A Simple Neuro-Fuzzy Controller for Car-Like Robot Navigation Avoiding Obstacles

Iluminada Baturone and Andrés A. Gersnoviez

***Abstract*— This paper describes how the combination of neuro-fuzzy techniques with geometric analysis offers a good trade-off between purely heuristics and purely physical approaches when solving the problem of car-like robot navigation. The controller described, which follows a reactive technique, generates trajectories of near-minimal lengths when no obstacles are detected and, in presence of obstacles, generates minimum deviations from them. All these reference paths meet the kinematic constraints of car-like robots and take into account dynamic issues. Besides its efficiency, the proposed controller is very simple and linguistically interpretable. The whole controller has been designed and verified by using the CAD tools of the Xfuzzy environment.**

I. Introduction

A basic task to be performed by an autonomous robot is to navigate safely among possible obstacles towards a goal destination. A large number of methods for solving this motion planning problem have been reported in the literature. In particular, local methods are widely used for implementing real-time navigation thanks to their simplicity and suitability for sensor-based navigation through partially unknown environments. Several local methods based on neuro and fuzzy paradigms have been reported recently [1]-[8]. Many of them employ fuzzy rules extracted from heuristic knowledge [4]-[5]. They are usually simple but often require a tedious and unreliable trial-error adjustment which does not obtain optimal results. The other approach is to combine neuro and fuzzy techniques to learn the trajectories provided by an expert [6]-[8]. However, these trajectories are not usually as optimal as a geometrical analysis of the problem could provide and the resulting controllers are not usually simple and may lose linguistic meaning.

The coarse structure of the controller described in this paper has been designed by using heuristic knowledge. The fine structure of its constituent modules has been obtained by applying supervised learning with numerical data resulting from a geometric analysis of the problem (which considers the kinematic and dynamic constraints of the robot). As a result, the trajectories provided by our controller satisfy the nonholonomic constraints of car-like robots, which are that speed direction must always be tangent to the trajectory and that the robot curvature is upper bounded. These trajectories are of near-minimal lengths, when no obstacles are detected, and, in presence of obstacles, the deviation from these paths is as small as possible. The controller consists of several neuro-fuzzy modules which have been optimized to be very simple and maintain linguistic meaning.

The paper is organized as follows. Section II describes the navigation problem and the modules of the reactive controller employed to solve it. Its subsections summarize the geometrical considerations that should be taken into account to meet car-like robot constraints and the advantages of using hierarchical neuro-fuzzy modules to obtain a very simple controller that requires a very low computational cost. The whole controller has been designed and verified by using the CAD tools of Xfuzzy 3, an environment to design neuro-fuzzy controllers developed at IMSE [9].

II. The Navigation Algorithm

The configuration of a car-like robot can be given by the position of the back wheel axle midpoint with regards to a global coordinate system, (x, y); its orientation, ϕ; the curvature defined by the front wheels, γ; and its speed, v (Figure 1). In our case, no model about the environment is used for navigation but only the information provided by the robot sensors. In particular, we consider a 2-D laser placed at the front of the robot that performs a scan of up to 180 degrees. The laser identifies the points of possible obstacles by their distance, h, and sweeping angle, φ. The navigation problem consists of generating a collision-free trajectory from an initial configuration (x, y, ϕ, v, γ) to a goal one, which is $(0, 0, 180º, 0, 0)$ in the global coordinate system we have selected.

Since our objective is to obtain a low-cost and real-time

This work was supported in part by the Spanish CICYT Projects DPI2005-02293 and TEC2005-04359, and by the Projects TEP2006-375 and TIC2006-635 from the Andalusian regional Government. The second author is supported by the Spanish Ministry of Education and Science under the program F.P.U. for PhD. students.

I. Baturone and A. Gersnoviez are with the Instituto de Microelectrónica de Sevilla (IMSE-CNM-CSIC) and the Dept. de Electrónica y Electromagnetismo, Univ. de Sevilla, Seville, SPAIN (phone: +34-955-056-666; fax: +34-955-056-686; e-mail: {lumi, andres}@imse.cnm.es).

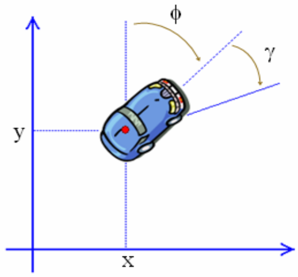


Figure 1. Parameters defining the robot configuration.

controller, only a small area near the robot is explored to detect obstacles; such area should be large enough to allow an avoidance maneuver. The analysis of a greater area would involve a higher computational cost and would increase unnecessarily the occurrence of avoidance maneuvers. The heuristic knowledge to design the coarse structure of our controller is the following:

- If there is an obstacle *very close* to the robot, the robot should stop to avoid collision.
- If there is an obstacle *close* to the robot, a maneuver to avoid it should be carried out.
- If the obstacles are *far* from the robot, there is no need to avoid them (by the moment) and the robot should navigate towards the goal configuration.

Many reactive controllers reported in the literature use fuzzy logic to deal with the concepts of ‘very close’, ‘close’ and ‘far’ and define membership functions for them that are based on heuristics as well as trial and error approach. Our proposal is to consider the dynamic and kinematic constraints of the robot so as to better define them.

A. *Very close obstacles*

Taking into account that the robot curvature is limited to γ_{max} and that 2m at both sides of the robot should be free of obstacles for safety purposes (the robot considered in our analysis has a width of 1m and the maximum speed considered is 1m/s), a geometric analysis of the problem gives that, approximately, those obstacles which enter the shadowed area in Figure 2 are ‘very close’ and, hence, they

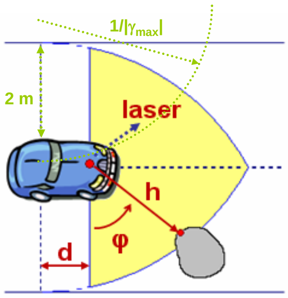


Figure 2. Area of unavoidable obstacles.

are unavoidable. The equations that define this area are as follows:

$$
\begin{aligned}
&(x_o - x + \frac{\cos\phi}{|\gamma_{max}|})^2 + (y_o - y - \frac{\sin\phi}{|\gamma_{max}|})^2 < (2 + \frac{1}{|\gamma_{max}|})^2 \\
&\text{if } \phi \in [0^\circ, 90^\circ] \\
&(x_o - x - \frac{\cos\phi}{|\gamma_{max}|})^2 + (y_o - y + \frac{\sin\phi}{|\gamma_{max}|})^2 < (2 + \frac{1}{|\gamma_{max}|})^2 \\
&\text{if } \phi \in [90^\circ, 180^\circ]
\end{aligned} \tag{1}
$$

where (x_o, y_o) are the coordinates of the obstacle with regards to the global reference system.

If the coordinates of the obstacle are referred to a system attached to the robot whose origin coordinates are placed at the back wheel axle midpoint, the area of ‘very close’ or unavoidable obstacles is defined by:

$$
h^2 + 2h[d \cdot |\sin\varphi| + \frac{|\cos\varphi|}{|\gamma_{max}|})^2 < 4(1 + \frac{1}{|\gamma_{max}|}) - d^2 \tag{2}
$$

where *d* is the distance of the laser to the back wheel axle.

Equation (2) is preferred to (1) not only for its simplicity but also for using the variables *h* and φ, which are provided directly by the laser (the variables x_o and y_o in (1) require processing the laser data). The shadowed area in Figure 3a illustrates the points satisfying (2).

A fuzzy classifier with two inputs, *h* and φ, and one output (Figure 4a) has been designed and tuned with the numerical data corresponding to (2). The environment Xfuzzy with their CAD tools *xfdm*, *xfsp*, and *xfsl* (for, respectively, extracting rule bases from numerical data, simplifying and tuning them) has been employed [10]. The system obtained contains the following 5 rules:

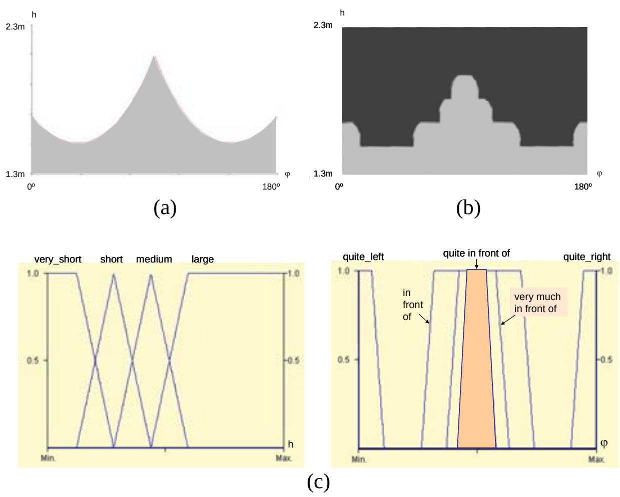
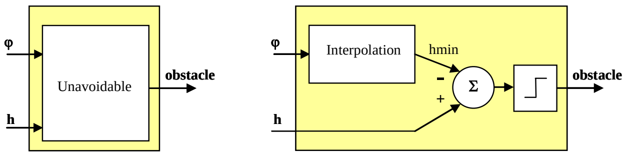


Figure 3. (a) Area of ‘very close’ obstacles according to (2). (b) Result provided by the monolithic system. (c) Fuzzy concepts employed.



(a) (b)

Figure 4. Neuro-fuzzy system to detect very close obstacles: (a) Monolithic structure. (b) Hierarchical structure.

1.- If h is *very short* then *obstacle* is *very close*.
2.- If h is *short* and φ is *quite left*, *quite right*, or *in front of* the robot then *obstacle* is *very close*.
3.- If h is *medium* and φ is *very much in front of* the robot then *obstacle* is *very close*.
4.- If h is *large* and φ is *quite in front of* the robot then *obstacle* is *very close*.
5.- In other case, *obstacle* is not *very close*.

The output provided by this neuro-fuzzy system is shown in Figure 3b (dark area represents ‘not very close’ obstacles), and the fuzzy concepts employed for the h and φ inputs are illustrated in Figure 3c.

Another approach to design this neuro-fuzzy system is to employ the hierarchical structure shown in Figure 4b. A fuzzy rule base, named ‘interpolation’ in Figure 4b, interpolates, for each angle φ, the minimum value of h, h_{min}, that an obstacle should have to be considered as ‘not very close’. This rule base has been trained by the numerical data (φ, h_{min}) that verify (2) when condition ‘<’ is substituted by ‘=’. Again using the CAD tools of Xfuzzy, the neuro-fuzzy system obtained contains the following rules:

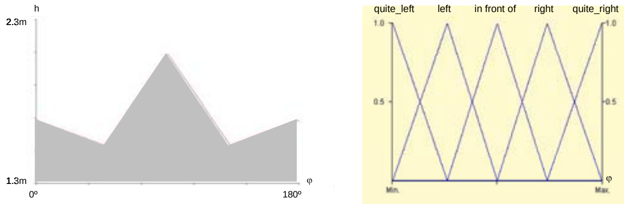
1.- If φ is *quite left* or *right* then h_{min} is 1.7m.
2.- If φ is *right or left* then h_{min} is 1.5m.
3.- If φ is *in front of* the robot then h_{min} is 2.1m.

The shadowed area in Figure 5a illustrates the points that this system classifies as unavoidable. The fuzzy concepts employed in the rules are shown in Figure 5b.

Performance of monolithic and hierarchical neuro-fuzzy systems is similar, but the hierarchical one has been selected for being slightly simpler.

B. Close obstacles

Let us now consider the obstacles that are ‘close’ to the robot and should be avoided. Taking into account the current robot curvature, γ, that 2m at both sides and ahead in the driving direction should be free of obstacles for safety purposes (considering the robot size and an speed of 1m/s), those obstacles which approximately enter the dark shadowed area in Figure 6a are ‘close’ and, hence, they should be avoided (the robot is assuming to be turning to



(a) (b)

Figure 5. (a) Result provided by the hierarchical neuro-fuzzy system. (b) Fuzzy concepts employed.

the right in this figure). Considering a relative coordinate system attached to the robot, the equations that define this area are as follows:

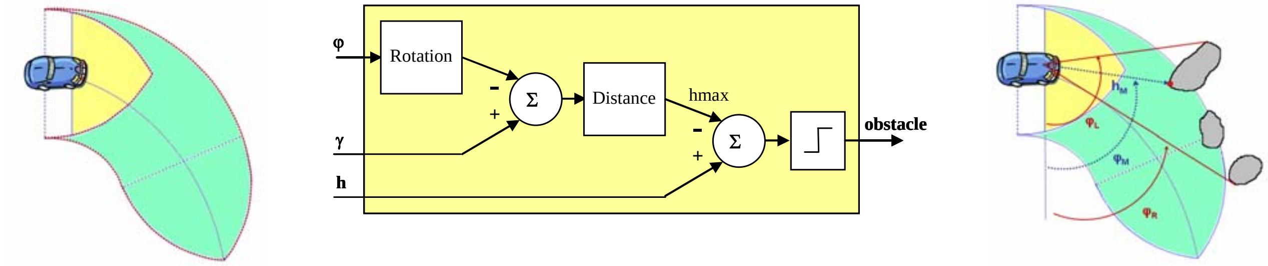
$$\left(2-\frac{1}{|\gamma|}\right)^2 < x_r^{\,2} + \left(y_r + \frac{1}{\gamma}\right)^2 < \left(2+\frac{1}{|\gamma|}\right)^2 \text{ and } x_R < f(y_R) \qquad (3)$$

where (x_r, y_r) are the coordinates of the obstacle with regards to the relative reference system and (x_R, y_R) are the coordinates of the obstacle with regards to a reference system displaced 2m in the driving direction:

$$\begin{aligned} &x_r = d + h\cdot\sin\varphi \qquad y_r = -h\cdot\cos\varphi \\ &x_R = \left(x_r - \frac{\sin 2\gamma}{\gamma}\right)\cos 2\gamma - \left(y_r + \frac{1-\cos 2\gamma}{\gamma}\right)\sin 2\gamma \quad \text{and} \\ &y_R = \left(x_r - \frac{\sin 2\gamma}{\gamma}\right)\sin 2\gamma + \left(y_r + \frac{1-\cos 2\gamma}{\gamma}\right)\cos 2\gamma \end{aligned} \qquad (4)$$

The function f() in (3) corresponds to expressions (1) or (2) but replacing ‘<’ by ‘=’.

As in the case of ‘very close’ obstacles, we can follow two approaches to define a fuzzy classifier that decides if an obstacle is ‘close’ or not: a monolithic system or a hierarchical one. A monolithic system would have now three inputs, h, φ, and γ, and one output. Hence, the difference between the monolithic and hierarchical approaches is now higher than in the case of very close obstacles, where the monolithic system had two inputs. As a consequence, the hierarchical approach has been selected directly. In particular, we have explored the highest (and, hence, simplest) hierarchical scheme, shown in Figure 6b, where rule bases with only one input are employed. This scheme maintains the linguistic meaning that a fuzzy system should have. The fuzzy rule base named ‘rotation’ in Figure 6b allows estimating approximately the position with regards to the robot that a possible obstacle would have in the future. The rule base named ‘distance’ interpolates the maximum value of h, h_{max}, that an obstacle should have to be considered as ‘close’. These rule bases have been trained by the numerical data (φ, γ, h_{max}) that verify (3) when condition ‘<’ is substituted by ‘=’. Using the CAD tools of Xfuzzy, two rules have been extracted for



(a) (b) (c)

Figure 6. (a) Area of 'close' obstacles. (b) Hierarchical system to classify obstacles as 'close' or not. (c) Parameters to calculate the new curvature.

the rule base 'rotation' and three ones for the rule base 'distance'. The latter ones can be expressed linguistically as follows:

1.- If the obstacle will be *in front of* the robot in future configurations then h_{max} is 4.0m.
2.- If the obstacle will be *on the right or left of* the robot in future configurations then h_{max} is 2.0m.
3.- If the obstacle will be *quite on the right or left of* the robot in future configurations then h_{max} is 1.6m.

According to (3) and (4), the surface in Figure 7a illustrates the maximum values of *h*, h_{max}, that have the obstacles considered as 'close' depending on its angular positions, φ, and curvature of the robot, γ. The surface in Figure 7b illustrates how the hierarchical fuzzy system designed approximates the already approximated geometrical analysis.

C. *Obstacle avoidance*

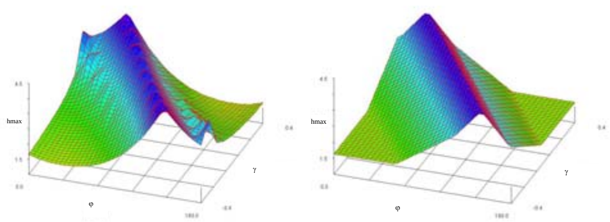
Once a 'close' obstacle is detected, the controller evaluates the angular aperture (φ_L - φ_R) seen by the laser from the obstacle (Figure 6c) and determines the sign of the curvature to avoid the obstacle by applying the following heuristic rules:

1.- If the obstacle is *on the right* then turn to the left (and vice versa).
2.- If the obstacle is *in front of* the robot then:
a) turn to the same side as in the previous control cycle (provided the obstacle was already detected).
b) If this is the first time the obstacle is detected, apply the sign as no obstacle was detected.

The minimum magnitude of the curvature to avoid the obstacle is determined by the closest point of the obstacle (h_M and φ_M in Figure 6c). Maintaining a free 2-m width corridor and, taking into account that a reference curvature is not adopted instantaneously by the robot but has a delay, a geometrical analysis provides that the minimum value of the curvature is given by the following formula:

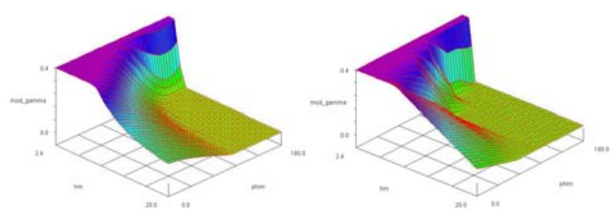
$$\gamma = \begin{cases} \dfrac{8 + 2h_M \cdot \cos\varphi_M}{d^2 + h_M{}^2 + 2h_M \cdot d \cdot \sin\varphi_M - 16} & \textit{turning to right} \\ \dfrac{8 - 2h_M \cdot \cos\varphi_M}{d^2 + h_M{}^2 + 2h_M \cdot d \cdot \sin\varphi_M - 16} & \textit{turning to left} \end{cases} \quad (5)$$

In order to approximate the already approximated analysis in (5) by a fuzzy system, a monolithic and a hierarchical approach have been again explored with the aid of Xfuzzy. Exploiting the symmetry of the problem (γ=f(φ_M,h_M), when turning to right and γ=f(180-φ_M,h_M), when turning to left), the turning to right problem has been only analyzed. Figure 8a shows these γ values (when turning to right) versus φ_M and h_M. In this case, the monolithic system has been selected because its complexity is somewhat lower (5 instead of the 9 rules extracted in the hierarchical system) achieving a better approximation. The obtained system, which decides how much turning to right



(a) (b)

Figure 7. (a) Maximum distance of obstacles considering as 'close' according to (3). (b) Result provided by the hierarchical system.



(a) (b)

Figure 8. (a) Curvature to avoid obstacles according to (5). (b) Result provided by the neuro-fuzzy system.

(once turning to right has been decided), forces the robot to turn more as more dangerous the obstacle is for the way the robot is going to take immediately. The more dangerous the obstacle is, the more the robot will turn to right. This can be understood from the rules this module applies (imagine the robot is going ahead or to its right):

1.- If h_M is *very short* then turn right at maximum.
2.- If h_M is *short* and φ_M is not *quite on the left* then turn right at maximum.
3.- If h_M is *medium* and φ_M is *in front of or on the right* then turn right at maximum.
4.- If h_M is *large* and φ_M is *right* then turn right at maximum.
5.- In other case, keep straight ahead.

Since these rules employ fuzzy concepts, the γ values provided by this system do not switch abruptly but vary smoothly between maximum turning to right and zero, as shown in Figure 8b.

D. *Navigation toward the goal*

If there are no obstacles, the robot should navigate toward the goal (0, 0, 180º, 0, 0) by the shortest path. The shortest paths for a car-like vehicle consist of a finite sequence of two elementary components: arcs of circle (with minimum turning radius) and straight line segments, as was proved by Dubins [11]. Analyzing the shortest paths geometrically, it can be found that there is an angle, α, which defines the orientation of the straight segment which contains the (x, y) point of the current robot configuration and is tangent to the arc of circle that defines the end of the path (Figure 9). Depending on the difference between this angle and the current robot orientation the robot will turn to right or left or will not turn (if the difference is zero). The value of the angle α depends on x and y as follows:

$$\alpha = \begin{cases} \mathrm{sign}(x)\cdot \mathrm{ar\cos}(\psi) & \text{if} \quad (|x| - R)^2 + y^2 > R^2 \\ \mathrm{sign}(x)\cdot \mathrm{ar\cos}\left(1 - \dfrac{|x|}{R}\right) & \text{if} \quad (|x| - R)^2 + y^2 \le R^2 \end{cases} \tag{6}$$

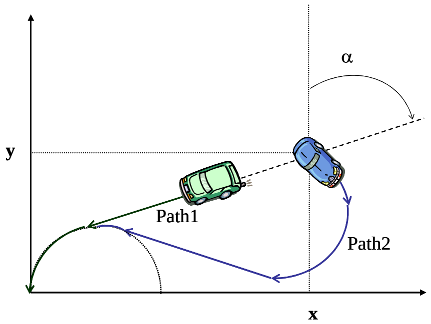


Figure 9. Minimal-length paths of car-like robots.

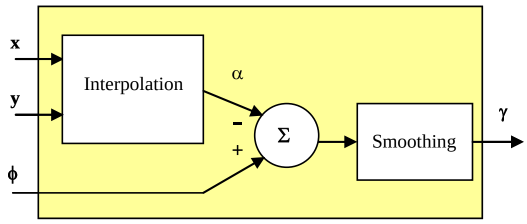


Figure 10. Hierarchical system to navigate without obstacles.

where R is the minimum turning radius and the value of ψ is given by:

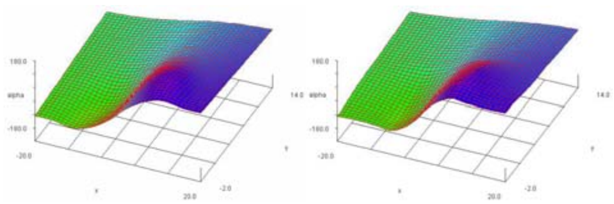
$$\psi = \frac{R(R - |x|) + y\sqrt{x^2 + y^2 - 2R|x|}}{x^2 + y^2 - 2R|x| + R^2} \tag{7}$$

A monolithic or a hierarchical fuzzy system can be employed to approximate this behavior, as in the previous steps. In this case, the hierarchical approach has been directly explored since the monolithic system is more complex for having three inputs (x, y, ϕ). The hierarchical scheme selected is shown in Figure 10. As in the previous cases, it consists of two rule bases connected in series. The first one (with 9 rules) provides approximately the value of the angle α, depending on the input variables x and y. Figure 11 illustrates how this module approximates the already approximated analysis summarized in (6) and (7). The second rule base (with 2 rules, exploiting symmetry) provides the curvature value depending on the difference between ϕ and α. These two rules are the following:

1.- If $(\phi - \alpha)$ is *somewhat smaller than 180º* then turn right a bit.
2.- If $(\phi - \alpha)$ is *much smaller than 180º* then turn right at maximum.

III. Results

The whole neuro-fuzzy-based controller has been described with the tool *xfedit* of Xfuzzy 3, which allows



(a) (b)

Figure 11. (a) Alpha value according to (6) and (7). (b) Result provided by the neuro-fuzzy system.

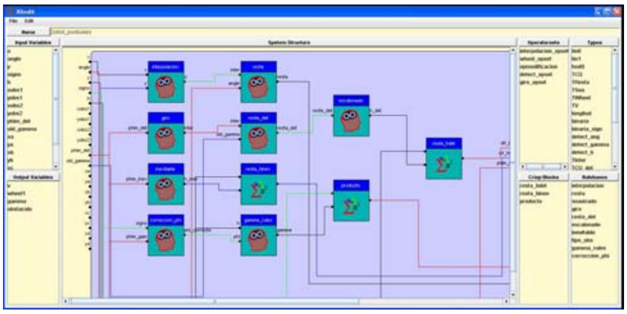


Figure 12. Description of the whole controller with Xfuzzy.

connecting fuzzy and non fuzzy rule bases as well as arithmetic modules. Figure 12 shows a capture of the main window of this tool illustrating part of this controller.

Simulations have been carried out with the tool *xfsim*. It simulates the controller working in a closed loop with a model of the robot. The robot model (which contains the models of its sensors) provides the new configuration of the robot and the new information given by the laser. This model is introduced in *xfsim* as a Java class.

Figure 13 shows several examples of how the robot is controlled to reach the goal configuration (x, y, ϕ, γ, v) = (0, 0, 180º, 0, 0) without colliding with obstacles. As illustrated, if no obstacles are detected, the robot goes to the goal by a near-minimal length path made up of arcs of circles with minimum turning radius and straight line segments, and, if obstacles are close, they are avoided by minimum deviations from the quasi-optimal path.

IV. Conclusion

A very simple controller to allow safe navigation of a car-like robot among possible obstacles has been described. The use of numerical data obtained from an approximated geometrical analysis of the problem has allowed this controller to provide quasi-optimal reference paths that meet the kinematic and dynamic constraints of the car-like robot considered. The CAD tools of the Xfuzzy environment have been employed to automate all the process of extracting rules from numerical data and heuristic knowledge, simplify, adjust them, and verify the whole system. The resulting controller contains 26 rules which are linguistically interpretable so that its performance can be understood and tested by any human.

References

[1] T. Fukuda, N. Kubota, "An intelligent robotic system based on a fuzzy approach", *Proc. of the IEEE*, vol.87, no. 9, pp. 1448-1470, Sept. 1999.

[2] D. Driankov, A. Saffiotti, eds., *Fuzzy logic techniques for autonomous vehicle navigation*, Springer-Physica Verlag, DE, 2001.

[3] F. Cuesta and A. Ollero, *Intelligent mobile robot navigation*, Springer Tracts in Advanced Robotics, vol. 16, 2005.

[4] B. Kimiaghalam, A. Homaifar, B. Suttikulvet, B. Sayyarrodsari, "A multi-layered multi fuzzy inference systems for autonomous robot navigation and obstacle avoidance", *Proc. IEEE Int. Conf. on Fuzzy Systems*, pp. 340-343, Dec. 2001.

[5] N. Zhang, D. Beetner, D. C. Wunsch II, B. Hemmelman, A. Hasan, "An embedded real-time neuro-fuzzy controller for mobile robot navigation", *Proc. IEEE Int. Conf. on Fuzzy Systems*, pp. 319-324, 2005.

[6] M. Sugeno, M. Nishida, "Fuzzy control of a model car", *Fuzzy Sets and Systems*, vol. 16, pp. 103-113, 1985.

[7] P. Rusu, E. M. Petriu, T. E. Whalen, A. Cornell, H. J. W. Spoelder, "Behavior-based neuro-fuzzy controller for mobile robot navigation", *IEEE Trans. on Instr. and Meas.*, vol. 52, no. 4, pp. 1335-1340, Aug. 2003.

[8] X. Wang, S. X. Yang, "A neuro-fuzzy approach to obstacle avoidance for a nonholonomic mobile robot", *Proc. IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, pp. 29-34, 2003.

[9] Xfuzzy home page: http://www.imse.cnm.es/Xfuzzy.

[10] I. Baturone, F. J. Moreno-Velo, A. A. Gersnoviez, "A CAD approach to simplify fuzzy system description", *Proc. IEEE Int. Conf. on Fuzzy Systems*, Vancouver, 2006.

[11] L.E. Dubins, "On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents", *American Journal Math.*, vol. 79, pp. 497-516, 1957.

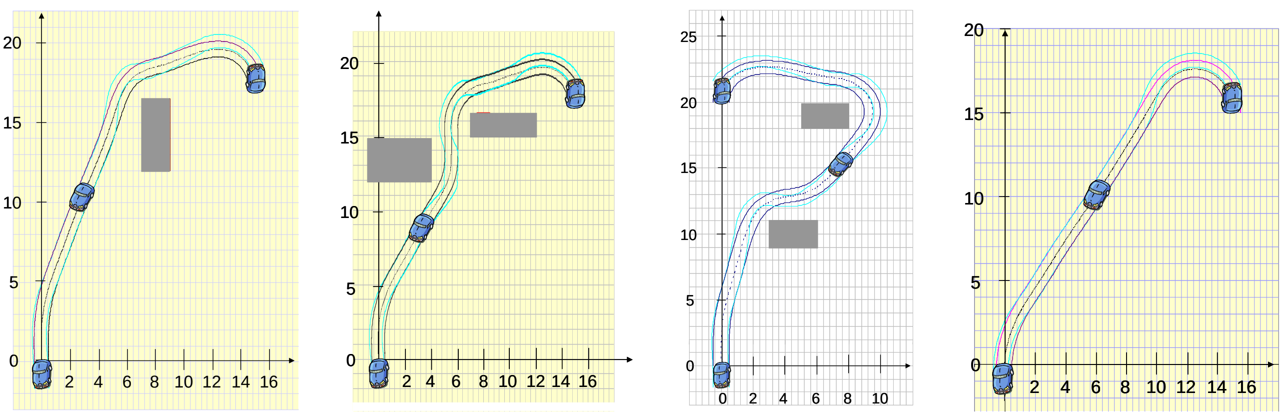


Figure 13. Simulation results obtained with Xfuzzy.