



Rogerio Cisi &lt;rogeriocisi@gmail.com&gt;

---

## Fuzzy parking

---

Rogerio Cisi &lt;rogeriocisi@gmail.com&gt;

15 de maio de 2016 16:05

Para: Rogerio Cisi &lt;rogeriocisi@gmail.com&gt;

# A fuzzy parking system

In this page you will find a fuzzy logic control system capable of parking a car.

You can play around with the applet, moving the car in any place with the mouse, and moving the parking room lenght, and then pressing "go". You can try the results using the Zadeh, Product and Lukasiewicz fuzzy logics and read a paper describing the theory behind it. I hope you find it interesting.

NOTE: You need Java Plug-in to run this applet. If it does not work, [get the latest Java Plug-in here](#).

It has been developed by Carlos Martín as a final project for the course "Razonamiento aproximado y con incertidumbre" in the Facultad de Informática of Universidad Complutense de Madrid.

Here is the complete paper of this project:

INFLUENCE OF DIFFERENT FUZZY strategies ON A decision system FOR car parking

carlos martín sánchez, luis garmendia salvador

Dpto. Ingeniería Software e Inteligencia Artificial. Facultad de Informática, Universidad Complutense de Madrid, 28040-Madrid, Spain

MATILDE SANTOS PEÑAS

Dpto. Arquitectura de Computadores y Automática, Facultad de Informática, Universidad Complutense de Madrid, 28040-Madrid, Spain

RAMÓN GONZÁLEZ DEL CAMPO

Dpto. Ingeniería Software e Inteligencia Artificial. Facultad de Informática, Universidad Complutense de Madrid, 28040-Madrid, Spain

In this paper, the design of a fuzzy decision system for car parking is presented. The modeling of the problem, the physics involved, the fuzzy sets assigned to the linguistic variables and the inference rules are explained. Different fuzzy operators have been proved and the results have been compared. The system is proposed as a benchmark to show the influence of the different fuzzy strategies on the decision. The final implementation of the parking system in a Java Applet can be tested in the web using any internet navigator.

**Keywords:** fuzzy operators, decision system, approximate reasoning, automatic parking, autonomous driving.

## 1. Introduction

This paper describes the design and implementation of a fuzzy decision system that shows different fuzzy

strategies for parking a car. The modeling of the problem, the physics involved, the fuzzy sets assigned to the linguistic variables and the inference rules are explained. For each of the different alternatives that are provided by the system, some criterions are calculated that allows us to decide the best one. That is, the best sequence of motions to park the system with fewer maneuvers, smoothly, etc.

The main goals of this parking system are:

- To park the car from any position and with any available room (given by the user in the implemented interface).
- To maintain the fuzzy rules as simple as possible.
- To have a tool where we can see the influence of different fuzzy strategies on the behaviour of that decision system.

So the resulting system is a good balance between simplicity, correct modeling and effectiveness. The system is proposed as a benchmark to show the influence of the logic operators on the decision, as in [Montenegro].

The final implementation of the parking system in a Java Applet can be tested in the web using any internet navigator at <http://www.fdi.ucm.es/profesor/lgarmend/SC/aparca/index.html>.

The paper is organized as follows. In section 2 the problem and the model of the car is described. In section 3 the fuzzy system that represents the car parking process is described. Section 4 shows the results of applying different fuzzy strategies. Conclusions end the paper.

## 2. Problem description

The problem description is a simplified model of the real world reverse parking strategy, with the following items:

- A car
- The street curb
- Front and rear obstacles

The goal is to park the car with the minimum number of maneuvers. A car is considered well parked when it is next to the curb, with an angle respect to it very close to zero, and it keeps certain distance to the obstacles.

### 2.1. The car model

The car is modeled as a vehicle with the ability to move forward or backward at a constant speed in the plane. No consideration was taken into account about acceleration or braking in this simplified model.

The vehicle can steer the front wheels in the interval  $[-45^\circ, 45^\circ]$ . The following equations (1) and Figure 1 [Steven 2006] describe the dynamic model of the car. The body frame of the car places the origin  $(x,y)$  at the center of rear axle, and the x-axis points along the main axis of the car;  $s$  denote the (signed) speed of the car, and  $\Phi$  the steering angle (it is negative for the wheel orientations shown in Figure 1). The distance between the front and rear axles is represented as  $L$ . If the steering angle is fixed at  $\Phi$ , the car travels in a circular motion, in which the radius of the circle is  $p$ . Note that  $p$  can be determined from the intersection of the two axes shown in Figure 1 (the angle between these axes is  $|\Phi|$ ).

$$\begin{aligned}\dot{x} &= f_1(x, y, \theta, s, \phi) \\ \dot{y} &= f_2(x, y, \theta, s, \phi) \\ \dot{z} &= f_3(x, y, \theta, s, \phi)\end{aligned}\quad (1)$$

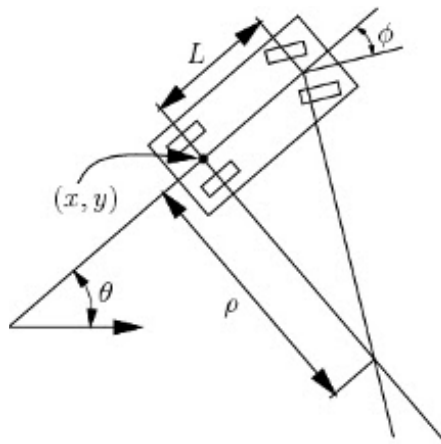


Figure 1. Car model

## 2.2. Modeling the scenario

The relevant details to describe the situation in a certain instant of time are:

- The angle formed between the car and the curb.
- The distance from the car to the curb. To measure this distance, the value of the car is taken from the most right wheel X value.
- The distance to the front and back obstacles. These distances are measured from the front and rear bumpers of the car.

One important aspect to notice is that the obstacles are not avoided as just something that the car cannot go through or something not to collide with. Instead of doing so, the distance to the obstacles is used as information that helps the car to take a decision, i.e. if the car is going forward and the front obstacle is quite near, it will decide to go backward and so on.

In order to simplify the modeling of the rules, three clearly different strategies, or maneuvers, are specified. These three strategies are shown in Figure 2, depending on the initial position of the car.

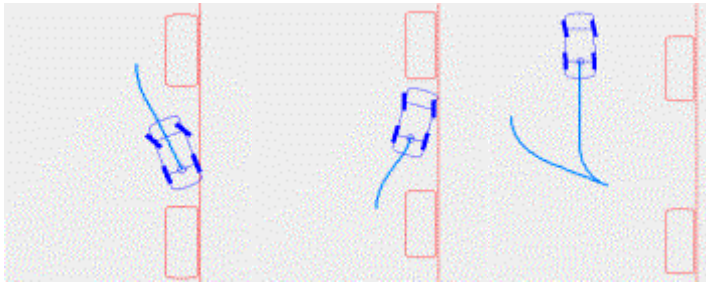


Figure 2. Three parking strategies.

In real life you would try to park in a single backward maneuver taking into account that the gap is suitable for your vehicle's size. That is what the car does as well in the applet, it goes backward while steering right, and then at the right distance from the curb, it starts to steer left.

If the car is not able to go backward any further, it will have to move forward. Depending on the distance to the curb, it will approach the curb while moving forward, or it will move away from the rear obstacle, so it gets some leeway to try the backward strategy again.

Unlike a real driver would do, the car will not try to find a good starting position, but it will try to park from wherever the user puts the car with the mouse.

### 3. The car parking fuzzy decision system

Fuzzy logic is a useful tool to represent approximate reasoning [Zadeh, Pajares]. It allows to work with vagueness and uncertain environments. In this paper we have used the software tool Xfuzzy [IMSE] to design the fuzzy-inference-based system.

The inputs of the system are shown on the left of the system displayed in Figure 3. The five input variables are: the orientation of the car (that is, the angle between the car and the curb,  $\theta$ ), the distances to the curb, to the front and to the rear obstacles, and the previous strategy.

The rules are grouped in four subsystems (Figure 3). Each of these fuzzy subsystems computes an output by the defuzzification method of the center of area.

The first three blocks are used to decide the angle in which the steering wheel should be steered depending on the strategy to be implemented: backward, forward approaching the curb, or forward moving away.

The outputs are shown on the right hand of the image (Figure 3), they are named: strategy, steering\_backward, steering\_forward\_approach, and steering\_forward\_moveaway. The three fuzzy subsystems calculate a steering output, but only one is going to be applied, the corresponding to the value of the output strategy.

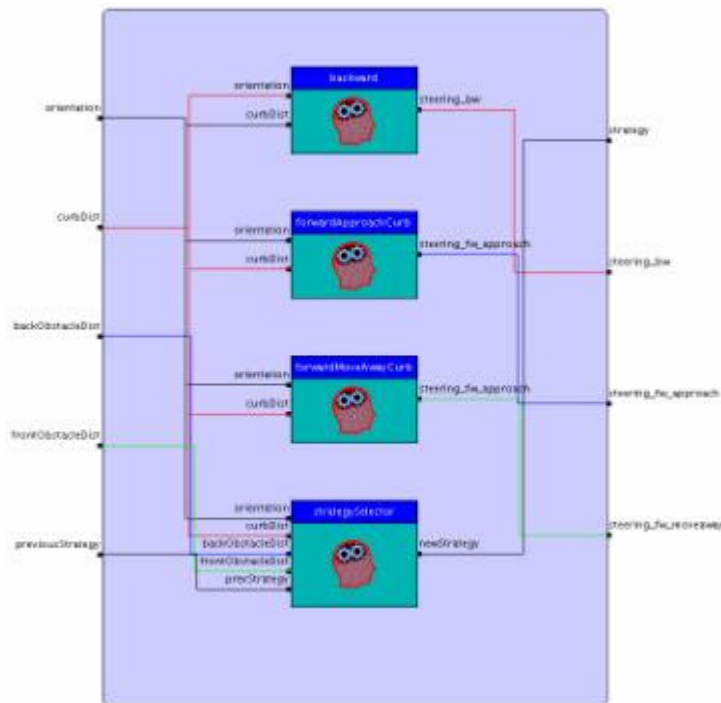


Figure 3. The fuzzy car parking system implemented in Xfuzzy.

The last group of rules decides which the best strategy to be applied is. The defuzzification function used in this case is the maximum value membership function in order to choose only one of the strategies.

### 3.1. Car parking fuzzy variables

#### 3.1.1. Input variables: Distance, Orientation and Previous Strategy

Figure 4 (left) shows the three fuzzy sets assigned to the variable distance, which has been defined for curbDist, backObstacleDist and frontObstacleDist variables. The labels correspond to “touching”, “close” and “far”, and the membership functions are trapezoidal.

The linguistic variable orientation has been defined by six fuzzy sets with trapezoidal membership (see Figure 4, right). The range goes from -90 to 90 degrees. The “straight” label means close to 0°. Although the functions representing negative and positive angles overlap between them, L1 and R1 do not cross the zero value. That way it is not possible to state that “it is clearly heading right and more or less heading left”, but “it is quite straight and also a bit turned to the right”.

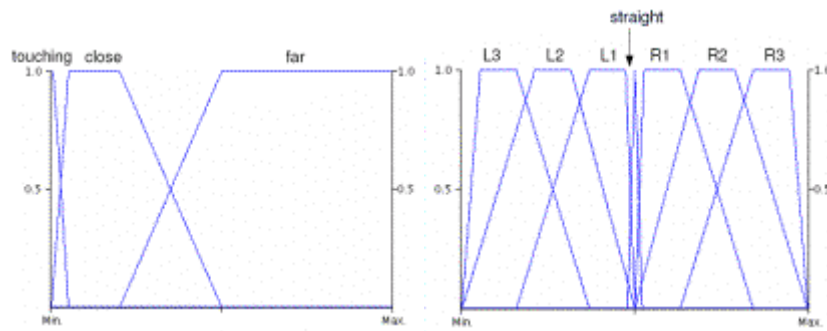


Figure 4. Fuzzy input variables: distance (left) and orientation (right).

The variable *previous\_strategy* is modeled by classical or “crisp” sets because only one strategy can be applied at a time. The four singleton values are: “stop”, “fwApproach”, “fwMoveAway” and “backward” (see Figure 5, right).

### 3.1.2. Output variables: Steering and Strategy

The fuzzy sets modeling the linguistic variables on the *steering* wheel are quite similar to the input variable orientation but the interval is now  $[-45^\circ, 45^\circ]$  and the overlap between them is a bit lower in order to produce a more precise output (Figure 5, left).

The variable *strategy* is modeled as explained above for *previous\_strategy* (Figure 5, right).

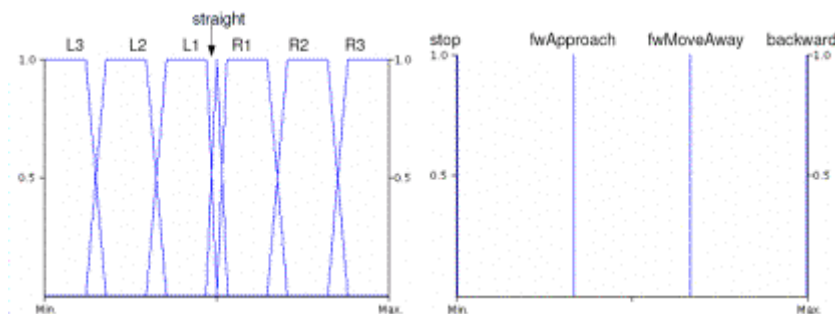


Figure 5. Fuzzy variables: steering (left) and strategy (right).

## 3.2. Car parking fuzzy rules

Keeping in mind that one of the goals of this system is to maintain the rule base as simple as possible while designing the three strategies, the following simplified notation is used:

$$\text{Distance to the curb} \rightarrow \text{Desired orientation}$$

That is, given a distance to the edge, the output should be the angle of the steering wheel that will make

the car become closer to the desired orientation. This is applied for each strategy.

Usually a premise of a rule has the conjunction (AND or &), implemented by a t-norm [Sweicher, Sklar, 1960], that combines more than one linguistic variables such as: "if  $x$  is  $X$  and  $y$  is  $Y$ ", or " $x==X$  &  $y==Y$ ", where  $x, y$  are fuzzy variables (for example, CurbDist), and  $X, Y$  are fuzzy sets.

In this application we also use expressions such as " $x < X$ ", that defines the fuzzy premise " $x$  is less than  $X$ ". For example, given a fuzzy set  $X$ , the fuzzy sets "less than  $X$ ", "less or equal than  $X$ ", "more than  $X$ ", are defined in figure 6 [IMSE]. For example, "less than touching" in the variable CurbDist means that the car is on the sidewalk.

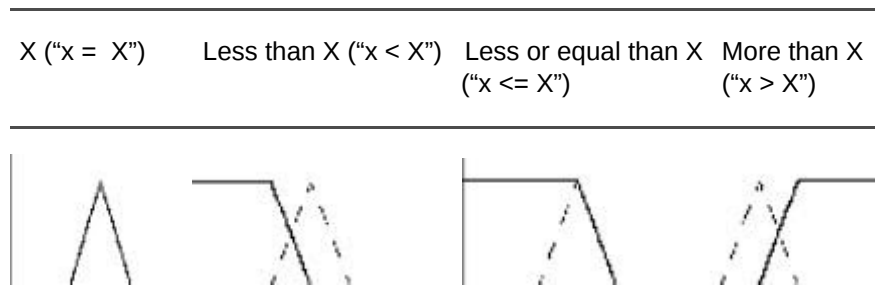


Figure 6. Interpretation of some linguistic labels used in the premises of the rules

The following four fuzzy subsystem have been implemented.

### 3.2.1. Backward strategy

According to the notation for the rules stated above, when the Previous\_strategy is backward, the rules to compute the steering\_backward angle can be expressed in a general way as (Table 1):

Table 1. Fuzzy general rules for the backward strategy

| Distance           | Desired_Orientation |
|--------------------|---------------------|
| Far                | L3                  |
| Close              | L1                  |
| Touching           | Straight            |
| Less than touching | R2                  |

That means that if the car is far from the curb, it should approach it turning to the left. But if the car is close to the curb, the approaching angle should be only a bit to the left. Finally, if the car is very close to the curb, or touching it, it should approach the curb with an angle next to zero. But if the distance to the

curb is less than touching, then the car is on the sidewalk. In that case, it should be approaching the curb at an angle quite to the right.

Let us take the first simplified rule as an example. When the car is far from the curb then CurbDist is bigger than close and the resulting output tries to drive the steering wheel so the car reaches that L3 desired angle.

To get this desired orientation L3, if the car's angle is less than L3 -more to the left-, the steering output must be R3. If the angle is more than L3 -more to the right-, then the steering output should be L3. And when the car is at the desired angle, the steering should be straight (zero degrees).

Therefore, the first row of Table is implemented in Xfuzzy by the following set of rules:

```
if(CurbDist == far & angle < L3) -> steering_backward = R3;

if(CurbDist == far & angle > L3) -> steering_backward = L3;

if(CurbDist == far & angle == L3) -> steering_backward = straight;
```

The same is done to implement the rest of the rules of Table 1.

### 3.2.2. Forward approaching the curb

The simplified rules for implementing the steering angle when the forward strategy is applied are (Table 2):

Table 2. Fuzzy general rules for the forward strategy

| Distance           | Desired_Orientation |
|--------------------|---------------------|
| Far                | R2                  |
| Close              | R2                  |
| Touching           | Straight            |
| Less than touching | L1                  |

That means that the car will try to move forward with an angle R2 (quite to the right) when it is not very near the curb. When the car is touching the curb, it will try to keep a straight angle. If the distance to the curb is less than zero, then the car will look for a left orientation angle.

Each of this rule of Table 2 is implemented by a set of fuzzy rules as in the previous example.

### 3.2.3. Forward moving away from the rear obstacle

There are not rules that define what happens when the car is close or touching the curb, because in that case this strategy is not chosen. In the other cases, when the car is far, the rules are:



Table 3. Fuzzy general rules for the forward moving away strategy

| Distance | Desired_Orientation |
|----------|---------------------|
| Far      | Straight            |

The rules that implement this statement are:

```

if(CurbDist == far & angle > straight) -> steering_fw_mov_away = L3;

if(CurbDist== far & angle==straight) -> steering_fw_mov_away = straight;

if(CurbDist == far & angle < straight) -> steering_fw_mov_away = R3;

```

The steering angle in this case refers to steering\_forward\_moving\_away as this is the strategy that this fuzzy subsystem implements.

### 3.2.4. Strategy selector

This rule base, or inference engine, decides which one of the three strategies must be applied. The reasoning is as follows,

- If the car is touching the curb and its orientation is straight, then stop; it is successfully parked.
- If there is enough back space, then go backward.
- If the back obstacle is really near and the car is close to (or near) the curb, then go forward towards the curb.
- If the back obstacle is really near and the car is far from the curb, then go forward moving away from the rear obstacle.

When the car is touching the back obstacle, it will move forward. But then, it will not be touching anymore the obstacle, and the next movement is going to be backward. That way, the car could be moving a little forward, then a little backward, then a little forward again, and so on.

That is the reason why there is an input that is the “previous strategy”. Knowing the previous strategy, this logic can be added to the preceding rules:

*If the previous strategy was forward (approaching the curb, or moving away), then continue moving forward (in the same strategy, approaching or moving away), unless the back obstacle is far, the front obstacle is very near, or the car is correctly parked.*

## 4. Results of the car parking fuzzy decision system with different fuzzy operators

Different fuzzy logic operators have been applied to check which the best strategy for parking is, and the results are compared in terms of the number of iterations (that is, the movements that the car has to carry out to park).

The next three logics are implemented to compute de conjunction (t-norm) operators in the premise of the

rules [Trillas]:

Table 4. Different implementation of the fuzzy operators

|             | and              | or             |
|-------------|------------------|----------------|
| Zadeh       | $\min(x,y)$      | $\max(x,y)$    |
| Łukasiewicz | $\max(0, x+y-1)$ | $\min(1, x+y)$ |
| Product     | $x*y$            | $x + y - x*y$  |

Figure 7 shows an example of the results of the different fuzzy logics when the car is far from the curb and at different positions regarding the obstacles. The number of maneuvers is presented in terms of the initial position of the car regarding the gap (Behind, inFront, at\_the\_same\_high, see Figure 2).

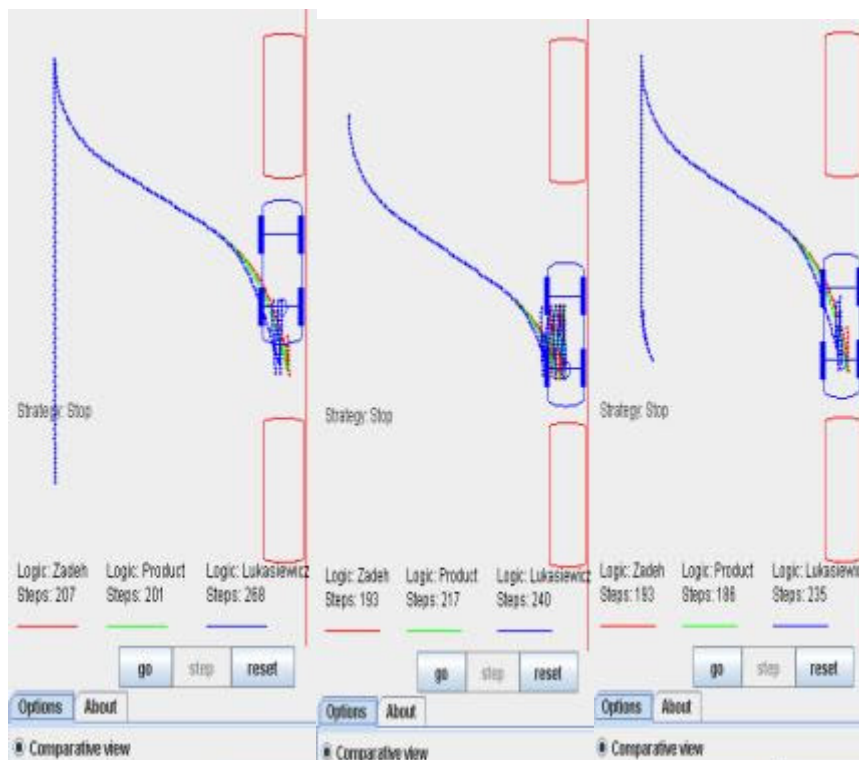


Figure 7. Different initial positions of the car regarding the gap and comparative of fuzzy logics.

In the following figures (8 to 10), the results of applying different fuzzy logic for different distances of the car to the curb and to the obstacles are shown. The car is moving from an initial position near to the curb (80) to a distance of 400 from the edge, in the X-axis. The Y-axis is fixed for each case (relative distance to the gap). When the number of movements is 1000 that means that the car did not achieve to park.

First, in Figure 8, the initial position of the car is backward the obstacle, that is, behind the gap. In this case, if the car is near the curb, the best logic is Zadeh, but while is moving away from the edge (from 130 to 200), Lukasiewicz performs better.

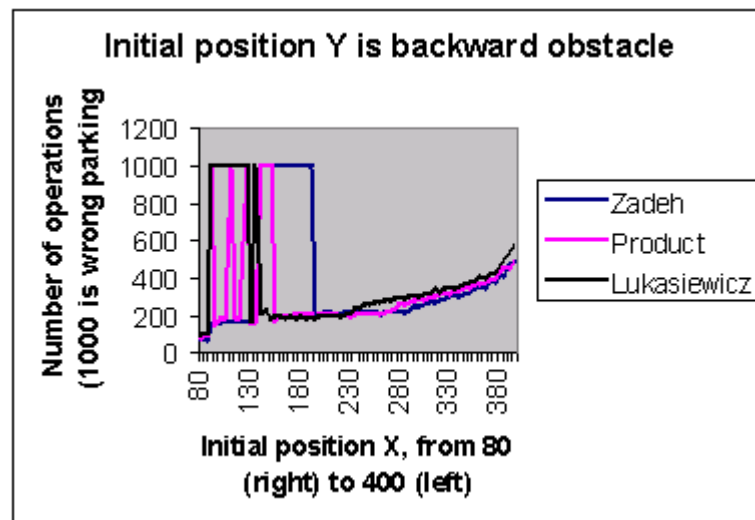


Figure 8. Initial position of the car behind the gap.

Figure 9 shows the manoeuvres the car makes from an initial position at the high of the gap, that is, between obstacles.

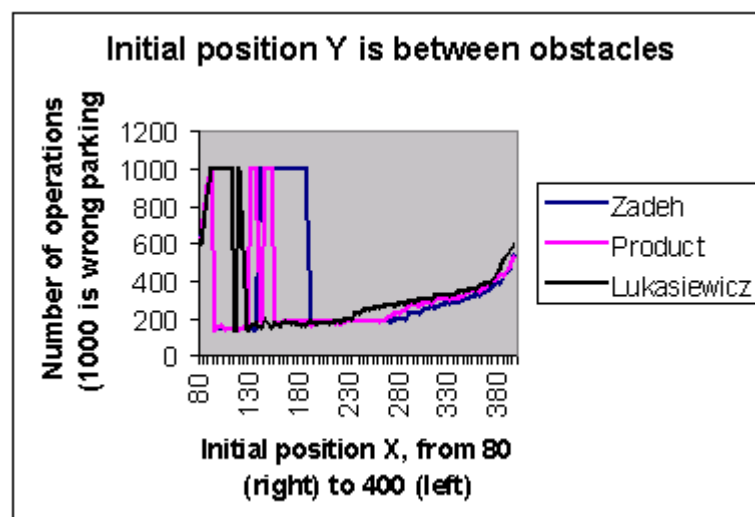


Figure 9. Initial position of the car at the high of the gap.

In this case, as it is shown in Figure 9, if the car is near the curb, the best logic is again Zadeh, but while is moving away from the edge (from 130 to 200), Lukasiewicz performs better.

In these two cases, when the car is far from the car (from 200 to 400), the performance of all the logics is quite similar.

Finally, in Figure 10 the results of applying different fuzzy logics when the car is in front of the car at different distances to the curb are presented.

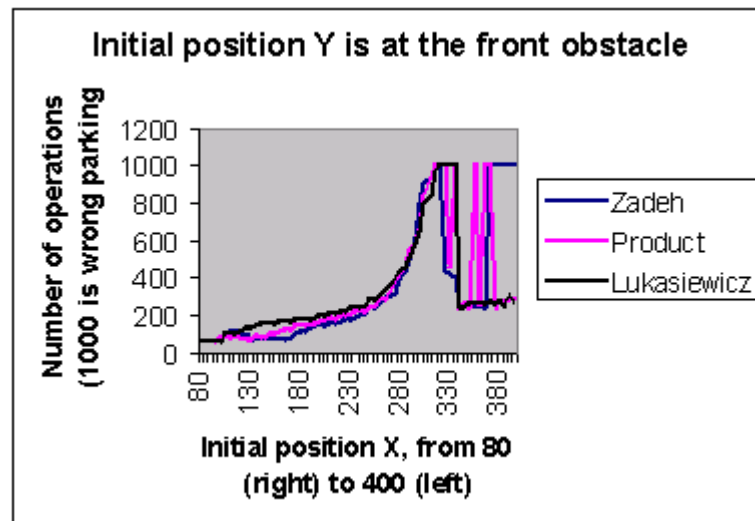


Figure 10. Initial position of the car in front of the gap.

When the car is in front of the gap (Figure 10), the three fuzzy logics perform well until the distance to the curb is around 280. In that case, the behavior gets worse for all of them, although it is slightly better for Lukasiewicz.

The results of applying those different fuzzy logics allow us to draw to some conclusions that are summarized in the following Table 5. The best logic is determined respect to the distance to the curb.

Table 5. Results of applying different fuzzy logics

|             | Behind | inFront | at_the_same_high |
|-------------|--------|---------|------------------|
| Zadeh       | near   | -       | near             |
| Łukasiewicz | medium | far     | medium           |
| Product     | -      | -       | -                |

## 5. Conclusions and future work

The use of fuzzy logic in this project has proved to be a simple, yet robust and efficient, decision system. The car in the simulation applet is able to park successfully from most of the initial situations.

The application of three different fuzzy logics allows us to show how the number of maneuvers that are needed to park the car depends on both the initial position and the chosen logic. For each initial position of the car, a specific fuzzy implementation of the operators gives the best results.

The implemented applet can be tested at <http://www.fdi.ucm.es/profesor/lgarmend/SC/aparca/index.html>.

## References

1. IMSE Centro Nacional de Microelectrónica. *Herramientas de CAD para Lógica Difusa. Xfuzzy 3.0*. <http://www.imse.cnm.es> (2003).
2. S.M. LaValle. *Planning algorithms*. Cambridge University Press (2006).
3. D. Montenegro, M. Santos and L. Garmendia. *7<sup>th</sup> IFAC Symposium on Advances in Control Education* (2006).
4. G. Pajares and M. Santos. *Inteligencia Artificial e Ingeniería del Conocimiento*, RA-MA (2005).
5. B. Schweizer A. and Sklar, *Probabilistic Metric Spaces*. North-Holland (1960).
6. E. Trillas and L. Valverde, *Approximate reasoning in expert systems*, North-Holland, 157-166 (1985).
7. L.A. Zadeh, *Information and Control*, **8**, 338-353, 1965.

[1] Bentalba, S., El Hajjaji, A. and Rachid, A., (1997) "Fuzzy Parking and Point Stabilization: Application Car Dynamics Model.", 5th IEEE Mediterranean , Cyprus.

[2] Bentalba, S., El Hajjaji, A. and Rachid, A., (1998) "Design fuzzy controller for vehicle dynamics" Proc. IEEE Int. Conf. on Advances in Vehicle Control and Safety, AVCS'98, Amiens

[3] Conde. L. Bento, Nunes, U., Mendes A. and Parent, M. (2003) "Path-Tracking Controller of a bi-steerable Cybernetic Car using Fuzzy Logic", Proceedings of ICAR, The 11th International Conference on Advanced Robotics, Coimbra, Portugal.

[4] Czogala, E., Pedrycz, W., (1981), "On Identification in fuzzy systems and its applications in control problems", Fuzzy Sets and Systems, 6, pp. 73-83.

[5] Dubois, D., Prade, H., (1980), "Fuzzy Sets and Systems. Theory and its Applications". Academic Press, New York.

[6] IMSE (2003). Centro Nacional de Microelectrónica. Herramientas de CAD para Lógica Difusa. Xfuzzy 3.0. Seville. <http://www.imse.cnm.es>.

[7] García Rosa, R., Pedro Lucio, T., Rosetti de Almeida, A., (1998)," Fuzzy driving strategies for cars in several traffic situations.", IFAC Symposium on Intelligent Autonomous Vehicles.

[8] Graham, B.P., Newell, R.B. (1989), "Fuzzy adaptive control of a first-order process". Fuzzy Sets and Systems, 31, pp. 47-65.

[9] Graham, B.P., Newell, R.B. (1991), "An adaptive fuzzy model-based controller". IJCAI, Fuzzy Logic and Fuzzy Control, 56-66, Sydney, Australia.

[10] Holve, R., Protzel, P., (1996) "Reverse Parking of a Model Car with Fuzzy Control.", Proceedings. of the 4th European Congress on Intelligent Techniques and Soft Computing - EUFIT-96, Aachen, pp. 2171-2175.

[11] Klir G. and Yuan, B. (1995). "Fuzzy Sets and Fuzzy Logic Theory and its Applications" Prentice Hall.

[12] Lewis, H.W. (1997), "The Foundations of Fuzzy Control", Springer.

[13] Nguyen, H.T. and Walker, E. A., (1996). "A first course in Fuzzy Logic". CRC Press.

- [14] Oh, S., Pedrycz, W., Park, B. (2002), "Hybrid identification of fuzzy rule-based models". Int. Journal of Intelligent Systems, vol. 17 (1), pp.77-103.
- [15] Pedrycz, W., (1991), "Fuzzy Dynamic Systems". IJCAI, Fuzzy Logic and Fuzzy Control, pp. 37-44, Sydney, Australia.
- [16] Pedrycz, W.: Modelling with fuzzy sets in fuzzy control. Fuzzy Days, 3-34, 1992.
- [17] Piegat, A., (2001). "Fuzzy Modeling and Control". Springer
- [18] Pradera, A., Trillas, E., Cubillo, S., (2000) "On modus ponens generating functions" Internat. J. Uncertain. Fuzziness Knowledge Based Systems 8, 1, pp. 7-19.
- [19] Sugeno, M., (1985), "An introductory survey of fuzzy control". Information Sciences, n. 36, pp. 59-83.
- [20] Schweizer B. and Sklar, A., (1960), "Probabilistic Metric Spaces". North-Holland, New York.
- [21] Trillas E. and Valverde, L. (1985), "On mode and implication in approximate reasoning", Approximate reasoning in expert systems, Eds. M. M. Gupta, North-Holland. pp. 157-166.
- [22] Trillas, E., Alsina, C. and Terricabras, J. M. (1995) "Introducción a la Lógica Borrosa". Editorial Ariel.
- [23] Trillas, E., and Valverde, L. (1985). "On mode and implication in approximate reasoning". Approximate reasoning in expert systems. Eds. M. M. Gupta. North-Holland. pp. 157-166.
- [24] Zadeh, L.A. (1965), "Fuzzy sets". Proceedings of the IEEE (Information and Control), Vol:8: pp. 338-353.

On Sun, May 15, 2016 at 4:05 PM Rogerio Cisi <[rogeriocisi@gmail.com](mailto:rogeriocisi@gmail.com)> wrote:  
| <http://www.fdi.ucm.es/profesor/lgarmend/SC/aparca/>