PERGAMON

Engineering Applications of Artificial Intelligence 14 (2001) 419–434

Engineering Applications of
ARTIFICIAL INTELLIGENCE

www.elsevier.com/locate/engappai

Parallel and diagonal parking in nonholonomic autonomous vehicles

F. Gómez-Bravo, F. Cuesta*, A. Ollero

Departamento Ingeniería de Sistemas y Automática, Escuela Superior de Ingenieros, Universidad de Sevilla,
Camino de los Descubrimientos, E-41092 Seville, Spain

Received 1 May 2000; accepted 1 December 2000

Abstract

This paper considers the problem of parallel and diagonal parking in wheeled vehicles. A method to plan in real-time a set of collision-free manoeuvres is presented. Artificial intelligent techniques, namely fuzzy logic, play an important role in the practical application of the method. Thus, a fuzzy system is used to select the most suitable manoeuvre from the solution set according with the environment, dealing with optimality, path tracking performance and collision avoidance trade-off. This technique has been implemented in a fuzzy behaviour-based control architecture combining planning and reactivity. The efficiency of the proposed method is demonstrated using the nonholonomic mobile robot ROMEO-3R designed and built at the University of Seville. © 2002 Published by Elsevier Science Ltd.

Keywords: Nonholonomic mobile robot; Manoeuvre planning; Fuzzy systems; Fuzzy control; Hybrid control

1. Introduction

Path-planning in nonholonomic systems has been studied by many authors. The parking problem of a nonholonomic vehicle consist of finding a path that connects the initial configuration to the final configuration and satisfies the existing nonholonomic constraints.

Numerous authors have presented manoeuvres solutions (Murray and Sastry, 1990), and collision-free manoeuvres (Sekhavat, 1997) in nonholonomic systems. Manoeuvres for parallel and diagonal parking are particular cases of nonholonomic motion.

Some approaches allow parallel parking using successive manoeuvres (Paromtchik and Laugier, 1996). The method presented in Laumont et al. (1994) consists of two steps: first, compute a collision-free path without considering nonholonomic constraints, or even geometric constraints; and second, compute a trajectory satisfying these constraints, closed enough to the one obtained in the first step, and verifying a noncollision algorithm test (Laumont et al., 1994). However, the solution and the computational time depends hardly on the starting point and the environment.

Other recursive methods (Paromtchik and Laugier, 1996), generate several sequential manoeuvres using sinusoidal paths in order to converge from a starting point to a goal point. For each manoeuvre, the path is tested against collisions. Nevertheless, if a possible collision is detected the vehicle has to move to another starting point, and to repeat the whole process from the new point. In Paromtchik et al. (1998), an off-line computation stored in a look up table is used in order to obtain a convenient start location. However, this method also generates several sequential manoeuvres even where direct motion is possible (see Fig. 1).

Several authors propose manoeuvres composed of circles and straight line segments (Neff Patten et al., 1994; Jiang and Seneviratne, 1999). Nevertheless they only deal with a constant value of the curvature. Then, given a parking place a unique collision-free manoeuvre is obtained.

This article, is focused on the design of direct motion into the parking instead of a sequence of manoeuvres. Thus, the parallel parking place should be wide enough (depending on the bound of the curvature). If it is not, one of the mentioned recursive methods should be applied. However, the proposed method is not as

*Corresponding author. Fax: +349-5448-7340.
E-mail addresses: fgbravo@cartuja.us.es (F. Gómez-Bravo), fede@cartuja.us.es (F. Cuesta), aollero@cartuja.us.es (A. Ollero).

0952-1976/01/$ - see front matter © 2002 Published by Elsevier Science Ltd.
PII: S0952-1976(01)00004-5

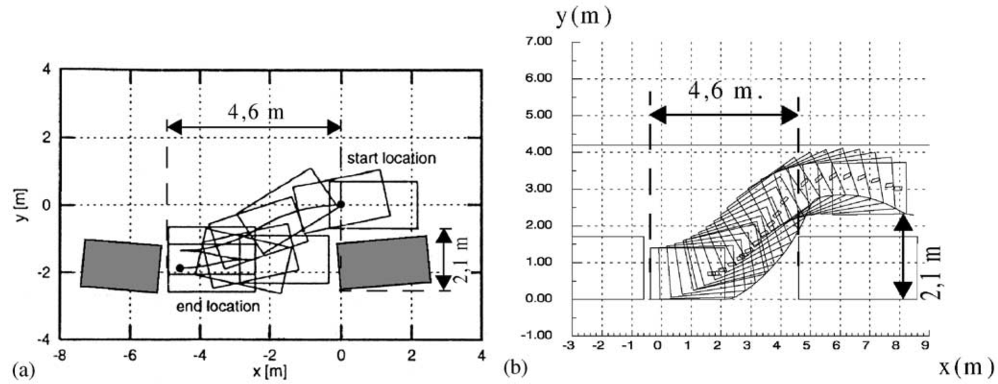


Fig. 1. (a) Multiple (five) manoeuvres from Paromtchik and Laugier (1996); (b) direct motion into the same parking place by means of the proposed approach.

restrictive as it could seem. Thus, it is able to compute a direct motion into parking lots where other methods, like the one proposed by Paromtchik and Laugier (1996), requires multiple manoeuvres (see Fig. 1).

In the proposed method, geometric constraints are directly considered to design a restricted manoeuvre, resulting in a collision-free path. Therefore, it is not necessary to implement a recursive algorithm to detect a collision. Even more, from the collision criterion, by means of a closed expression, the set of starting locations where a collision-free solution exists can be obtained. This task can be done by an on-line process. This characteristic is a significant advantage for on-line control purposes since, previously to stop, the robot can decide if the parking place is wide enough to park into and to obtain the set of points where the vehicle may finish the forward movement in order to avoid the further collisions.

The geometric constraints are satisfied using heuristic considerations which are easily understood by considering them for a vehicle leaving the parking place. Then, first of all a manoeuvre is planned for a virtual parked vehicle which would perform the manoeuvre to arrive to the initial configuration. After that, the actual manoeuvre is designed tracking the virtual manoeuvre in reverse.

On the other hand, studies of optimal parallel parking manoeuvres has been also considered taking into account shorter path generation (Jacobs, 1990; Neff Patten et al., 1994). Moreover, these techniques put the stress on parallel parking, and they do not deal with diagonal parking generation. In the proposed method, a new type of optimal manoeuvre is proposed (by taking into account collision with the opposite wall or parked vehicles line). Moreover, this article deals with both parallel and diagonal parking.

Artificial intelligent techniques, namely fuzzy logic, play an important role in the practical application of the method. Thus, a fuzzy system is used to select a manoeuvre from the solution set according with the environment, dealing with optimality, path tracking performance and collision avoidance trade-off.

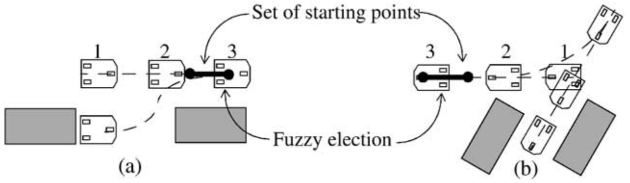


Fig. 2. (a) Parallel parking; (b) diagonal parking.

This article concentrates in a practical approach for parking of an intelligent autonomous vehicle and proposes the implementation of a fuzzy behaviour-based control architecture combining reactivity and planning. For this aim, an on-line path planning method is needed in order to avoid to stop the vehicle while computing the manoeuvre.

Thus, the vehicle is driven reactively, parallel to a wall (line of vehicles), looking for a parking place (step 1 in Fig. 2). Once the parking lot has been found, a set of starting points is calculated on-line (step 2 in Fig. 2). Then fuzzy logic is used to decide the best starting point taking into account the environment where the vehicle has to park. When the vehicle is stopped, a simple collision-free manoeuvre is generated: a direct motion into the parking, for parallel parking (step 3 in Fig. 2a); or a reorientation manoeuvre and direct motion into the parking, for diagonal parking (step 3 in Fig. 2b). Then, the manoeuvre is executed.

The fundamentals of the method to plan the manoeuvres are presented in Sections 2 and 3. The method is applied to parallel and diagonal parking in Section 4. In Section 5, the fuzzy system used to select the parking manoeuvre is presented in detail. In Section 6, the application to a real autonomous vehicle, ROMEO-3R, is presented. Section 7 is for the conclusions.

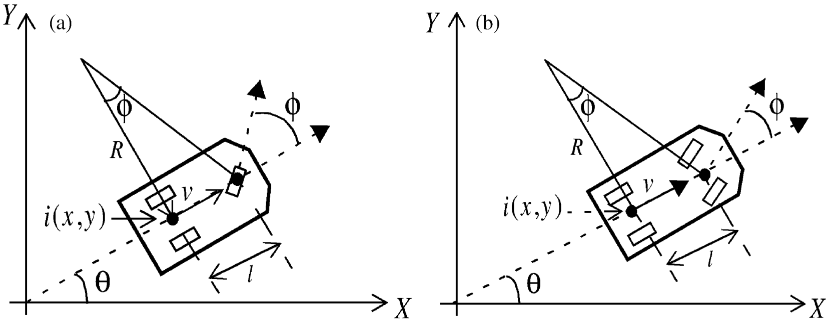


Fig. 3. (a) Cart like vehicle, (b) car like vehicle.

2. Manoeuvre design

Nonholonomic systems are characterized by nonintegrable differential expressions such as

$$\sum_{i=l}^{n} f_{ij}(q_1, q_2 \ldots q_n, t)\dot{q}_i = 0, \quad j = 1, 2 \ldots m, \tag{1}$$

where $\dot{q}_i$ represents the n generalized coordinate (state variable), m is the number of equations defining the nonholonomic constrains, $\dot{q}_i$ represents the generalized speed and f_{ij} are nonlinear functions of q_i and time t. The velocity vector $\dot{q}_i$ verifying these equations spans a $(n - m)$ subspace in the tangent space (Δ) (Latombe, 1991).

The idea of designing closed paths for some variables (independent variables), in order to change the value of the other variables (dependent variables) has been proposed in the literature (Mukherjee and Anderson, 1994; Vafa and Dubowsky, 1987). This is what will be called *restricted manoeuvre*.

Manoeuvres for parallel and diagonal parking are particular cases in which it is required that the values of some variables in the initial and final states should be the same (see Fig. 2). Thus, they can be solved using this type of manoeuvre. However, this problem is not a trivial task.

In order to give a formal description of restricted manoeuvres, concepts from differential geometry are used (Laumont et al., 1994) (see Appendix A for details). The general manoeuvre generation is based on the selection of a sequence of n vectors V_i, belonging to Δ, and values of s_i in such a way that the manoeuvre

$$\Phi = p_0 e^{s_1 V_1} \ldots e^{s_n V_n} \tag{2}$$

provides the desired increment in the dependent variable.

The solution proposed in this article is based on the use of vectors that make the velocity of the dependent variable to accomplish $\dot{q}_{\mathrm{d}} = 0$. Such vectors are noted as *vector* V_p, and are characterized by substituting this condition in Eq. (1) (Gómez-Bravo and Ollero, 1995). This method can be applied to path planning in complex systems like free floating manipulator or a tractor trailer system (Gómez-Bravo, 2000). For each system, a suitable closed path has to be designed by studying a particular application of vector V_p.

3. Manoeuvre in wheeled vehicles

Cars and Carts are wheeled vehicles (see Fig. 3) which has been used in many autonomous vehicle applications. The kinematic of these types of vehicles (Laumont et al., 1994) can be expressed in the form

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ v(t)\rho(t) \end{bmatrix}, \tag{3}$$

where (x, y) are the coordinates of the reference point i, θ is the vehicle's heading, $v(t)$ is the linear velocity of point i, and $\rho(t)$ is the curvature of the path described by point i.

For the kinematic model (3), the following nonholonomic constraint exists:

$$\dot{x}\sin\theta - \dot{y}\cos\theta = 0. \tag{4}$$

The curvature radius for any path can be written as

$$R(t) = \frac{1}{\rho(t)} = \frac{l}{\tan\phi}, \tag{5}$$

where l is the distance between the front and back wheels, and ϕ (namely the steering angle) is the angle defined by the main axis of the vehicle and the velocity vector of the front wheel, for cart like vehicles (see Fig. 3a), or the central front point, for car like vehicles (see Fig. 3b). The value of $R(t)$ is usually bounded by $R_{\min}$, the minimum curvature radius.

In this paper two different *restricted manoeuvres* are presented. The first one, is designed to change the value of the cartesian variable y; while the objective of the second one is to change the value of θ. For this aim, a reference frame with the origin in the goal point, and parallel oriented to the initial vehicle configuration is chosen.

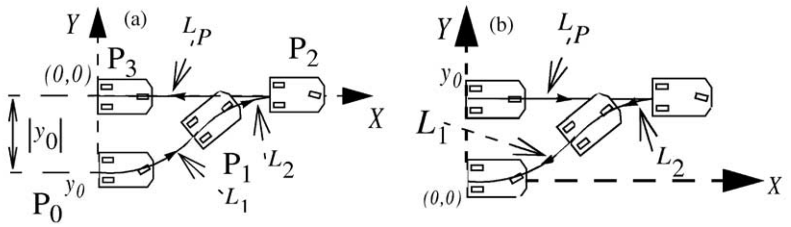


Fig. 4. (a) Restricted manoeuvre with positive change of variable y; (b) negative.

3.1. Restricted manoeuvre with y as dependent variable

In this manoeuvre the initial and final value of the variables θ and x are the same, while the value of the variable y is changed. Due to the election of the reference frame, this manoeuvre consists of moving the vehicle from the configuration (0, y_0, 0) to (0, 0, 0) (see Fig. 4).

In order to design the manoeuvre, three vectors are chosen: V_P, V_1, and V_2. The vector V_P is chosen as (see Appendix B)

$$V_P = [v(t)\,0\,0]^{\mathrm{T}}. \tag{6}$$

On the other hand, vectors V_1 and V_2 are chosen as

$$V_1 = \begin{bmatrix} v(t)\cos(\theta) \\ v(t)\sin(\theta) \\ \dfrac{v(t)}{R_1} \end{bmatrix}, \quad V_2 = \begin{bmatrix} v(t)\cos(\theta) \\ v(t)\sin(\theta) \\ \dfrac{v(t)}{R_2} \end{bmatrix}, \tag{7}$$

where $R_1 = \mathrm{cte}\rangle|R_{\min}|$ and $R_2 = \mathrm{cte}\langle -|R_{\min}|$.

These vectors satisfy (4). It could be shown (see Appendix B) that a positive change in the variable y can be obtained from a restricted manoeuvre Φ^{+y} defined as

$$\Phi^{+y} = P_0 \mathrm{e}^{s_1 V_1} \mathrm{e}^{s_2 V_2} \mathrm{e}^{s_p V_p} \tag{8}$$

where the values of the parameters are

$$s_1 = R_1\theta_1, \quad s_2 = -R_2\theta_1, \quad s_p = (R_2 - R_1)\sin\theta_1$$

and

$$\theta_1 = \operatorname{acos}\left(1 - \frac{|y_0|}{R_1 - R_2}\right). \tag{9}$$

This manoeuvre is composed of two segments of arcs of circumference (see Fig. 4a), $L_1(P_0 \rightarrow P_1)$ and $L_2 \times (P_1 \rightarrow P_2)$, generated by V_1 and V_2, respectively, and the straight segment $L_P(P_2 \rightarrow P_3)$. The parameters s_1, s_2 and s_p represent the length of these segments.

In a similar way, a restricted manoeuvre can be defined for a negative increment on y (see Fig. 4b) by using the same vector fields but in an inverse sequence, that is

$$\Phi^{-y} = P_0 \mathrm{e}^{-s_p V_p} \mathrm{e}^{-s_2 V_2} \mathrm{e}^{-s_1 V_1}, \tag{10}$$

where s_p, s_2 and s_1 are defined in (9).

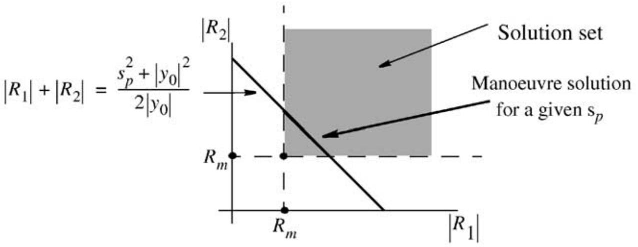


Fig. 5. Manoeuvre solution set.

Summarizing, for a desired Δy, infinite manoeuvres can be found. Each one is determined by a pair of values of R_1 and R_2. Thus, the positive quadrant defined by the values of $|R_2|$ and $|R_1|$ which accomplish $|R_2| > R_{\min}$ and $|R_1| > R_{\min}$ can be considered as the manoeuvre solution set (see Fig. 5). Each point belonging to this area represents a feasible manoeuvre and a corresponding value for s_1, s_2, and s_p.

As s_p represents the vehicle forward displacement, from (9), the relation between $|R_2|$, $|R_1|$ and the forward displacement can be obtained as (note that $R_1 - R_2 = |R_1| + |R_2|$)

$$|R_1| + |R_2| = \frac{s_p^2 + |y_0|^2}{2|y_0|}. \tag{11}$$

Then, for a given s_p, all the feasible manoeuvres are determined by the values of $|R_2|$ and $|R_1|$ lying on the straight line (11) (see Fig. 5).

3.2. Restricted manoeuvre with θ as dependent variable

This problem involves changing the vehicle's orientation while the cartesian position of the reference point does not change. Due to the chosen reference frame, this manoeuvre consist of moving the vehicle from the configuration $(0, 0, 0)$ to $(0, 0, \theta_f)$ (see Fig. 6).

The vector V_P, is chosen as

$$V_P = [v(t)\cos(\theta), v(t)\sin(\theta), 0]^{\mathrm{T}}. \tag{12}$$

Using the same ideas applied in the previous subsection and Appendix B it can be shown that the change of the variable θ can be obtained from a restricted manoeuvre in the form

$$\Phi^{\theta} = P_0 \mathrm{e}^{s_{p1} V_p} \mathrm{e}^{s_2 V_2} \mathrm{e}^{s_{p2} V_p}, \tag{13}$$

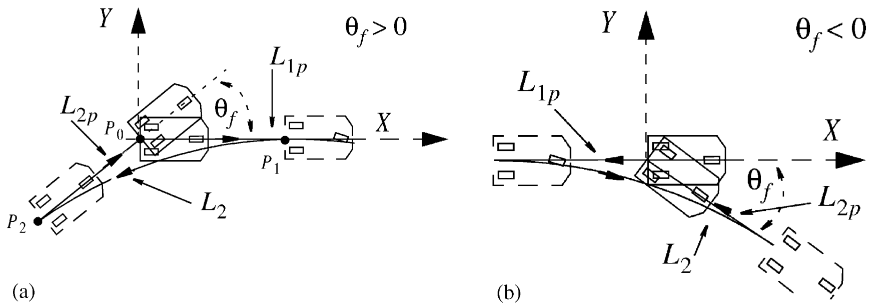


Fig. 6. Restricted manoeuvre: (a) positive change in θ; (b) negative change in θ.

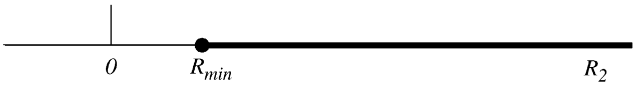


Fig. 7. Manoeuvre solution set.

where V_2 is defined in (7), and the value of the manoeuvre parameters are

$$s_2 = R_2\theta_f \text{ and } s_p = s_{p1} = s_{p2} = \frac{R_2(\cos\theta_f - 1)}{\sin\theta_f}. \tag{14}$$

Due to the way in which V_p and V_2 were chosen, $L_{p1}(P_0 \rightarrow P_1)$ and $L_{p2}(P_2 \rightarrow P_3)$ are straight segments, and $L_2(P_2 \rightarrow P_0)$ is an arc of circumference (see Fig. 6). The parameters s_{p1}, s_{p2} and s_2 represent the length of these segments.

Then, for a desired θ_f, infinite manoeuvres can be found. Each one is determined by a value of R_2. Thus, the positive half-straight line defined by the values of $|R_2|$ which accomplish $|R_2| > R_{\min}$ can be considered as the manoeuvre solution set (see Fig. 7). Each point belonging to this half-straight line represents a value for s_p.

Furthermore, as s_p represents the vehicle forward displacement, from (14) the minimum forward displacement is given by

$$s_{p\min} = \frac{R_{\min}(\cos\theta_f - 1)}{\sin\theta_f}. \tag{15}$$

4. Parallel and diagonal parking in car like vehicles

Parallel and diagonal parking problems can be approached to by using the restricted manoeuvres introduced in the previous section.

4.1. Parallel parking

Assume rectangular models of the vehicles and obstacles. Let $R_{\min}$ be the minimum curvature radius of the vehicle. Assume that a parking to the right is considered, as illustrated in Fig. 8. In order to park the vehicle, a restricted manoeuvre has to be designed in such a way that the initial and final values of variables θ and x are the same, but the value of y is changed.

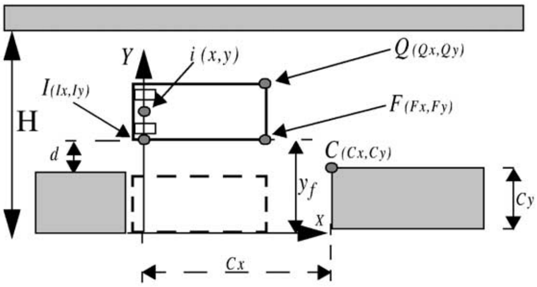


Fig. 8. Parallel parking definition.

In order to design a collision-free manoeuvre, consider the geometric parameters and the points I, F, G and C defined in Fig. 8.

The right rear point, I, is placed along the rotation axe of the vehicle reference point i. The kinematic behaviour of I is similar to i. The Cartesian coordinates of point I will be used in the following to design the parallel parking manoeuvre. The Cartesian reference frame is attached to the point I of the parked vehicle position (see Fig. 8).

4.1.1. Collision-free parallel parking

Collisions are avoided by planning a collision-free path for a virtual vehicle leaving the parking place. Final manoeuvre is obtained using this design in a reverse sequence. For this virtual vehicle, the manoeuvre Φ^{+y}, defined in Section 3.1, is considered

$$\Phi^{+y} = P_0 e^{s_1 V_2} e^{s_2 V_2} e^{s_p V_p} \tag{16}$$

The first requirement for a continuous collision-free path leaving the parking place consists of avoiding the collision of the front of the vehicle with the obstacle ahead (see Fig. 9a). The study of the first possible collision gives the minimum parking place for a collision-free parallel parking (see Appendix C.1)

$$C_{x\min} = \sqrt{R_{\min} 2C_y + l^2 - C_y^2}. \tag{17}$$

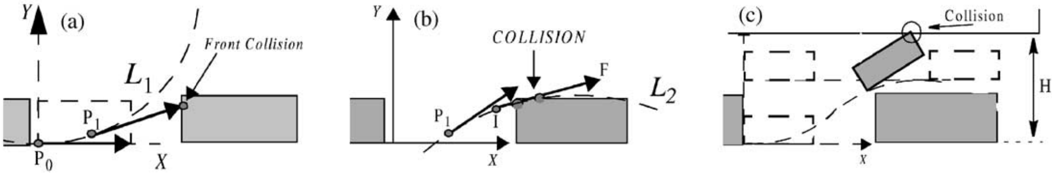


Fig. 9. Possible collisions in parallel parking.

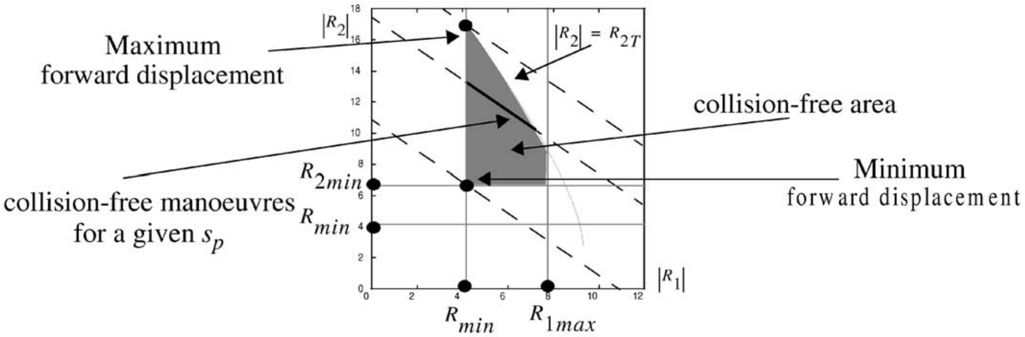


Fig. 10. Collision-free area.

From Appendix C.1 the first noncollision condition is

$$|R_1| \in [R_{\min},\ R_{1\max}] \tag{18}$$

where $R_{1\max} = (C_x^2 + C_y^2 - l^2)/2C_y$.

Even more, the planned path should avoid the collision of the vehicle's lateral side, with the obstacle ahead, along the path L_2 (see Fig. 9b). The condition for avoiding this collision is

$$|R_2| < |R_{2T}(R_1)|, \tag{19}$$

where R_{2T} is a function of R_1 (see Appendix C.2).

The last possible collision is shown in Fig. 9c. It represents the collision with the wall opposite to the parking place. There is no collision in the path if

$$|R_2| > R_{2\min}, \tag{20}$$

where $R_{2\min} = (W^2 + l^2 - (H - y_f)^2)/2(H - y_f - W)$ (see Appendix C.2).

Thus, the second noncollision condition can be expressed as

$$|R_2| \in [R_{\min}, |R_{2T}|]. \tag{21}$$

Then Φ^{+y} is a collision-free parallel parking manoeuvre if

$$|R_1| \in [R_{\min},\ R_{1\max}] \text{ and } |R_2| \in [R_{\min}, |R_{2T}|]. \tag{22}$$

If constraints (22) are represented in the positive half-plane $(|R_1|, |R_2|)$, a bounded region of the manoeuvre solution set is obtained (the dark area in Fig. 10).

This dark region is the collision-free area. Each point belonging to this region represents a free parallel parking manoeuvre. Furthermore, for a given forward displacement, s_p, the collision-free area is reduced to a straight segment (see Fig. 10).

4.1.2. Maximum and minimum forward displacement

According with expression (11), the forward displacement can be expressed as

$$|s_p| = \sqrt{(|R_1| + |R_2|)(2|y_0|) - |y_0|^2}. \tag{23}$$

Then, the maximum and the minimum forward displacement for a collision-free trajectory is given by (Gómez-Bravo, 2000)

$$|s_p|_{\min} = \sqrt{(R_{\min} + R_{2\min})(2|y_0|) - |y_0|^2},$$

$$|s_p|_{\max} = \sqrt{(R_{\min} + R_{2T}(R_{\min}))(2|y_0|) - |y_0|^2}.$$

These values correspond to the straight lines tangent to the collision-free area (see Fig. 10). From $|s_p|_{\min}$ and $|s_p|_{\max}$ a range of points where the vehicle could stop and perform a collision-free manoeuvre is obtained (see Fig. 11)

4.1.3. Minimum lateral displacement

Minimum path length manoeuvres have been previously reported in Neff Patten et al. (1994). In that paper the path is defined by using $R_{\min}$ for both arcs of circumference ($R_1 = R_2 = R_{\min}$).

Nevertheless, the main drawback of this method is the wide motion required in the y-axis direction. Therefore, parallel parking in narrow corridors could be a difficult task. For this aim, minimum lateral displacement can be obtained by considering the minimum value of the y coordinate of point Q (see Fig. 12) (Gómez-Bravo, 2000)

$$Q_y = \sqrt{(|R_2| + W)^2 + l^2} + y_f - |R_2|. \tag{24}$$

This manoeuvre is obtained using the value $R_2 = R_{2T}(R_{\min})$. Thus, it is the same manoeuvre which produce the maximum forward displacement. In Fig. 13 both manoeuvres are compared by simulation.

4.2. Diagonal parking

Diagonal parking can be solved in the same way. Rectangular models of vehicles and obstacles are also considered. Assume that a diagonal parking to the left is defined as shown in Fig. 14.

The problem of parking the vehicle can be solved in two steps: (1) planning a nonholonomic manoeuvre, Φ^θ, to modify the orientation of the vehicle, with θ as the dependent variable; (2) follow a straight line to the end of the parking place. The design of the manoeuvre, will be made by taking into account the geometric parameters and points B and Q defined in Fig. 14.

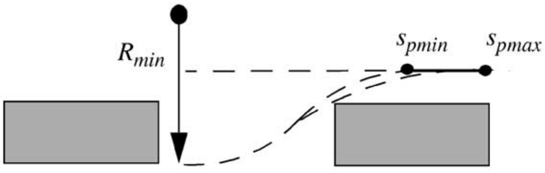


Fig. 11. Range of starting points.

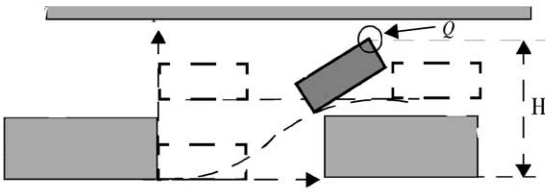


Fig. 12. Point Q lateral displacement.

4.2.1. Collision-free diagonal parking

Expression (13) is used to design a collision-free diagonal parking manoeuvre (see Fig. 6a)

$$\Phi^\theta = P_0 e^{s_{p1}V_p} e^{s_2 V_2} e^{s_{p2}V_p}. \tag{25}$$

The requirement for a collision-free path, consists of avoiding the possible collisions of the points B and Q. These constraints make the value $|R_2|$ be upper and lower bounded (see Appendix D). From expressions (D.3) and (D.5) the noncollision condition is

$$|R_2| \in [|R_2|_{\min}, \ |R_2|_{\max}]. \tag{26}$$

Thus, the straight segment defined by the values of $|R_2|$ which accomplish (26) can be considered as the collision-free solution set (see Fig. 15).

4.2.2. Maximum and minimum forward displacement

According with expression (14), minimum and maximum values for the forward displacement can also be

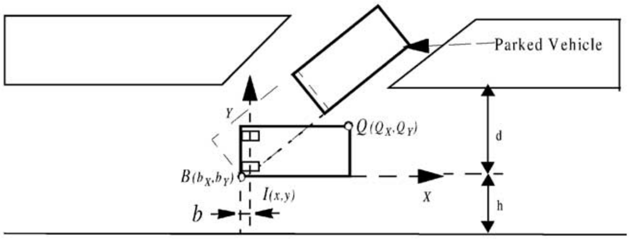


Fig. 14. Diagonal parking definition.

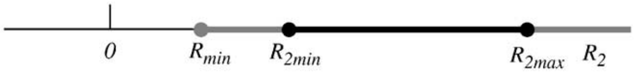


Fig. 15. Manoeuvre solution set.

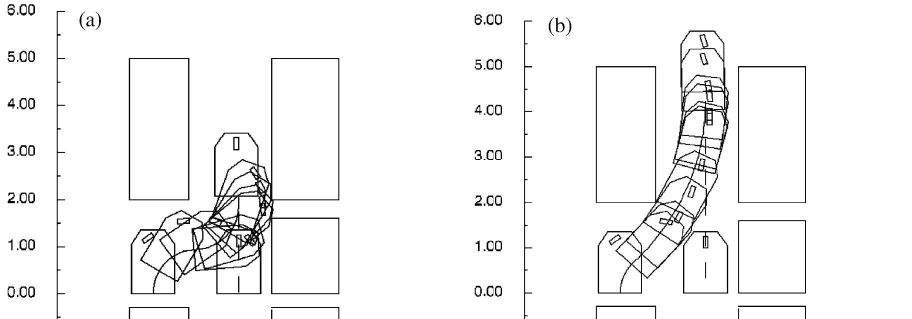


Fig. 13. (a) Minimum path length; (b) minimum lateral displacement.

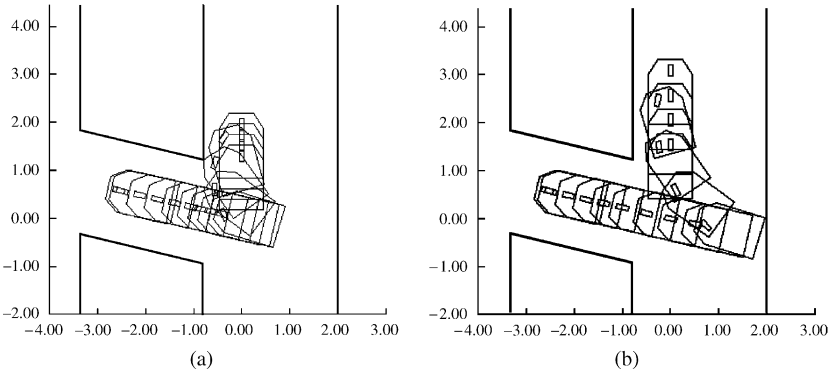


Fig. 16. (a) Minimum forward displacement; (b) maximum forward displacement.

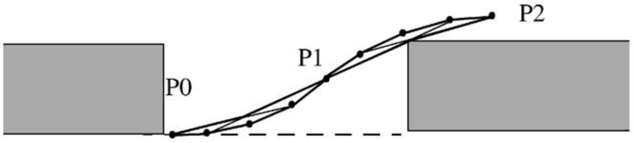


Fig. 17. Collision-free polygon.

obtained

$$s_{p\min} = \frac{|R_2|_{\min}(\cos\theta_f - 1)}{\sin\theta_f}$$

$$s_{p\max} = \frac{|R_2|_{\max}(\cos\theta_f - 1)}{\sin\theta_f}. \quad (27)$$

In Fig. 16 these two extreme manoeuvres are shown.

4.3. Smooth manoeuvres

The main drawback of the path planning methods presented above is the discontinuity in the value of the curvature in the computed paths. Curvature discontinuity could decrease the path tracking performance. In order to avoid this problem, different solutions can be considered. Thus, for parallel parking manoeuvres, a straight segment could be placed between the arcs of circumference (Cuesta et al., 1998). Thus, the curvature discontinuity is decreased.

The solution proposed in this paper consists of applying a recursive algorithm to compute a new path, close enough to the first one, but with curvature continuity. In this way, curves with smooth properties, like β-splines, are applied (Barsky, 1987; Muñoz et al., 1994). Thus, the path $P_2 \rightarrow P_1 \rightarrow P_0$ composed of the arcs of circumference L_1 and L_2, is approximated by a β-spline curve as shown in Fig. 17. In the case of diagonal parking the path $P_1 \rightarrow P_2$ is approximated in order to obtain curvature continuity between the arc of circumference and the straight line segments.

A β-spline is a parametric piece-wise cubic. This curve presents continuity of first and second derivative at the joints between adjacent segments. This type of curves is specified by a set of points called *control vertices*. These vertices connected in sequence build up a *control polygon*. The curve tends to mimic the overall shape of the control polygon.

The most significant property of the β-splines curves from the point of view of this application, is the Hull-convex property (Barsky, 1987). By this property, it is assured that the surface where the curve goes across, is bounded by the control polygon. There are other approaches for generating paths with continuous curvature form arcs of circumferences by using clothoids (Scheuer and Fraichard, 1997). Nevertheless, the collision-free polygon obtained with this method is bigger than the polygon generated with β-splines curves. Moreover, clothoids require higher computational cost than β-splines curves.

Thus, the smoothing technique is performed in three steps: (1) For a given forward displacement, determined by the point where the robot has stopped, one of the manoeuvres belonging to the collision-free segment (see Section (4.1.1) and Section (4.2.1)) is selected; (2) from the collision-free manoeuvre selected in Step 1, a collision-free control polygon is built by choosing points from the arcs of circumference (see Fig. 17) (in Muñoz et al. (1994) and Gómez-Bravo, 2000) it is described the technique to select these points in order to assure the existence and built this collision-free polygon); (3) then, a collision-free β-spline curve can be computed.

Fig. 18 shows a simulated parking manoeuvre in a realistic dimension scene in which a β-spline path is generated from two arcs of circumference. The steering angle is bounded at 0.4 rad and $v = 0.2$ m/s, showing the efficiency of this method in realistic situations.

5. Election of parking manoeuvre by fuzzy logic

In the previous section, different optimal manoeuvres have been introduced. Each of such manoeuvres is optimal only with respect to a simple given criteria:

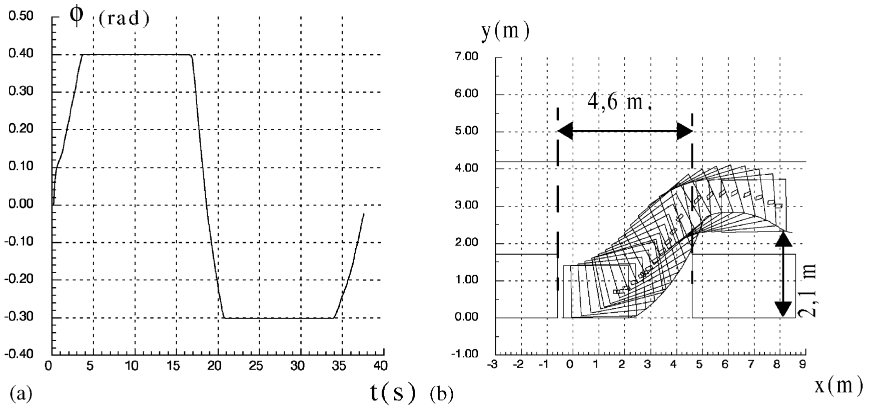


Fig. 18. β-Spline curves for a parallel parking.

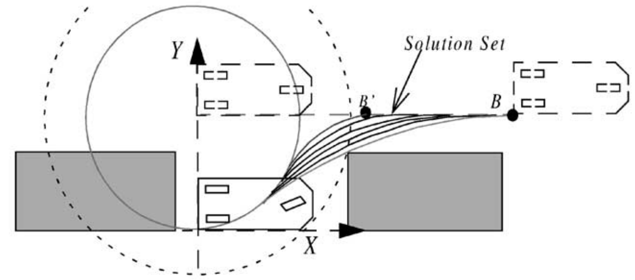


Fig. 19. Different manoeuvres from the solution set.

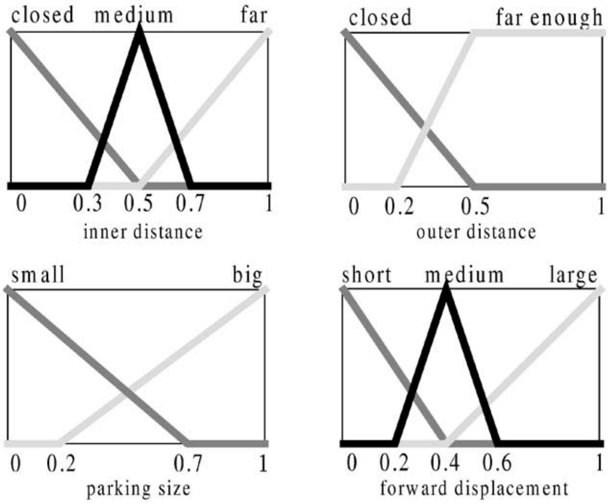


Fig. 20. Membership functions for input and output variables.

minimum lateral displacement, minimum forward displacement, minimum control effort, and so on. These manoeuvres build up a solution set of collision-free manoeuvres, where the optimal manoeuvres are the extreme ones (see Fig. 19). However, from a practical point of view, depending on the environment, the election of one of such extreme manoeuvre could be not a good option. Thus, for instance, a minimum forward displacement could imply a higher lateral displacement (the vehicle could navigate very close to the opposite parking line) and a poor path tracking performance due to the high curvature involved.

Therefore, in practice, it is interesting to select a manoeuvre from the solution set. Such election should be made by considering the environment where the vehicle have to park in, and can be done by a fuzzy logic system in the same way that a human driver selects the starting point for the reverse movement depending on the characteristics of the parking place.

Thus, fuzzy logic plays an important role in the practical application of this parking method, by considering the trade-off between optimal manoeuvre and environment conditions in order to perform a more robust and safer parking manoeuvre.

The fuzzy election system will compute the best forward displacement (or more precisely, the zone in the solution segment where the vehicle should start the reverse movement) based on the lateral distances, to the inner and outer parking lines, and how large is the parking place (measured with respect to $C_{x\min}$). It is interesting to note that the output of the fuzzy system should be considered as a point around which the vehicle should stop, but not as the exact point where the vehicle must stop. Indeed, once the vehicle is stopped, the parking manoeuvre is computed from the actual position (that will be collision-free since it starts from the solution set).

The inputs to the fuzzy system are: the *inner distance* (*id*), *outer distance* (*od*) and *parking size* (*ps*); and the output is the *forward displacement* (*fp*). The membership functions of these variables are shown in Fig. 20.

The knowledge of a human driver can be incorporated into the fuzzy election system by means of the following rule base:

IF *outer distance* IS *closed* OR *inner distance* IS *far*
  THEN MAKE *forward displacement large*
IF *inner distance* IS *closed*
  THEN MAKE *forward displacement short*

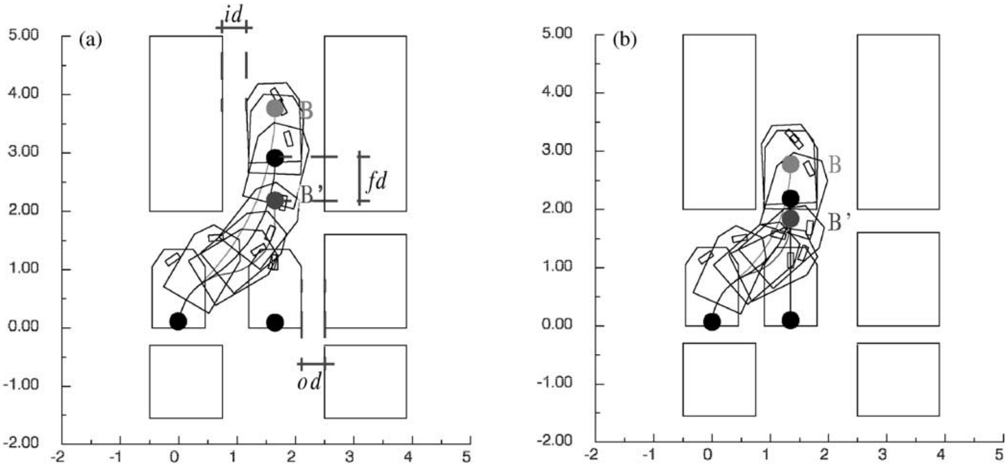


Fig. 21. Election of parking manoeuvres (*fd*) in different situations; (a) *id* = 0.45; *od*=0.45; *ps*=0.23; *fd*=0.43. (b) *id*=0.2; *od*=0.7; *ps*=0.23; *fd*=0.19.

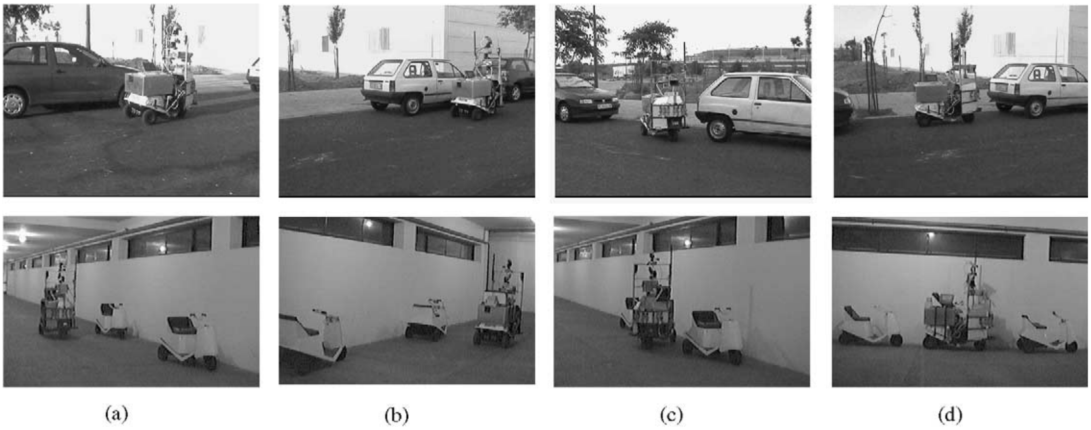


Fig. 22. ROMEO-3R performing a lateral parking.

IF *outer distance* IS *far enough* OR *inner distance* IS *medium*
THEN MAKE *forward displacement medium*
IF *parking size* IS *small*
THEN MAKE *forward displacement medium*
IF *parking size* IS *big*
THEN MAKE *forward displacement short*

Fig. 21 shows the application of the fuzzy election system in different situations to obtain a convenient forward displacement, giving a robust manoeuvre keeping away from the obstacles around. In the first case, inner and outer distances are 0.45, while the parking size is 0.23 with respect to $C_{x\min}$ (which takes a value of 2 for the given environment), yielding a medium forward displacement, around 0.43 (note that B and B′ correspond to the extreme manoeuvres). In the second case, the vehicle is closer to the left line of vehicles (inner distance is 0.2) and a shorter forward displacement (namely around 0.19) is preferred.

In a similar way, the fuzzy system can be used to decide on the best diagonal parking manoeuvre, but then, the parking size can be neglected.

6. Autonomous parking with ROMEO-3R vehicle

The method presented in the above sections has been implemented in the autonomous vehicle ROMEO-3R (see Fig. 22). This vehicle has been designed and built at the University of Seville for the experimentation of intelligent components and autonomous navigation strategies (Ollero et al., 1999a). ROMEO-3R is the result of the adaptation of a conventional three-wheel electrical vehicle. Vehicle's length and width are 1.34 and 0.7 m, respectively, with a maximum steering angle of 0.7 rad. It carries on several sensors including video cameras for teleoperation and autonomous navigation, scanner laser range finder and an heterogeneous configuration of ultrasonic sensors. The system also has a differential GPS for outdoor position estimation.

The ROMEO-3R controller has been implemented in an industrial 486 PC system, under Lynx real time operating system, with motion control and data acquisition boards to obtain the signals from navigation and ultrasonic sensors.

The low level position estimation is performed by means of dead reckoning and sensor fusion using navigation sensors. Model (3) is used for dead-reckoning. Furthermore, the vehicle heading (θ in Eq. (3)) can be measured by using an orientation gyro and a compass. Then, an extended Kalman filter is applied to provide a more accurate and reliable position estimation.

The autonomous parking has been implemented in the frame of a behaviour-based control architecture, combining planning and reactivity (Cuesta et al., 1998). This architecture takes advantage of the reactive navigation robustness, and relatively low precision required performing some task such as navigating parallel to a line of vehicles looking for a parking place. At the same time planned navigation allows to consider the nonholonomic constraints of the vehicle, and to perform an accurate navigation where high precision is required due to the characteristics of the manoeuvre.

There are multiple simultaneous behaviours working in a cooperative scheme toward a given goal. In this strategy each behaviour working in parallel produces its own motion control command, named $\boldsymbol{c_i}$ (steering and velocity), where i stands for each behaviour. The output of each behaviour is weighted by its behaviour weight (BW_i), and combined by means of fuzzy logic producing the motion control law

$$c = \frac{\sum_i (BW_i c_i)}{\sum_i BW_i}, \quad i = \text{flw, frw, oa}, \ldots . \tag{28}$$

The behaviour weights are calculated dynamically taking into account the situation the vehicle is in, and determining the applicability of each behaviour according to the context. Furthermore, it is possible to define priorities between behaviours, taking into account the weight of one behaviour to compute the weight of the other ones. For example, if the behaviour weight for left wall following BW_{flw} and for obstacle avoidance BW_{oa} get a high value at the same time, the priority of the left wall following behaviour will decrease using the following equation:

$$BW_{\text{flw}} = BW_{\text{flw}}(1 - BW_{\text{oa}}). \tag{29}$$

When the vehicle is looking for a parallel parking (see Fig. 22a), three behaviours cooperate: *wall or line of vehicles following*, *obstacle avoidance* and *lateral parking manoeuvre*. In this phase the parallel parking behaviour is only searching for a place large enough to park and does not generate commands to the vehicle.

The line of vehicles is followed navigating reactively, using a virtual sensor which is implemented taking into account the measurements of the two left-hand side sonars of the vehicle. Both control and sensor data fusion are made by means of fuzzy logic (Ollero et al., 1999c). The obstacle avoidance behaviour is also implemented using fuzzy logic.

When a parking place is found (Fig. 22b), length and depth of the parking place are estimated. The estimation is performed by taking into account the specifications of the ultrasonic sensors in order to ensure the robustness of such estimation (see Ollero et al., 1999b, for details). Most of the experiments presented in this section were carried out using only the two ultrasonic sensors located at the left-hand side of the vehicle, while in other examples the right-hand sensors were also used (when the vehicle had to park between two parking lines). These sensors have a sensing range of 0.6–3 m, a resolution of 2 mm. and an angle of 15°. Their switching and ultrasonic frequencies are 1 Hz and 80 kHz, respectively.

If the place was not wide enough it is rejected and the vehicle keeps navigating without stop looking for a new parking place. In other case, lateral parking behaviour performs the following steps: (1) selects the most convenient starting location by means of the fuzzy election system; (2) stops the vehicle around that point (wall following behaviour weight is set to zero); (3) computes the collision-free trajectory satisfying nonholonomic constraints, starting from the vehicle has stopped; (4) activates the tracking of the planned path in reverse (Fig. 22c–d).

Several techniques have been implemented in the controller of the autonomous vehicle to perform the path tracking, including Pure Pursuit, Generalized Predictive Path Tracking, and Fuzzy Path Tracking (Ollero et al., 1994). It has been experimentally demonstrated that all these techniques give similar performance for low speed parking manoeuvring. Thus, the simplest technique, Pure Pursuit, has been used.

Efficiency and applicability of this strategy has been widely supported by the implementation in the autonomous vehicle. Several experiments have been carried out in complex environments, where the robot had to decide whether or not the parking place was large enough to park in.

Examples of different parallel parking manoeuvres, with different scenes, displayed from the data recorded during two experiments with the ROMEO vehicle, are shown in Fig. 23. For each one, the manoeuvre in cartesian space and the curvature radii vs. the path length is represented. In these experiments, the effect of the lateral distances on the selection of the starting point is shown.

Fig. 24 shows three real experiments which illustrate the efficiency of these manoeuvring methods with

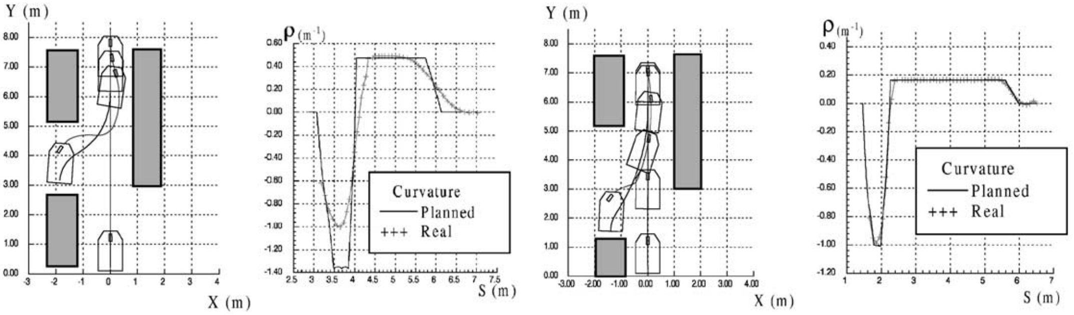


Fig. 23. Different parallel parking manoeuvres, with ROMEO 3R.

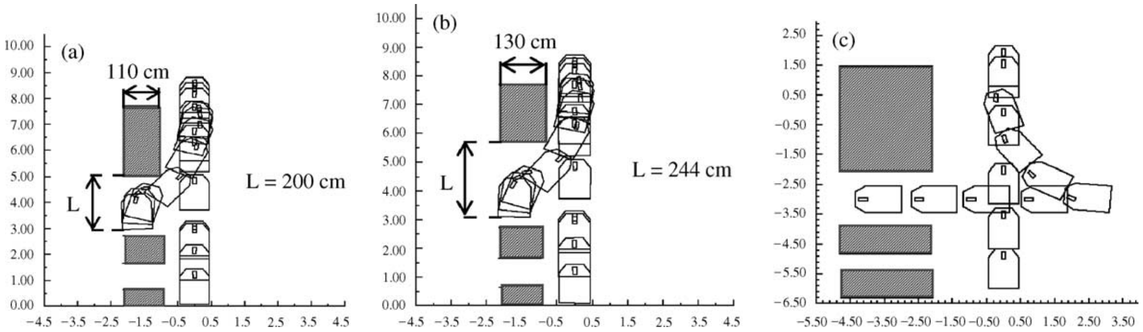


Fig. 24. Different parking manoeuvres, with ROMEO 3R.

different parking place dimension. In all of them, ROMEO first found a parking place that was not wide enough and it was rejected; later a suitable place is found and the manoeuvre is performed. Fig. 24a illustrates a parallel parking manoeuvre in an environment with obstacles similar to ROMEO's size. Fig. 24b illustrates a situation with a wider obstacle ahead (which implies a large value in reverse displacement). Finally, Fig. 24c represents a manoeuvre displayed from the data recorded in a row parking situation.

7. Conclusions

In this paper a method to generate manoeuvres for nonholonomic wheeled vehicles has been presented. The method is based on the election of a sequence of vectors that generate a closed path in which only the value of one of the variables changes. A restricted manoeuvre can be computed composed of two arcs of circumference and a straight line segment. Furthermore, the tracking performance of that manoeuvre can be improved computing a β-spline curve based on the original manoeuvre providing a collision-free smother path.

The main advantages of the proposed technique with respect to other existing methods are that (1) both parallel and diagonal parking manoeuvres are considered; (2) a set of collision-free manoeuvres and feasible starting points can be obtained on-line (instead of a single manoeuvre from a given starting location); (3) several optimal manoeuvres can be defined.

Artificial intelligent techniques, namely fuzzy logic, play an important role in the practical application of the method. Thus, a fuzzy system is used to select a manoeuvre from the solution set according with the environment, dealing with optimality, path tracking performance and collision avoidance trade-off.

This technique has been straightforward implemented in the frame of a fuzzy behaviour-based control architecture combining planning and reactivity. It shows the flexibility and capabilities of fuzzy logic to approach the so called hybrid control problem. Thus, it takes advantage of the reactive navigation robustness to navigate parallel to a line of vehicles; on the other hand, planned navigation allows to consider the nonholonomic constraints and to perform an accurate navigation where high precision is required (due to the manoeuvre characteristics).

The efficiency and low hardware requirements of the proposed method has been demonstrated using the nonholonomic mobile robot ROMEO-3R.

Acknowledgements

The authors are indebted to the anonymous reviewers and the Deputy Editor for their helpful and detailed comments, which have helped to improve the quality and presentation of the paper. This work has been partially supported by the CYCIT TAP99-0926-C04-01.

Appendix A. Exponential manoeuvre notation

Let p_0 be a point on a manifold and V a vector field defined around this point. There is exactly one parametric path, $L(s)$, starting at p_0 and following V. That path satisfies

$$L(0) \equiv p_0 \text{ and } \dot{L}(s) = V_{L(s)}, \tag{A.1}$$

where $\dot{L}(s)$ is the time derivative of $L(s)$ and $V_{L(s)}$ is the value of V along $L(s)$.

The relation between L and V is described by the expression

$$p(s) = L(s) = p_0 \mathrm{e}^{sV}. \tag{A.2}$$

Therefore, the exponential e^{sV} of a vector field means: "*starting from the point* 'p_0' *and following the vector* 'V' *the amount* 's' *of the path parameter*, *the point* $p(s)$ *is reached*". Thus, let be Φ a manoeuvre where p_0 is the starting point. Let the vehicle follows V_1 for a given s_1, until a point p_1, and then follows V_2 for a given s_2, until a new point p_2. Thus Φ can be expressed as

$$\Phi = p_0 \mathrm{e}^{s_1 V_1} \mathrm{e}^{s_2 V_2} \tag{A.3}$$

with

$$p_1 \equiv p_0 \mathrm{e}^{s_1 V_1},$$

$$p_2 \equiv p_1 \mathrm{e}^{s_p V_p} \equiv (p_0 \mathrm{e}^{s_2 V_2}) \mathrm{e}^{s_p V_p}. \tag{A.4}$$

Appendix B. Restricted manoeuvre with *y*

The vector V_P which generates the path L_p, is obtained from $\dot{y} = 0$ (see Section 2), i.e.,

$$\dot{x} \sin\theta = 0. \tag{B.1}$$

Thus, if $\dot{x} \;/\!= 0$ then $\sin\theta = 0$, $\cos\theta = 1$, $\dot{\theta} = 0$ and it follows $V_P = [v(t)\, 0\, 0]^{\mathrm{T}}$. Therefore, L_p is given by $\theta = n\pi$ where $n = 0, 1, \ldots$ (Thus, L_p is a straight line parallel to the X-axis, and s_p corresponds to the vehicle forward displacement).

It could be shown that a positive change in the variable y can be obtained from the restricted manoeuvre

$$\Phi^{+y} = P_0 \mathrm{e}^{s_1 V_1} \mathrm{e}^{s_2 V_2} \mathrm{e}^{s_p V_p}, \tag{B.2}$$

where V_1 and V_2 are defined in (7).

The path (see Fig. 4a) can be calculated by integrating (3) along paths $P_0 \to P_1 \to P_2 \to P_3$. Let $\Delta_{\mathrm{T}}x, \Delta_{\mathrm{T}}y, \Delta_{\mathrm{T}}\theta$ be the increments of the variables x, y, θ respectively in full manoeuvring, then the conditions for the restricted manoeuvre are $\Delta_{\mathrm{T}}x = 0, \Delta_{\mathrm{T}}y = |y_0|, \Delta_{\mathrm{T}}\theta = 0$. Thus, the manoeuvre is defined by the equations:

$$\Delta_{\mathrm{T}}x = 0 = (R_1 - R_2)\sin\theta_1 + s_p,$$

$$\Delta_{\mathrm{T}}y = |y_0| = (R_2 - R_1)(\cos\theta_1 - 1),$$

$$\Delta_{\mathrm{T}}\theta = 0 = \Delta_1\theta + \Delta_2\theta = \frac{s_1}{R_1} + \frac{s_2}{R_2}, \tag{B.3}$$

Therefore, the value of the manoeuvre parameters are

$$s_1 = R_1\theta,\ s_2 = -R_2\theta,\ s_p = (R_2 - R_1)\sin\theta_1$$

and

$$\theta_1 = \operatorname{acos}\left(1 - \frac{|y_0|}{R_1 - R_2}\right). \tag{B.4}$$

It should be noted that Eq. (B.4) admits two symmetric solutions due to the fact that θ_1 also presents two symmetric values, the solution selected for this application is the one shown in Fig. 4a.

Appendix C. Noncollision conditions in parallel parking

C.1. First noncollision condition

The first requirement for a continuous collision-free path leaving the parking place consists of avoiding the collision of the front of the vehicle with the obstacle ahead. Note that Φ^{+y} satisfies this constraint if there is no collision in the path L_1 (when the vehicle is following vector V_1 from P_0 to P_1) (see Fig. 25a).

Let IF be the segment defined by the vehicle's right-hand lateral side. From Fig. 25b the noncollision requirement can be expressed as

$$F_x = \sqrt{R_1 2C_y + l^2 - C_y^2} < C_x. \tag{C.1}$$

Thus, the minimum parking place for a collision-free parallel parking is determined by

$$C_{x\min} = \sqrt{R_{\min} 2C_y + l^2 - C_y^2}. \tag{C.2}$$

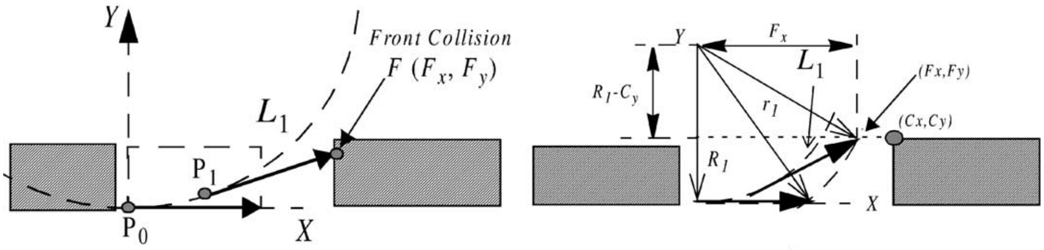
Therefore, the maximum value of R_1 for a collision-free manoeuvre is

$$R_{1\max} = \frac{C_x^2 + C_y^2 - l^2}{2C_y}. \tag{C.3}$$

C.2. Second noncollision condition

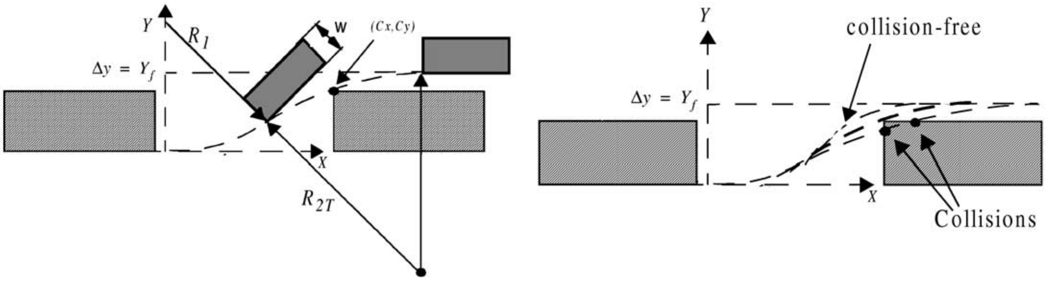
The planned path should avoid the collision of the vehicle's lateral side with the obstacle ahead along the path L_2 (when the vehicle is following vector V_2 from P_1 to P_2) (see Fig. 9b).

There is no collision in the path L_2, if the segment IF is out of the polygon defined by the obstacle ahead.



(a) (b)

Fig. 25. Avoiding front collision.



(a) (b)

Fig. 26. Definition of manoeuvre Φ_T^{+y}.

Since the path L_2 is an arc of circumference, and the segment IF is tangent to the arc at the point I, this condition will be fulfilled for every point E on segment IF if it is verified by I.

Let Φ_T^{+y} be the manoeuvre with the point (C_x, C_y) belonging to the arc of circumference in L_2, and R_1 satisfying (18) (see Fig. 26a). Then, according to the above considerations Φ_T^{+y} is a collision-free manoeuvre.

This manoeuvre has a significant property, if a manoeuvre is designed with this value of R_1 and it is under Φ_T^{+y} then it will present collisions, however, if it is over Φ_T^{+y} it will be a collision free manoeuvre (see Fig. 26b). This type of manoeuvre is defined by (Gómez-Bravo, 2000):

$$R_2 = R_{2T}(R_1) = R_1 - \frac{y_f}{1 - \cos\theta_T}, \tag{C.4}$$

where $\theta_T(R_1) = \mathrm{a}\cos(k)$ and k is the real solution of $Ak^2 + Bk + C = 0$, with

$$\begin{aligned}
A &= (\vartheta - \alpha + \partial - \Gamma)^2 + \xi^2,\\
\alpha &= C_x^2, \quad \partial = 2(C_y - y_f)R_1,\\
B &= 2(\vartheta - \alpha + \partial - \Gamma)(\alpha + \vartheta - \partial + \Gamma + \varepsilon),\\
\vartheta &= y_f^2, \quad \varepsilon = 2(C_y - y_f)y_f,\\
C &= (\alpha - \vartheta - \partial + \Gamma + \varepsilon)^2 - \xi^2,\\
\xi &= 2C_x y_f, \quad \Gamma = (C_y - y_f)^2,
\end{aligned} \tag{C.5}$$

For each value of R_1 a value of R_{2T} exist. Then for a given forward displacement, this pair of values represent the manoeuvre in which the point (C_x, C_y) belong to the arc of circumference in L_2. Thus, in order to avoid the collision, any of the solution with $R_2 > R_{2T}$ and $|R_2| < |R_{2T}|$ should be selected. The maximum value of $|R_{2T}|$ is obtained by

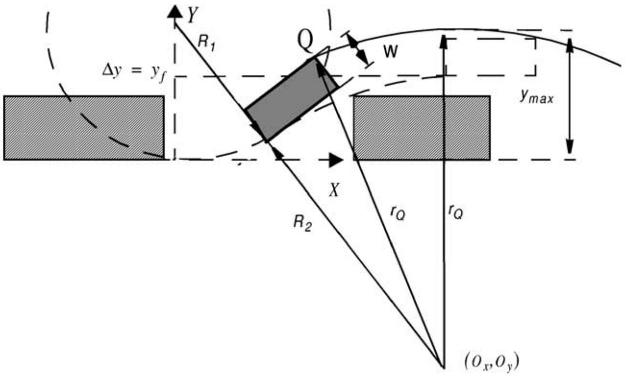


Fig. 27. Point Q trajectory.

$$R_{2T\max} = |R_{2T}(R_{\min})| \tag{C.6}$$

The last possible collision is shown in Fig. 9c. It represents the collision with the wall opposite to the parking place. According with Figs. 27 and 9c this situation is expressed by

$$\begin{aligned}
Q_y &= |r_Q| + O_y = \sqrt{(|R_2| + W)^2 + l^2} + O_y\\
&= \sqrt{(|R_2| + W)^2 + l^2} + y_f - |R_2| = H
\end{aligned} \tag{C.7}$$

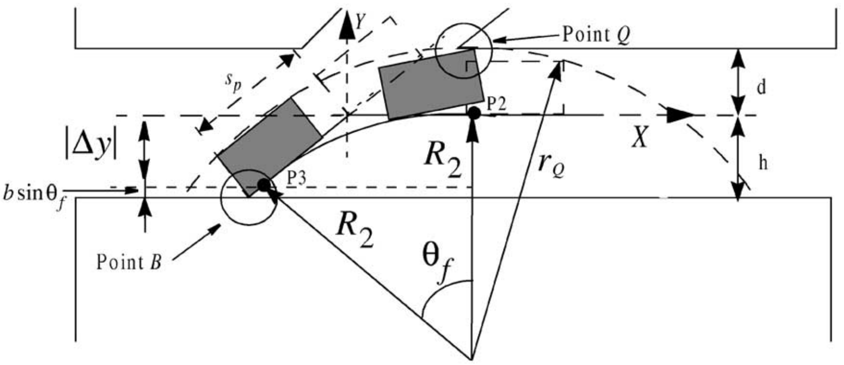


Fig. 28. Possible collisions in diagonal parking.

From expression (C.7) the minimum value of R_2 can be obtained as

$$R_{2\min} = \frac{W^2 + l^2 - (H - y_f)^2}{2(H - y_f - W)}.$$

Values below that $R_{2\min}$ will produce manoeuvres collide with the opposite wall.

Appendix D. Noncollision conditions in diagonal parking

The requirement for a collision-free diagonal parking, consists of avoiding the collision of the points B and Q when the vehicle is following vector V_2 from $P2$ to $P3$ (see Fig. 28). From this figure, it can be written

$$s_p \sin\theta_f = |\Delta y| = |R_2|(1 - \cos\theta_f) = h - b\sin\theta_f \qquad \text{(D.1)}$$

A collision of point B can be described by the expression

$$|\Delta y| + b\sin\theta_f = h. \qquad \text{(D.2)}$$

Then, the constraint for avoiding the collision of point B is

$$|R_2| < |R_2|_{\max} = \frac{h - b\sin\theta_f}{(1 - \cos\theta_f)}. \qquad \text{(D.3)}$$

A collision of point Q can be described by the expression

$$d = Q_{y\max} = |r_Q| - |R_2| = \sqrt{(|R_2| + W)^2 + l^2} - |R_2|, \qquad \text{(D.4)}$$

where W represents the vehicle's width. Thus, the condition for avoiding point Q collision is

$$|R_2| > |R_2|_{\min} = \frac{w^2 + l^2 - d^2}{2(d - w)}. \qquad \text{(D.5)}$$

with $d \in [w, \sqrt{w^2 + l^2}]$. In case of $d > \sqrt{w^2 + l^2}$, $|R_2|$ would be lower bounded by $|R_{\min}|$.

References

Barsky, B., 1987. Computer Graphics and Geometric Modelling using Beta-Splines. Springer, Berlin.

Cuesta, F., Gómez-Bravo, F., Ollero, A., 1998. A combined planned/reactive system for motion control of vehicles manoeuvres. IFAC Workshop on Motion Control, IMC98. Grenoble, France, September, pp. 303–308.

Gómez-Bravo, F., 2000. Planificación de maniobras en sistemas robóticos no holónomos. Aplicaciones en robots móviles. Report GAR00-14, Dept. Ingeniería de Sistemas y Automática, Univ. de Sevilla.

Gómez-Bravo, F., Ollero, A., 1995. Dynamic path planning of free floating manipulator. ICAR'95, pp. 447–457.

Jacobs, P., 1990. Minimal lenght curvature constrained path in the presence of obstacles. LAAS/CNRS Report 90042, Toulouse, France.

Jiang, K., Seneviratne, L.D., 1999. A sensor guided autonomous parking system for nonholonomic mobile robot. Proceedings of the 1999 IEEE International Conference on Robotics and Automation, pp. 311–316.

Latombe, J.C., 1991. Robot motion Planning. Kluwer Academic Publishers, Dordrecht, pp. 424–425.

Laumont, J.P., Jacobs, P.E., Taix, M., Murray, M., 1994. A motion planner for nonholonomic mobile robots. IEEE Transactions on Robotics and Automation 10 (5), 577–593.

Mukherjee, R., Anderson, A., 1994. A surface integral approach to the motion planning of nonholonomic systems. Journal of Dynamic System, Measurement, and Control 116 (3), 315–325.

Muñoz, V., Ollero, A., Prado, M., Simón, A., 1994. Mobile robot trajectory planning with dynamic and kinematic constraints. Proceedings of the IEEE International Conference on Robotics and Automation, vol. 4, pp. 2802–2807.

Murray, R., Sastry, S.S., 1990. Steering nonholonomic system using sinusoids. Proceedings of the IEEE International Conference on Decision and Control, pp. 1136–1141.

Neff Patten, W., Wu, H., Cai, W., 1994. Perfect parallel parking via pointryagin's principle. Journal of Guidance, Control and Dynamics 116, 723–729.

Ollero, A., Arrue, B.C., Ferruz, J., Heredia, G., Cuesta, F., López-Pichaco, F., Nogales, C., 1999a. Control and perception components for autonomous vehicle guidance. Application to the Romeo vehicles. Control Engineering Practice 7 (10), 1291–1299.

Ollero, A., Cuesta, F., Braunstingl, R., Arrue, B.C., Gómez-Bravo, F., 1999b. Perception for autonomous vehicles based on proximity sensors. Application to autonomous parking. 14th IFAC World Congress, pp. 451–456.

Ollero, A., García-Cerezo, A., Martinez, J., 1994. Fuzzy supervisory path tracking of mobile robots. Control Engineering Practice 2 (2), 313–319.

Ollero, A., Ulivi, G., Cuesta, F., 1999c. Fuzzy logic applications in mobile robotics. In: Verbruggen, H.B., Zimmermann, H.-J., Babuska, R. (Eds.), Fuzzy Algorithms for Control. Kluwer Academic Publishers, Dordrecht, pp. 301–324.

Paromtchik, I., Laugier, C., 1996. Motion generation and control for autonomous car maneuvering. Proceedings of the 1996 IEEE International Conference on Robotics and Automation, pp. 3117–3122.

Paromtchik, I., Laugier, C., Gusev, S.V., Sekhavat, S., 1998. Motion control for autonomous an autonomous vehicle. Proceedings of the International Conference on Control, Automation, Robotics and Vision, Vol. 1, pp. 136–140.

Scheuer, A., Fraichard, Th., 1997. Continuous-curvature path planning for car-like vehicles. Proceedings of IROS 97, pp. 997–1002.

Sekhavat, S., 1997. Planification de mouvements sans collision pour systèmes non holonomes. Ph.D. Thesis, LAAS-CNRS.

Vafa, Z., Dubowsky, S., 1987. On the dynamics of manipulators in space using the V.M. manipulator approach. Proceedings of the IEEE International Conference on Robotics and Automation, IEEE Computer Society Press, Washington, DC, pp. 579–585.