a corresponding linguistic term to form an antecedent.

The consequent of a rule represents the action we want the fuzzy controller to take if the linguistic terms of the input linguistic variables in the rule are met. A consistent rule base is a rule base that has no contradictory rules. The total number N of possible rules for a fuzzy system is defined by the following equation:

$$N = p_1 * p_2 * \ldots * p_n$$

Where p_n is the number of linguistic terms for the input linguistic variable n. If each input linguistic variable has the same number of linguistic terms, the total number N of possible rules is defined by the following equation:

$$N = p^m$$

Where p is the number of linguistic terms for each input linguistic variable and m is the number of input linguistic variables. The Vehicle Position x input linguistic variable has five linguistic terms, and the Vehicle Orientation β linguistic variable has seven linguistic terms. Therefore, the rule base of the vehicle maneuvering example consists of $N = 5 * 7 = 35$ rules. We can document the complete rule base in matrix form, as shown in Table 1.

Each column or row represents an antecedent of a rule. The term at the intersection of a column and a row is the consequent of the rule corresponding to the aggregated rule antecedent. For example, the following rule is highlighted in Table 1.

IF Vehicle Position x is Left Center AND Vehicle Orientation β is Left, THEN Steering Angle ϕ is Negative Small.

TABLE I. COMPLETE RULE BASE FOR THE VEHICLE MANEUVERING

AND		Vehicle Position X [m]				
		Left	Left Center	Center	Right Center	Right
Vehicle Orientation β [°]	Left Down	Negative Small	Negative Medium	Negative Medium	Negative Large	Negative Large
	Left	Positive Small	Negative Small	Negative Medium	Negative Large	Negative Large
	Left Up	Positive Medium	Positive Small	Negative Small	Negative Medium	Negative Large
	Up	Positive Medium	Positive Medium	Zero	Negative Medium	Negative Medium
	Right Up	Positive Large	Positive Medium	Positive Small	Negative Small	Negative Medium
	Right	Positive Large	Positive Large	Positive Medium	Positive Small	Negative Small
	Right Down	Positive Large	Positive Large	Positive Medium	Positive Medium	Negative Small

D. Specifying an Antecedent Connective

If a rule has more than one antecedent, we must specify an antecedent connective to determine how to calculate the truth value of the aggregated rule antecedent. Because linguistic variables can have partial degrees of membership within linguistic terms, we cannot use Boolean operators from conventional dual logic as antecedent connectives. The PID and Fuzzy Logic Toolkit uses the following antecedent connectives instead.

- **AND (Minimum):**
  µA•B= min (µA, µB)
- **AND (Product):**
  µA•B= (µA, µB)
- **OR (Maximum):**
  µA+B= max (µA, µB)
- **OR (Probabilistic):**
  A+B= ((A+B)-(AB))

Assume the following rules are invoked for a particular set of input values.

(1) IF Vehicle Position x is Center (degree of membership = 0.8) AND (Minimum)

Vehicle Orientation β is Left Up (degree of membership = 1.0) = 0.8 THEN Steering Angle ϕ is Negative Small

(2) IF Vehicle Position x is Right Center (degree of membership = 0.1) AND (Minimum)

Vehicle Orientation β is Left Up (degree of membership = 1.0) = 0.1 THEN Steering Angle ϕ is Negative Medium

If these two rules are the only rules invoked for a given set of input values, the other linguistic terms for the Steering Angle ϕ output linguistic variable have a truth value of 0. The following table describes the final truth values for each of the linguistic terms.

Negative Large	to a degree of	0.0
Negative Medium	to a degree of	0.1
Negative Small	to a degree of	0.8
Zero	to a degree of	0.0
Positive Small	to a degree of	0.0
Positive Medium	to a degree of	0.0
Positive Large	to a degree of	0.0

We can specify a degree of support, between 0 and 1, for each rule of a fuzzy system. The degree of support represents the relative significance of each rule and allows for fine-tuning of the rule base. In most cases, the degree of support is 1. The final weight of a rule is equal to the degree of support multiplied by the truth value of the aggregated rule antecedent.

E. *Defuzzification Method*

Defuzzification is the process of converting the degrees of membership of output linguistic variables within their linguistic terms into crisp numerical values.

In this paper we use Center of Area (CoA). In the Center of Area (CoA) defuzzification method, the fuzzy controller first calculates the area under the scaled membership functions and within the range of the output variable.

$$CoA = \frac{\int_{x_{min}}^{x_{max}} f(x).xdx}{\int_{x_{min}}^{x_{max}} f(x)dx}$$

Where CoA is the center of area, x is the value of the linguistic variable, and x_{min} and x_{max} represent the range of the linguistic variable. The Center of Area defuzzification method effectively calculates the best compromise between multiple output linguistic terms.

III. RESULT AND DISSCUTION

The following figures illustrate the resulting graph of the estimated states and estimated outputs. The *x* and *y* positions of the estimated states closely match the true system states.

In this figure, we have a grid which car apply the changes toward the garage. Beside this we have vehicle angle which we can move the cursor to set the angle of the vehicle. We can set the angle of vehicle from -90 to 270 degree. Steering angle shows the degree of the steering from the start point to the end point. By activating fuzzy controller, vehicle moved towards garage very smoothly by moving forward and backward and when it reached the garage the controller stops and the program finished. By clicking stop vehicle stops.

Here, we have initial position of car (2, 4) with initial vehicle angle of 90 and steering angle of 0. In the following, figure 3 illustrates the car parking figure with initial values and destination when the vehicle arrived.

Figure 4 shows how the vehicle moves in X and Y grid and figure 5 illustrates the total time of operation. After start, the vehicle moved forward and make a curve then moved backward, again it moved forward with vehicle angle of 90 degree and then moved backward to the garage. In figure 5 we can see that the total time of operation is 195 seconds.

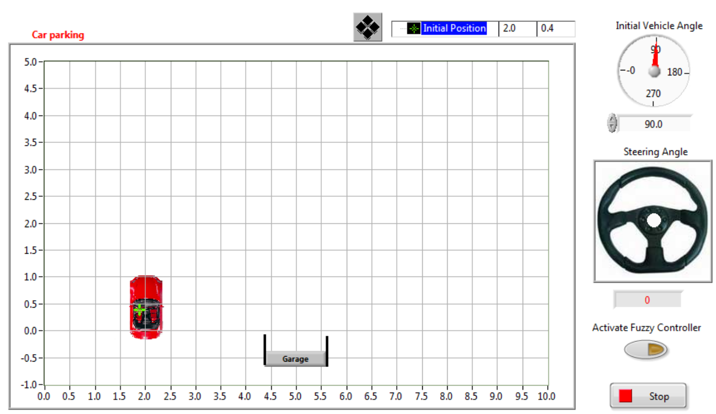

Fig 3. Initial position with 90 degree

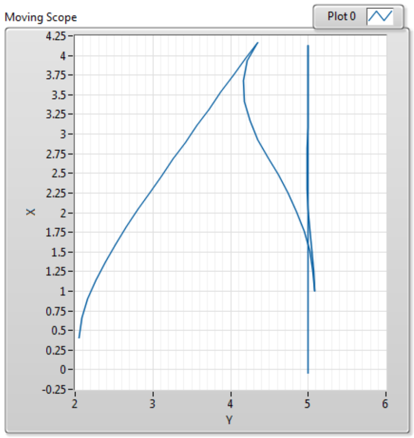

Fig 4. Moving scope of vehicle in XY grid with angle 90

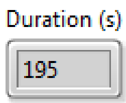

Fig 5. Total time of angle 90

Now we set the initial angle of vehicle on 0 degree and we can see the results of this position. Figure 6 shows the initial position of vehicle and the destination. Figure 7 shows how vehicle moves in XY grid and in figure 8 we can see that the total time is 236 seconds.

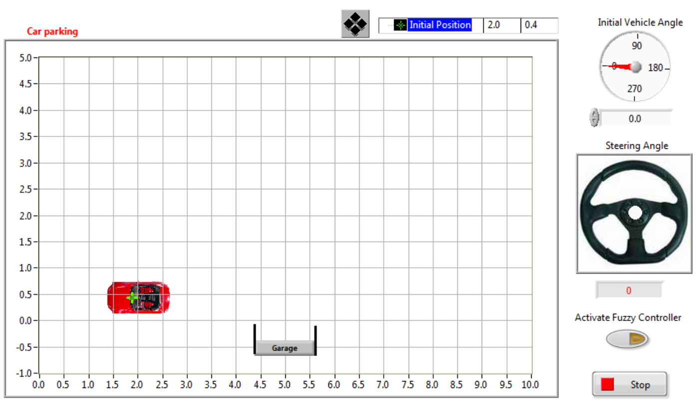

Fig 6. Initial position with vehicle angle 0

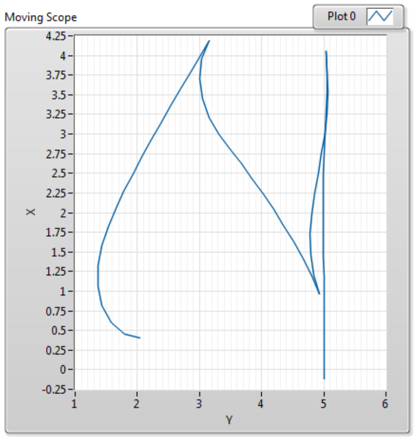

Fig 7. Moving scope of vehicle in XY grid with angle 0

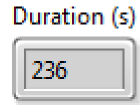

Fig 8. Total time of angle 0

Now we set the vehicle angle on -90 or 270 and here is the results. In figure 9 we can see the initial position of vehicle and the destination. Figure 10 shows how the vehicle moves toward the garage and figure 11 illustrates total time of vehicle which means how many seconds took to the end.

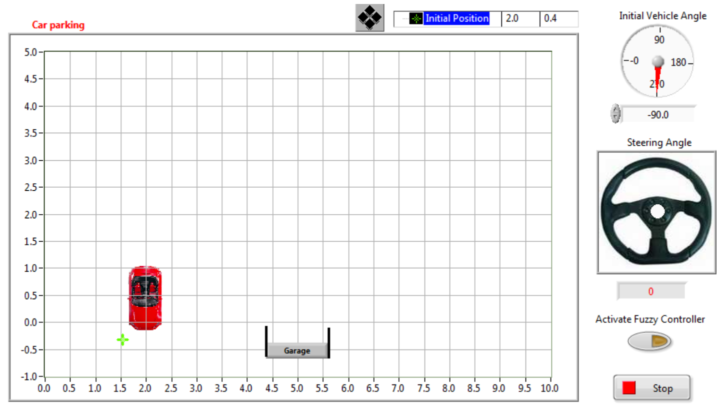

Fig 9. Initial position with vehicle angle -90

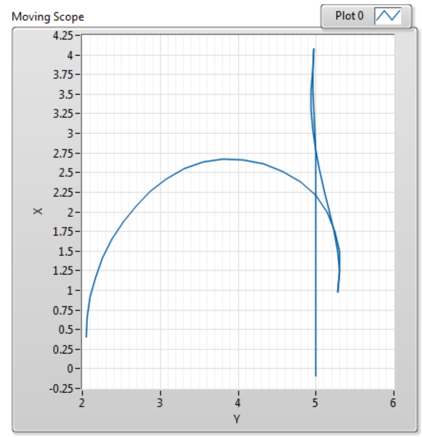

Fig 10. Moving scope of vehicle in XY grid with angle -90

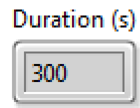

Fig 11. Total time of angle -90

Now we want to try another vehicle angle and we set that on 180 degree. In the following we can see the results. Here is figures 12, 13 and 14.

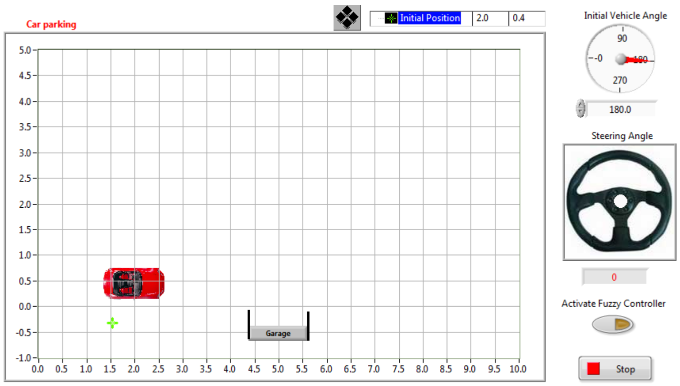

Fig 12. Initial position with vehicle angle 180

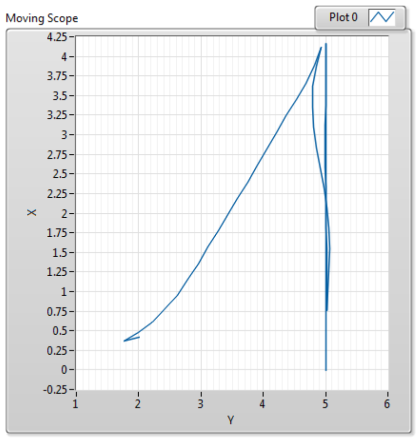

Fig 13. Moving scope of vehicle in XY grid with angle 180

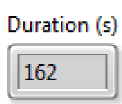

Fig 14. Total time of angle 180

IV. CONCLUSION

This paper present a system for car self-parking by using fuzzy logic controller. This controller provided by Labview in a free collision space and it was shown that this controller move very smoothly and accurately find the garage and park there. In this system, it's important to set precise parameters such as membership functions and rules which we provided accurately to cover all aspects of the problem. By moving cursor and setting new situations for car, it can be seen that car move very carefully and precisely to the destination without any mistakes. Here we set 4 vehicle angles such as 90, 0, -90 and 180 and our aim is to find the fastest position to parking in garage. So, we investigated time for each vehicle angle and as results has shown the vehicle angle with 180 degree parked very quickly in parking lot because it moves straight without any curves. This paper showed that our fuzzy logic controller work very accurately.

REFRENCES

1. Kane, S., *Family Cars with Self-Parking Technology.* Retrieved from The Car Connection: http://www. thecarconnection. com/news/1067819_2012-family-cars-with-selfparking-technology, 2012.
2. Panomruttanarug, B., et al. *An emulation of autonomous parallel parking system using fuzzy logic control*. in *2009 ICCAS-SICE*. 2009. IEEE.
3. Lian, K.-Y., C.-S. Chin, and T.-S. Chiang. *Parallel parking a car-like robot using fuzzy gain scheduling*. in *Proceedings of the 1999 IEEE International Conference on Control Applications (Cat. No. 99CH36328)*. 1999. IEEE.