Soft Comput (2007) 11: 249–257
DOI 10.1007/s00500-006-0066-6

FOCUS

Darío Maravall · Javier de Lope

Multi-objective dynamic optimization with genetic algorithms for automatic parking

Published online: 26 September 2006

Abstract This paper addresses the problem of automatic parking by a back-wheel drive vehicle, using a biomimetic model based on direct coupling between vehicle perceptions and actions. This problem is solved by means of a bio-inspired approach in which the vehicle controller does not need to know the car kinematics and dynamic, neither does it call for a priori knowledge of the environment map. The key point in the proposed approach is the definition of performance indices that for automatic parking happen to be functions of the strategic orientations to be injected, in real time, to the car-like robot controller. This solution leads to a dynamic multi-objective optimization problem, which is extremely hard to be dealt analytically. A genetic algorithm is therefore applied, thanks to which we obtain a very simple and efficient solution.

1 Introduction

Optimization is one of the most common and pervasive issues in real-world engineering and economic systems and it is at the heart of any decision-making task in which a choice must be made between several alternatives to achieve multiple, sometimes conflicting, objectives. The objectives are generally formalized as analytical functions or performance indices.

The technical literature on optimization methods is really extensive, as this fundamental subject has received a tremendous amount of attention since it came of age about 50 years ago. Single-objective optimization is by far the most researched problem in this field, although many real-life situations are multi-objective optimization problems per se. A multi-objective optimization problem is very often converted into a single-objective optimization case by integrating the multiple performance indices into a single one [1].

The standard solution for truly multi-objective optimization problems is to find the so-called Pareto-optimal front. The Pareto-front is formed by the solutions in which any change in any of the decision variables aimed at improving a particular performance index will produce a deterioration in some of the other performance indices. Due to the inherent difficulties in calculating the analytical Pareto-optimal surfaces for many real-world systems, evolutionary methods have lately been actively applied to solve multi-objective optimization problems [2–4].

In this paper, we present an engineering problem (automatic car parking in the presence of unknown obstacles) that ultimately leads to the formalization of an active multi-objective dynamic optimization problem as a cooperative game. However, instead of using evolutionary controllers to search for an optimal, or at least efficient, set of *if–then* reasoning rules, we propose a much simpler method based on what we call a biomimetic approach to sensory–motor coordination. Sensory–motor coordination is a central issue in the design of most real-world engineering systems, as the designer somehow has to deal with the coordination of perceptual information and physical actions. We have already applied this biomimetic approach to solve specific manipulation and locomotion problems in articulated mechanisms [5, 6] and wheeled vehicles [7,8]. In this paper we illustrate our approach to sensory–motor coordination with the problem of automatic car parking, including the autonomous avoidance of unknown obstacles. Actually, in this paper itself we enlarge our previous work [9] on automatic car parking by considering a much more complex situation, i.e., the presence of unknown obstacles which makes the basic problem of the coordination of multiple, conflicting objectives an extremely harder multi-criteria optimization problem, as explained in paragraph 3 of this paper. The relevance of evolutionary computation in our optimization problem lies in the analytical intractability of the cooperative coordination of the antagonistic objectives appearing in the automatic parking problem.

D. Maravall (✉) · J. de Lope
Department of Artificial Intelligence, Faculty of Compute Science, Universidad Politécnica de Madrid, Campus de Montegancedo, 28660 Madrid, Spain.
E-mail: dmaravall@fi.upm.es
E-mail: jdlope@fi.upm.es

We therefore introduce a kind of meta-heuristic coordination which is optimized by applying a genetic algorithm.

Since Zadeh [10], the father of the novel concepts of fuzzy sets and linguistic variables, published his seminal paper on fuzzy systems, in which he made some prescient comments on the interest of human car driving as an inspiring task and research topic for building artificial intelligence systems, and since Zadeh [11] himself illustrated the concept of fuzzy algorithm with the example of a human being parking a car. As commented several times before since Zadeh pioneering papers, and not "surprisingly" fuzzy logic controllers (FLCs) have been one of the most frequently proposed approaches to the design and implementation of automatic par parking systems. The majority of them are based on simulated car models [12,13] or on small laboratory prototypes [14,15], or even on commercial prototypes. As for the most recent research efforts on the application of the FLCs to automatic car parking it is worthwhile to *mention the work of Fraichard and colleagues at the French* INRIA (*Institut National de Recherche en Informatique et en Automatique*) where they have been developing during last years [16,17] on a research program on road transport, more specifically, within the European research program Prometheus – which stands for Programme for an European Traffic with Highest Efficiency and Unprecedented Safety, and also in the Praxitèle Project, a French research program on public individual transport.

The FLC developed by Fraichard and colleagues has been tested on an electric car-like vehicle prototype for performing control trajectories including collision avoidance and parallel parking maneuvers.

The remainder of the paper is organized as follows. The biomimetic approach that we advocate for the design of autonomous robots is very briefly outlined in the next section. This approach has two noteworthy properties. First, the controller does not need to know the agent or robot kinematics or dynamics and, second, neither does it call for a priori knowledge of the spatial distribution or map of the working environment. Thanks to these two features, very simple designs can be obtained without prior knowledge being injected into the robot controller. This approach is then tailored to the case of automatic car parking, and we go on to address the specific question mentioned above of coordinating contradictory criteria, proposing a solution based on genetic algorithms.

2 Biomimetic approach for sensory and motor coordination in autonomous robots

The idea underpinning the method is to optimize the measurable behavior indexes using appropriate sensors. The optimization is solved by means of heuristic techniques, which makes the robot controller highly flexible and very simple. In the manner that living beings solve their physical control problems, like manipulation and locomotion, the robot develops a behavior strategy based on the perception of its environment, embodied as behavior indexes and aimed at improving (optimizing) the evolution of the aforementioned behavior indexes. It does all this following the known *perception–decision–action* cycle.

The tailoring of this biomimetic approach to the parking problem is illustrated in Fig. 1.

The robot vehicle considered in this paper is a conventional back-wheel drive car, whose dynamic equations can be modeled, for the low-speed range typical of parking maneuvers, as:

$$\begin{aligned} \dot{x}(t) &= v(t)\cos\theta(t), \\ \dot{y}(t) &= v(t)\sin\theta(t), \\ \dot{\theta}(t) &= v(t)/L\,\tan\phi(t), \end{aligned} \tag{1}$$

where (x, y) are the coordinates for the point of application of the force of traction on the vehicle; θ is the heading of the vehicle on the plane on which it is moving; v is its speed; L is the distance between the front and back axles and the variable ϕ is the direction of the driving wheels with respect to the vehicle heading θ (see Fig. 2). Obviously, (v, ϕ) are the robot control variables and (x, y, θ) are its state variables. The discrete version of (1) is:

$$\begin{aligned} x_{k+1} &= x_k + v_k\cos\theta_k, \\ y_{k+1} &= y_k + v_k\sin\theta_k, \\ \theta_{k+1} &= \theta_k + v_k/L\,\tan\phi_k, \\ |\phi_k| &< \phi_{\max}, \end{aligned} \tag{2}$$

where $\phi_{\max}$ is the maximum angle that can be applied to the direction of the driving wheels.

In the case of automatic parking, there are two behavior indexes of interest: J_1 and J_2. These two indexes quantify the goal that the robot should park in the final position $(x_\mathrm{d}, y_\mathrm{d})$ and the goal that the robot should park in line with the parking space direction, θ_d, respectively. Hence:

$$\begin{aligned} J_1 &= \tfrac{1}{2}\left[(x - x_\mathrm{d})^2 + (y - y_\mathrm{d})^2\right], \\ J_2 &= \tfrac{1}{2}(\theta - \theta_\mathrm{d})^2 . \end{aligned} \tag{3}$$

Supposing that the vehicle maneuvers at constant speed, the other available control variable, ϕ, should minimize both indexes:

$$\dot{\phi}(t) = -\mu_1\frac{\partial J_1}{\partial\phi} - \mu_2\frac{\partial J_2}{\partial\phi}, \tag{4}$$

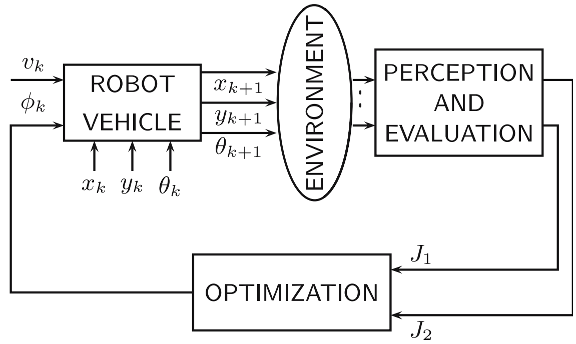

Fig. 1 Conceptual diagram of the biomimetic model

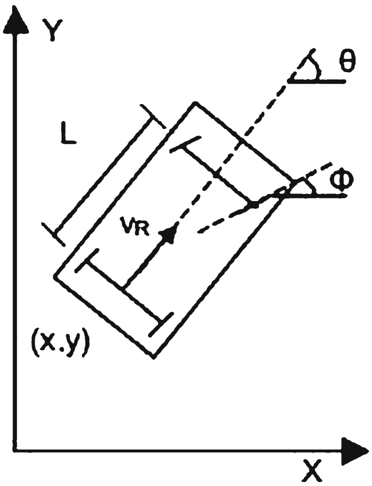

Fig. 2 Configuration of the car-like vehicle

where μ_1 and μ_2 weight the importance of each goal. The discrete version of (4) is:

$$\phi_{k+1} = \phi_k - \mu_1 \left.\frac{\partial J_1}{\partial \phi}\right|_{\phi(k)} - \mu_2 \left.\frac{\partial J_2}{\partial \phi}\right|_{\phi(k)}. \quad (5)$$

Equations (4) and (5) raise an important practical problem, which is what we might refer to as the relationship between the *distal* sensory information scale (given by the gradients or changes of the behavior indexes ΔJ_1 and ΔJ_2) and the *proximal* actions scale (given by the gradients of the control actions $\Delta\phi_k = \phi_k - \phi_{k-1}$ or $\Delta\phi_{k+1} = \phi_{k+1} - \phi_k$).

A practical way of solving this scale problem is to establish a tabular relationship between the distal levels and the proximal levels. In particular, the range of robot actions (speed v and steering wheel turn ϕ) can be distributed on a discrete scale of values. In view of the practical importance of the distal/proximal relationship (which in living beings takes a lot of learning), one open line of research is to develop flexible diagrams to quantify this relationship, where fuzzy linguistic variables or even genetic algorithms could play a role in adequate tuning. In this paper, however, we have not addressed this problem and have used a range of just three steering wheel action values. The action values are expressed as angular speeds of the steering wheel: $+10°, 0°, -10° s^{-1}$, depending on whether the ratios of the distal and proximal gradients are positive, zero or negative, respectively. As we have used a control time cycle of 100 ms, the actual steering wheel turns applied to the vehicle have been $+1°, 0°, -1°$, respectively. We have achieved good results with this small scale of actions.

3 Multi-objective optimization

Set out in the terms described in the preceding section, automatic parking can be considered as a standard multi-criteria optimization problem, where the agent control actions (in this case, the direction of the steering wheel, because the robot is moving at constant speed) should simultaneously minimize the two indexes that appear in expressions (3) (4) and (5). As any driver will have found in practice, the dynamic coordination of these indexes in a parking maneuver is an extremely complex control problem, as, in non-holonomic vehicles, any slip in the combination of the actions suggested by the approach and heading indexes may be disastrous for the parking maneuver.

Apart from the fact that our working methods including the one we are concerned with here, are based on the strategic goal of minimizing the inherent rigidity of model-based control methods and, consequently, our working methods are motivated by an exploration of the potential of bioinspired heuristic procedures, model-based methods do not appear to be able to successfully tackle the simultaneous optimization of goals J_1 and J_2 anyway. Thus, taking the simplest case, where the magnitude of the steering wheel control actions, ϕ, is small, we have that:

$$\dot{\theta}(t) = \frac{v(t)}{L} \tan\phi(t) \approx \frac{v(t)}{L}\phi(t). \quad (6)$$

In this case, the optimization of J_2 can be assured by a simple PD control:

$$\phi^2(t) = K_\mathrm{p}\left[\theta(t) - \theta_\mathrm{d}\right] + K_\mathrm{d}\,\dot{\theta}(t), \quad (7)$$

where the superindex 2 refers to the fact that it is a control path aimed at minimizing index J_2, and in which θ_d is the desired or set point vehicle's orientation, as computed by a planner module of the robot car controller. Note also in (7) that K_p and K_d are the usual gains of the standard PD control algorithm. After substituting the control law (7) in (6), we would get:

$$\dot{\theta}(t) = \frac{v(t)}{L} K_\mathrm{p}\left[\theta(t) - \theta_\mathrm{d}\right] + \frac{v(t)}{L} K_\mathrm{d}\,\dot{\theta}(t) \quad (8)$$

and, hence, the vehicle would rapidly converge towards the desired final heading that the parameters of the PD control gains are correctly tuned. As far as the minimization of the other index is concerned, let us first express this index as an explicit function of the control variable $\phi(t)$:

$$J_1 = \frac{1}{2}\left[\int_{t_0}^{t_\mathrm{f}} \cos\left[\frac{v(t)}{L}\phi(t)\right] t - x_\mathrm{d}\right]^2 + \frac{1}{2}\left[\int_{t_0}^{t_\mathrm{f}} \sin\left[\frac{v(t)}{L}\phi(t)\right] t - y_\mathrm{d}\right]^2. \quad (9)$$

Although more complex than the heading index, due to its non-linear dependence on the control variable, the optimization of J_1 can be tackled using analytical procedures to find a control path $\phi^1(t)$ that makes J_1 tend to zero. But the key question is the dynamic coordination of the two control paths $\phi^1(t)$, which minimizes J_1, and $\phi^2(t)$, which minimizes J_2. That is, the key is to find the coordination function:

$$\phi(t) = f\left[\phi^1(t), \phi^2(t)\right] \quad (10)$$

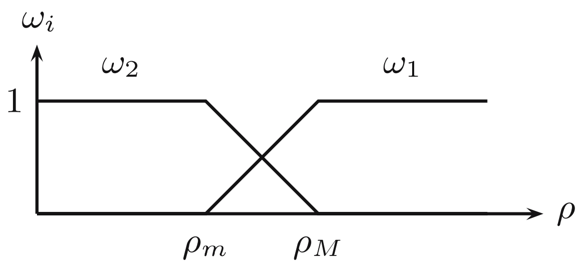

Fig. 3 Basic goal coordination rules

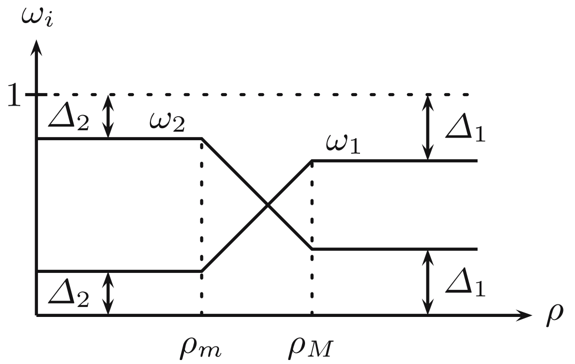

Fig. 4 Rules with region of attraction influence

that assures the correct execution of the parking maneuver. A simple approach to the problem of goal coordination would be to formulate it in linear terms as:

$$\phi(t) = \omega_1(t)\phi^1(t) + \omega_2(t)\phi^2(t), \tag{11}$$

where $\omega_1(t)$ and $\omega_2(t)$ dynamically designate the instantaneous weight to be assigned to each goal.

The difficulty with the problem of dynamically coordinating contradictory goals, at least in the automatic parking problem, lies in the fact that the weight coefficients depend primarily on the relative positions of the robot ($x(t)$, $y(t)$, $\theta(t)$) and the parking space as opposed to time. This means that the analytical approach to optimizing the weight coefficients $\omega_1(t)$ and $\omega_2(t)$ further complicates the already complex analytical processing required to output control laws $\phi^1(t)$ and $\phi^2(t)$ separately.

Finally, appealing to the reader's intuition, we would like to stress that the dynamic coordination of the optimization paths of the two indexes J_1 and J_2 is a hypersensitive issue. Just imagine a driver giving more priority to heading than to approach instructions when he is far away from the parking space or doing just the opposite, and giving more priority to the approach goal, when he is near to the space: the respective paths would be disastrous. Note, also, the additional difficulty of having to formalize the linguistic terms near and far.

In sum, we believe that a problem that is as complex to address using analytical procedures as the dynamic coordination of opposing goals in automatic parking is a natural candidate for being tackled by evolutionary heuristic techniques, which is what the remainder of the paper deals with.

4 Goal coordination with genetic algorithms

Let us consider how a human driver performs the automatic parking operation to see if we can find any clues as to how to define the appropriate parameters and variables so that a genetic algorithm can solve the coordination problem.

Very briefly, we find that parking operations obey the following two basic rules:

1. If the vehicle is *far* from the target, then priority should be given to the approach goal (index J_1).
2. If the vehicle is *near* to the target, then priority should be given to the heading goal (index J_2).

Note the ambiguity of the linguistic terms far and near. If we take the simplest case of linear coordination

$$\phi(t) = \omega_1(t)\phi^1(t) + \omega_2(t)\phi^2(t), \tag{12}$$

then these two basic rules of parking goal coordination can be simply formalized as shown in Fig. 3, which resembles the standard membership functions used in fuzzy subset theory.

This graph, where ρ is the distance of the vehicle from the target or parking space, is straightforward to interpret. Supposing that the curves of the coefficients are symmetric, the two parameters that determine the coordination dynamics are ρ_m and ρ_M. Thus, it would now be feasible to think of applying a genetic algorithm to optimize the choice of these parameters for dynamically coordinating the goals, i.e.,

$$\omega_1 = f_1(\rho_m, \rho_M); \quad \omega_2 = f_2(\rho_m, \rho_M). \tag{13}$$

A practical problem here is that undesired situations may materialize, if the genetic algorithm search is confined to the parameters ρ_m and ρ_M, which somehow determine the concepts of far and near in the implementation of the two basic parking rules. Specifically, once the vehicle has entered the region of attraction of coefficient ω_2 (i.e., when it is *near* to the target), it will park parallel to, but not inside the space, if it reached this region by "connecting" with the straight line defined by the prolongation of the direction of the space. Such situations could be ruled out by allowing both goals to have some influence in the region of attraction of near and far, as shown in Fig. 4.

In this case, the genetic algorithm would be responsible for optimizing the additional parameters Δ_1 and Δ_2, apart from ρ_m and ρ_M.

Figure 5 shows a conceptual diagram of the whole system, in which there are two players (controllers), each aiming at minimizing a particular objective. Player 1 controls the vehicle steering wheel to minimize a distance-based performance index, J_1, and, similarly, player 2 controls the steering wheel in order to minimize a direction-based performance index J_2. The global coordination is performed by means of a fuzzy-like meta-heuristic, which, in turn, is optimized by a genetic algorithm.

Looking again at how humans park, we find that one very efficient maneuver, provided there are no obstacles, is to approach, in almost any direction, an area close to the position and direction of the space, as shown in Fig. 6. As of then

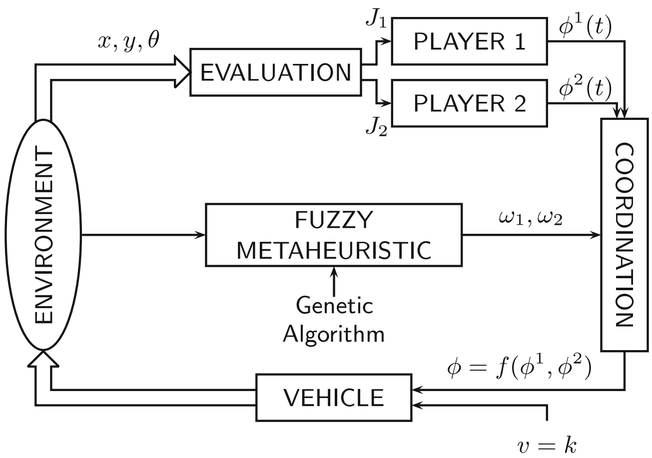

Fig. 5 Conceptual scheme of the global system in which the fuzzy-like metaheuristic is evolved to optimize the coordination of both controllers

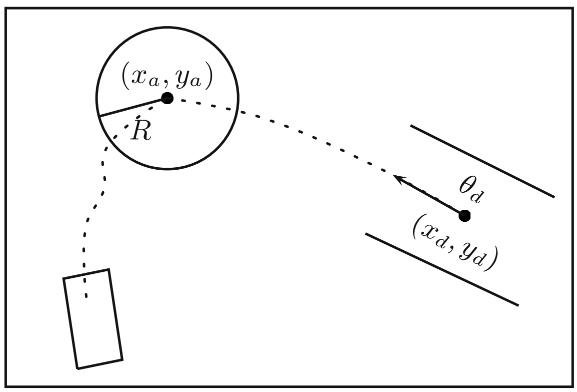

Fig. 6 Illustration of the third method described in the text

priority, albeit not absolute, is given to heading and, when the vehicle is aligned with the space, the approach goal takes maximum, but again not exclusive, priority.

Let us take a qualitative look at the execution of this maneuver.

Phase 1. Transfer region approach. This phase can be performed, in principle, without concern for the heading goal J_2. However, the position of the vehicle in this transfer region should be as aligned as possible with respect to the direction of the space.

Phase 2. Alignment with the direction of the space. Once it is positioned in the transfer or subtarget region, the vehicle gives maximum, but not exclusive, priority to the heading goal. Obviously, the more aligned it is, the smoother the maneuver will be.

Phase 3. Parking space approach according to the desired heading. After alignment with the direction of the space, the vehicle's only concern will be to reduce its distance to the space (maximum priority of index J_1). To prevent possible losses of alignment, goal J_2 should retain some, albeit a very weak, influence.

5 Experimental results for parking without obstacles

The experiments were conducted using the University of Sheffield's Genetic Algorithm Toolbox for Matlab [18]. For all cases, a 20-bit resolution binary coding was used for the parameters processed; the parameter ranges depend on the variables to be optimized.

The stochastic universal sampling method was used to select individuals. The crossover probability used is 0.7; the mutation probability is set proportionally to the size of the population, and is never over 0.02. Additionally, elitism from generation to generation is used. Hence, 90% of the individuals of each new population are created by means of the selection and crossover operators and 10% of the best individuals of each generation are added directly to the new population.

Quality is determined by rewarding the individuals that simultaneously minimize the two indexes J_1 and J_2, i.e., the closer an individual is to the position and direction defined as the target, at the end of the path, the better this individual is. Additionally, individuals who manage to reach the target along a shorter path are also considered better, although the weighting of this factor is lower.

A measure of the number of changes of vehicle heading, i.e., the number of operations effected on the steering wheel along the path causing the vehicle to modify its heading instructions (brusque changes from right to left and vice versa) was also used in some experiments. However, this index appears to be somehow included in the shortest path index, and was, therefore, not used in the final experiments.

The experiments were actually designed by defining a set of initial and final vehicle position and heading pairs that would cover the different relative situations between the source and target. Each individual generated in the evolutionary process was simulated with these initial and final conditions to thus determine its problem-solving quality.

As explained in the previous section, the first parking method is based on evolving parameters ρ_m and ρ_M (Fig. 3) that denote the distance to the target and determine whether the position index ($\rho > \rho_M$), the heading index ($\rho < \rho_m$) or a combination of the two ($\rho_m \leq \rho \leq \rho_M$) should be used.

For this method, the parameters ρ_m and ρ_M were initially left to evolve freely, which encouraged situations where the fittest individuals had a ρ_m greater than the distance at which they were positioned at the start. In other words, only the heading index was taken into account in parking, which led to executions such as the one shown in Fig. 7a. The area denoted by the internal circumference represents the region in which $\rho < \rho_m$, and the external circumference starting at $(-10, -5)$ and $(-10, 35)$ matches the region $\rho > \rho_M$.

After detecting this erroneous behavior, the parameters ρ_m and ρ_M were forced to evolve so that the initial distance between the vehicle and the target fell within the subintervals they determine. Figure 7b shows the path followed by one of the individuals included in this procedure.

Apart from evolving parameters ρ_m and ρ_M, the second parking strategy also considers Δ_1 and Δ_2 (Fig. 4), which,

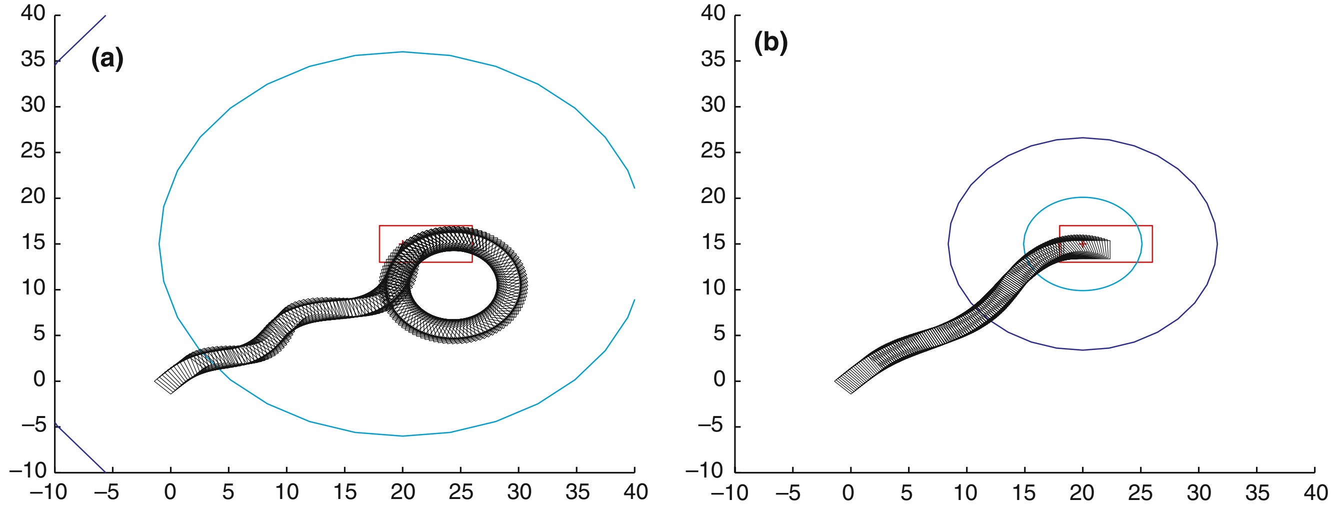

Fig. 7 Parking maneuvers evolving ρ_m and ρ_M: **a** Erroneous parking, **b** successful parking

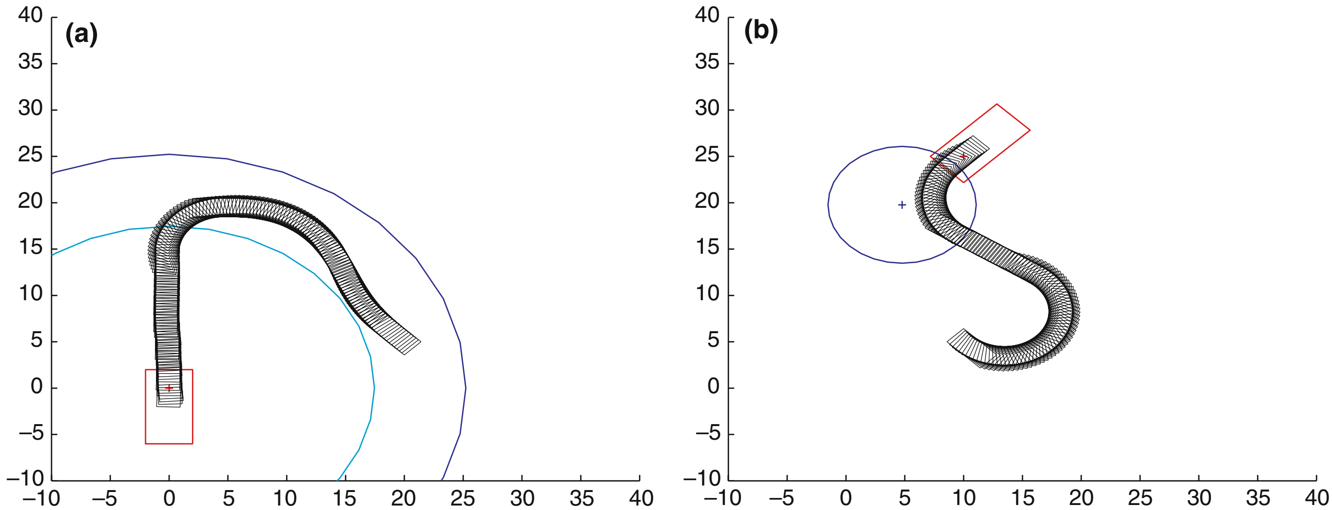

Fig. 8 **a** Parking by evolving ρ_m, ρ_M, Δ_1 and Δ_2, **b** parking with subtarget in a tricky situation

as mentioned above, means that both the position and the heading indexes are taken into account at all times.

The results obtained are more or less equivalent to the results of the first parking procedure, although, as four parameters had to be tuned in this case, the number of individuals in each population had to be doubled (80 as compared to 40 used in the experiments with the first procedure) and convergence to the best solution was also much slower (at least 100 generations were needed to reach the first correct solution). Figure 8a shows the path achieved by the fittest individual.

In the third and last parking method, the parameters evolved are distances from the goal to the subtarget and radius of the subtarget region depicted in Fig. 6. The parking paths using this method are much better and more natural than those achieved with the other two. As an illustration Fig. 8b shows a relatively tricky situation that calls for a complicated and successful parking maneuver.

6 Parking with collision avoidance

We address now the problem of automatically parking the vehicle with the additional difficulty created by the presence of a priori unknown obstacles, so that the car controller has to autonomously perform, in real time, two different tasks: parking and collision avoidance.

As regards the general task of collision avoidance, we introduced in [7] a generalization of the well-known artificial potential field theory [19]. More specifically, we added to the customary normal orientation the tangential orientation, making it possible for the robot to perform smoother and more efficient trajectories for collision avoidance. We suppose that the car-like robot only knows the space parking coordinates (x_d, y_d) and its direction θ_d. To perform the parking maneuver, the car is equipped exclusively with four sets of ultrasound sensors placed at each of its four edges.

Roughly speaking, the normal orientation represents the objective or goal of flying from the obstacle, whereas the two tangential orientations, right and left, are meant for the vehicle to round up the obstacles, either by following the right or the left direction. Obviously, the other important orientation for the vehicle is the one that provides the trajectory towards the parking space.

It is straightforward to show [8] that the normal orientation is given by

$$\phi^{\mathrm{n}} = \tan^{-1} \frac{\partial U_{\mathrm{r}}(x, y)/\partial y}{\partial U_{\mathrm{r}}(x, y)/\partial x}, \tag{14}$$

where $U_{\mathrm{r}}(x, y)$ is the repulsive potential field:

$$U_{\mathrm{r}}(x, y) = \begin{cases} \frac{1}{2} K_{\mathrm{r}} \left[\frac{1}{\rho(x,y)} - \frac{1}{\rho_0} \right]^2; & \rho \geq \rho_0, \\ 0; & \rho < \rho_0, \end{cases} \tag{15}$$

in which $\rho(x, y)$ is the shortest distance from the car to the nearest obstacle and ρ_0 is a threshold distance beyond which the obstacles do not influence the car movements. The tangential orientations are

$$\phi^{\tau} = \tan^{-1} \frac{\pm \partial U_{\mathrm{r}}(x, y)/\partial x}{\mp \partial U_{\mathrm{r}}(x, y)/\partial y}, \tag{16}$$

where the opposing signs correspond to the right and left tangential orientations.

One important drawback of expressions (14) and (16) is that we need to compute, at each vehicle position, the repulsive potential field to obtain the two orientations for collision avoidance. In [5] we applied a simpler, approximate method for the computation of both orientations ϕ^{n} and ϕ^{τ}, by means of a ring of ultrasonic sensors, without using the repulsive potential, so that we shall suppose that the normal and tangent orientations are available at each robot position without computing the exact potential field function.

Summarizing, we have now at each instant all the necessary orientations to guide the car towards the parking space without colliding with the existing obstacles: (1) orientation ϕ^1 that optimizes the performance index J_1 which drives the robot towards the final position $(x_{\mathrm{d}}, y_{\mathrm{d}})$; (2) orientation ϕ^2 that optimizes the performance index J_2 responsible of making the car to park in line with the parking space θ_{d}; (3) the normal orientation ϕ^{n} that precludes the robot from being too close to the obstacles and the tangential orientation ϕ^{τ} that constraints the robot to follow the optimum trajectory to circumvect the obstacles.

The next and crucial step is to coordinate all these competitive and opposing control orientations:

$$\phi^{\mathrm{d}}(t) = J \left[\phi^1(t), \phi^2(t), \phi^{\mathrm{n}}(t), \phi^{\tau}(t) \right], \tag{17}$$

where by $\phi^{\mathrm{d}}(t)$ we are referring to the desired and final orientation to be applied to the vehicle. The simplest, although by no means trivial, coordination of the competitive orientations is a linear one:

$$\phi^{\mathrm{d}}(t) = \omega_1(t)\phi^1(t) + \omega_2(t)\phi^2(t) + \omega_{\mathrm{n}}(t)\phi^{\mathrm{n}}(t) + \omega_{\tau}(t)\phi^{\tau}(t). \tag{18}$$

Therefore, at each instant the car controller computes its final orientation $\phi^{\mathrm{d}}(t)$ as a function of the four basic angles, so that the key point is to select the suitable set of instantaneous weights $\omega_1(t)$, $\omega_2(t)$, $\omega_{\mathrm{n}}(t)$ and $\omega_{\tau}(t)$. This is a rather difficult optimization problem, due to the dynamical nature of the four competitive objectives, as they depend on the relative position and shape of the existing obstacles, which are highly uncertain and a priori unknown by the designer of the robot controller.

Furthermore, as discussed in the case of parking the car without obstacles, direct genetic encoding of the four weighting coefficients is not feasible, as they are dynamic and time-varying parameters that depend, at all times, of the relative position and heading of the vehicle with respect to the unknown obstacles and, as well, the parking space.

To make more manageable this multi-criteria optimization problem, let us first merge the two orientations $\phi^1(t)$ and $\phi^2(t)$ into a single goal orientation:

$$\phi^{\mathrm{g}}(t) = f \left[\phi^1(t), \phi^2(t) \right], \tag{19}$$

so that the final car orientation can be rewritten as

$$\phi^{\mathrm{d}}(t) = \omega_{\mathrm{g}}(t)\phi^{\mathrm{g}}(t) + \omega_{\mathrm{n}}(t)\phi^{\mathrm{n}}(t) + \omega_{\tau}(t)\phi^{\tau}(t). \tag{20}$$

Now, let us consider the following rule-based reasoning.

1. *If* the nearest obstacle is *very close* to the robot, *then* give maximum priority to the normal orientation.
2. *If* the nearest obstacle is at an *intermediate distance*, *then* the tangential orientation has maximum priority.
3. *If* the nearest obstacle is *far* from the robot, *then* give maximum priority to the goal orientation.

This reasoning process, expressed as linguistic rules, can be implemented either by means of a fuzzy logic-based controller or by a process based on the dynamical coordination of multiple performance indices [5,8]. More specifically, these rules can be formalized, as far as the numerical values of the weights ω_{g}, ω_{n} and ω_{τ} are concerned, in the way shown in Fig. 9, where we have represented the distribution of the coordination parameters as a function of the distance of the car to the nearest obstacle. Note the absolute similarity of this graphic with the one corresponding to the case of parking without obstacles.

Figure 9 is self-explanatory. Thus, d_m is a critical distance in the sense that it acts as a security threshold beyond which the robot behavior is dominated by the objective of flying out

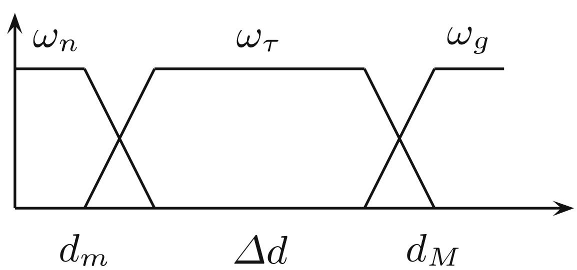

Fig. 9 Distribution of the coordination parameters ω_{g}, ω_{n} and ω_{τ} as a function of distance d between the vehicle and the nearest obstacle

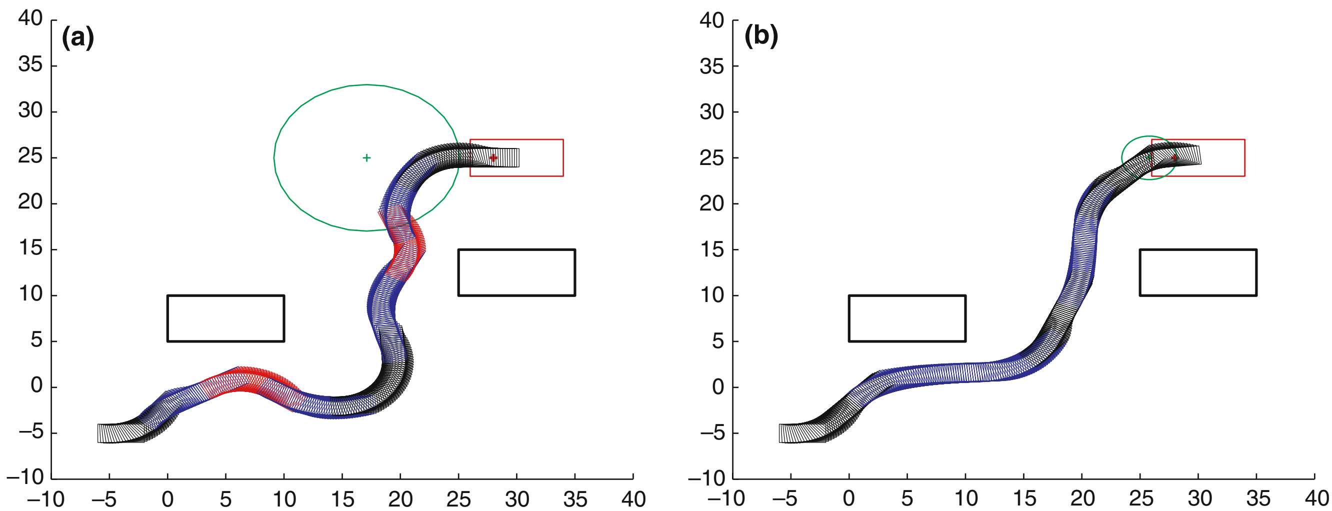

Fig. 10 Paths achieved by the fittest individual. Left: 20th generation, $d_m = 5.113$, $d_M = 2.068$, $\Delta d = 1.702$, $d'_m = 7.982$, $d'_M = 18.868$ and Right: 60th generation, $d_m = 2.559$, $d_M = 5.998$, $\Delta d = 2.746$, $d'_m = 2.199$, $d'_M = 4.576$

the nearest obstacle by following the normal orientation. In a similar way, d_M is another crucial distance beyond which the robot behavior is dominated by the objective of going straightforwardly to the goal. Finally the third parameter Δd determines the region of influence of the tangential navigation; i.e., when the robot's main objective is to round up the nearest obstacle by following the tangent orientation.

Thanks to the introduction of this dependence on distance of the coordination weights $\omega_{\mathrm{g}}(t)$, $\omega_{\mathrm{n}}(t)$ and $\omega_{\tau}(t)$ the search space has been dramatically simplified and reduced. Now, the multi-objective optimization problem exclusively depends on the three critical distances d_m, Δd and d_M, which unlike the parameters $\omega_{\mathrm{g}}(t)$, $\omega_{\mathrm{n}}(t)$ and $\omega_{\tau}(t)$ are not time-dependent, making the optimum search one order of magnitude simpler, as compared with the direct optimization based on the coordination weights.

Obviously, as regards the coordination of the two subgoal orientations $\phi^1(t)$ and $\phi^2(t)$, embedded in the global goal orientation ϕ^{g}, we make use of the results obtained previously on automatic parking without obstacles.

In summary, there are two embedded optimization processes. The first one concerns the computation of the goal orientation $\phi^{\mathrm{g}}(t)$ as a result of the coordination of the two subgoals for the parking task – i.e., approximation orientation $\phi^1(t)$ and alignment orientation $\phi^2(t)$. Once $\phi^{\mathrm{g}}(t)$ has been obtained, the next optimization process affects the coordination of this goal orientation and the two orientations for collision avoidance – i.e., normal and tangential orientations. The optimum search is still harder to be solved by means of analytical methods or, alternatively, by means of supervised, gradient-based techniques than the previous case of parking without obstacles, due to the highly uncertain and a priori unknown spatial distribution and shape of the obstacles. Therefore, we have applied a genetic algorithm to solve this twofold multi-objective optimization problem.

The five parameters evolved were d_m, d_M, Δd, d'_m and d'_M. The number of individuals in each population was 40 and the convergence to the best solutions was reached in the 60th generation. Figure 10 shows the path achieved by the fittest individual of two different generations.

7 Conclusions

We have presented a solution for conventional non-holonomic vehicle automatic parking. The proposed solution is based on a biomimetic approach that can be used to design extremely simple and robust autonomous robot control systems, as the designer has to inject only the robotic system goals. This approach means, therefore, that the use of dynamic and kinematic robot models and even the aprioristic formal descriptions of their working environments can be ignored. Tailoring this approach to the case of automatic parking leads to a problem of dynamic antagonistic goal coordination, which, owing to its complexity and instability, cannot be dealt with using analytical methods. Therefore, this multi-objective optimization problem has been formalized to be tackled by genetic algorithms. Three automatic parking strategies have been designed, and the results obtained using genetic algorithms have been presented.

Finally, we have also considered a more complex situation in which a priori unknown obstacles are present in the parking place. By adding a new objective (i.e., collision avoidance) to the multi-criteria performance function the vehicle is able to park without colliding with the existing obstacles.

Acknowledgements This work has been partially funded by the Spanish Ministry of Science and Technology, project DPI2002-04064-C05-05.

References

1. Deb K (2001) Multi-objective optimization using evolutionary algorithms. Wiley, New York
2. Fonseca CM, Fleming PJ (1993) Genetic algorithms for multiobjective optimization: formulation, discussion and generalization. In: Proceedings of the 5th International Conference on genetic algorithms, Morgan Kaufmann, San Francisco, pp. 416–423
3. Coello CA, Van Veldhuizen DA, Laumont GB (2003) Evolutionary algorithms for solving multi-objective problems. Kluwer, New York
4. Coello CA (2003) Special issue on evolutionary multiobjective optimization. IEEE Trans Evol Comput 7(2):97–99
5. Maravall D, De Lope J (2003) A bio-inspired robotic mechanism for autonomous locomotion in unconventional environments. In: Zhou C, Maravall D, Ruan D (eds) Autonomous robotic systems: soft computing and hard computing methodologies and applications. Physica-Verlag, Springer, Berlin Heidelberg New York, pp. 263–292
6. De Lope J, Maravall D (2005) A biomimetic approach to the stability of biped robots. In: Armada M, González P (eds) Climbing and walking robots. Springer, Berlin Heidelberg New York, pp. 593–600
7. De Lope J, Maravall D (2003) Integration of reactive utilitarian navigation and topological modeling. In: Zhou C, Maravall D, Ruan D (eds) Autonomous robotic systems: soft computing and hard computing methodologies and applications. Physica-Verlag, Springer, Berlin Heidelberg New York, pp. 103–139
8. Maravall D, De Lope J (2002) Integration of artificial potential field theory and sensory-based search in autonomous navigation. In: Proceedings of the 15th IFAC World Congress, International federation of automatic control
9. Maravall D, De Lope J, Patricio MA (2004) Competitive goal coordination in automatic parking. In: 1st European workshop on evolutionary algorithms in stochastic and dynamic environments, EVOSTOC-2004. LNCS 3005, Springer, Berlin Heidelberg New york, pp. 537–548
10. Zadeh L (1965) Fuzzy sets. Inform Control 8:338–353
11. Zadeh L (1968) Fuzzy algorithms. Inform Control 12:94–102
12. Sugeno M, Nishida M (1985) Fuzzy control of a model car. Fuzzy Sets Syst 16:100–110
13. Sugeno M, Murofushi T (1989) Fuzzy algorithm control of a model car by oral instructions. Fuzzy Sets Syst 32:207–219
14. Sugeno M, Murakami K (1985) An experimental study of fuzzy parking control using a model car. In: Sugeno M (eds) Industrial applications of fuzzy control. North-Holland, Amsterdom, pp. 125–135
15. Tanaka K (1995) Design of model-based controllers using Lyapunov's stability approach and its applications to trajectory stabilization of a model car. In: Nguyeng HT et al (eds) Theoretical Aspects of Fuzzy Control. Wiley, New York, pp. 31–50
16. Laugier C, Fraichard Th, Garnier Ph, Paromtchik IE, Scheuer A (1999) Sensor-based control architecture for a car-like vehicle. Auton Robots 6(2):165–185
17. Petti S, Fraichard Th (2005) Safe motion planning in dynamic environments. In: Proceedings of the IEEE-RSJ International Conference on intelligent robots and systems, Edmonton, Canada
18. Chipperfield A, Fleming P, Pohlheim H, Fonseca C (1994) Genetic algorithm toolbox for matlab, Department of Automatic Control and Systems Engineering, University of Sheffield
19. Khatib O (1986) Real-time obstacle avoidance for manipulators and mobile robots, Int J Robot Res 5(1):90–98