Robotics and Computer-Integrated Manufacturing 27 (2011) 558–568

Contents lists available at ScienceDirect

Robotics and Computer-Integrated Manufacturing

journal homepage: www.elsevier.com/locate/rcim

Switching control approach for stable navigation of mobile robots in unknown environments

Juan Marcos Toibero [a,*], Flavio Roberti [a,b], Ricardo Carelli [a,b], Paolo Fiorini [c]

[a] Instituto de Automática (INAUT), Universidad Nacional de San Juan, Av. San Martin Oeste 1109, 5400 San Juan, Argentina
[b] Consejo Nacional de Investigaciones Científicas y Técnicas, Argentina
[c] Dipartimento di Informatica, Università degli Studi di Verona Strade Le Grazie 15, 37134 Verona, Italy

ARTICLE INFO

Article history:
Received 23 February 2009
Received in revised form
8 June 2010
Accepted 1 October 2010

Keywords:
Wheeled mobile robots
Nonlinear control
Multiple Lyapunov functions
Switched systems
Robot navigation
Stability

ABSTRACT

This paper presents a stable switching control strategy for the parking problem of non-holonomic mobile robots. First, it is proposed a positioning-orientation switching controller for the parking problem. With this strategy robot backwards motions are avoided and the robot heading is always in the direction of the goal point facilitating the obstacle handling. Second, the avoidance of unexpected obstacles is considered in a reactive way by following the contour of the obstacles. Next, the stability of the switching parking/obstacle-avoider controller is analyzed showing stability under reasonable conditions. Finally, the good performance and the feasibility of this approach are shown through several experimental results.

1. Introduction

The problem of programming a mobile robot to move from one place to another is of course as old as the first mobile robot. In mobile robotics almost every task to solve deals with the problem of parking [1], which includes the ability to set a desired final orientation instead of the classical behavior "move-to-goal" in behavior-based architectures [2] in which only a final point is needed (positioning problem). This distinction adds complexity when working with unicycle-like mobile robots since its model has a non-holonomic constraint making impossible to design a continuous invariant control law that guarantees to reach a final posture in Cartesian coordinates [3]. Intuitively a non-holonomic constraint restricts a vehicle motion locally but not globally. For the kinematic unicycle, the non-holonomic restriction implies no sideways motion of a point on the wheel axis. Note that under this constraint, there is a feasible trajectory between any two postures. The price paid for free motion of an off-axis point is the lost of orientation control. Several works have been developed in this area; Fierro and Lewis [4] uses the dynamic model of the mobile robot and achieve the objective by means of neural networks, in [5,6,1] a change in the coordinates of the kinematic model has been introduced.

In this paper we present a solution for the parking problem based on the stable switching of a positioning controller and its complementation with an only-bearing controller that brings the robot the ability to set desired headings. The robot must reach the final posture $\begin{bmatrix} x_d & y_d & \theta_d \end{bmatrix}^T$ starting from any initial posture $\begin{bmatrix} x_i & y_i & \theta_i \end{bmatrix}^T$ as can be seen in Fig. 1. This approach takes advantage of the non-holonomic constraint of the unicycle-like mobile robots decomposing the robot movement in such a way that backward motions are avoided and the robot heading is always in the direction of the goal point. A recent switching approach to this problem is [7] where it is proposed a more complex switching system concluding stability at the goal point based on passivity concepts. However, in spite of the stability results, the performance shown in the experimental results of this paper is questionable.

Besides the parking problem, this paper is intended to cope with the avoidance of unexpected obstacles [8–11]. This problem itself has relevant importance since its solution has direct application in areas such as exploration of unknown environments [27] for map construction [12], rescue missions [13] or landmark finding [14]. Most of the reports on this subject are strongly based on path planning strategies where the robot recognizes its surroundings using cameras or laser range-finders. However, our approach is closer to reactive strategies such as in [11,26,29,32], since we propose to follow the obstacle's boundaries using a stable contour-following controller (CF) [15], with full independence on the obstacle shape.

The main contribution of this work is the full description of a new stable switching system composed by a parking controller and a CF controller. Different from related papers [11,29,32], our work not only

* Corresponding author. Tel.: +54 2644213303; fax: +54 2644213672.
E-mail address: mtoibero@inaut.unsj.edu.ar (J.M. Toibero).

0736-5845/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.rcim.2010.10.002

Fig. 1. Problem description.

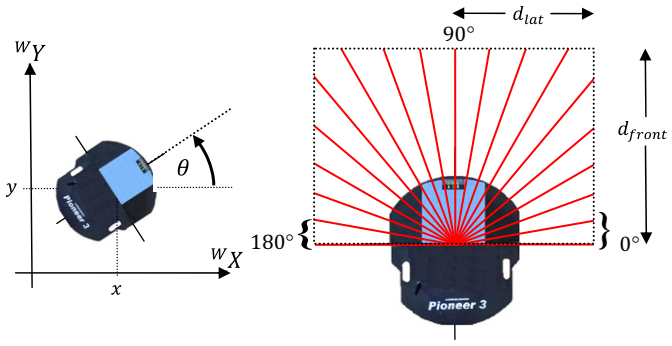

Fig. 2. Unicycle-like mobile robot and laser-range finder: curly brackets indicate beams used to estimate the obstacle contour angle.

deals with the switching rules, but also includes the autonomous navigation control algorithms with formal stability analysis. This way, the proposed switching rules guarantee both an appropriated path length (when compared with reactive classical algorithms) and also the stability of the overall control system considering multiple Lyapunov functions [16]. The proposed strategy allows the robot to handle real conditions, including confinement or traps situations in large-scale settings [18] and can be scaled for more complex navigation systems such as multi-robot systems [19].

The remainder of this paper is organized as follows: in Section 2 the well-known kinematics for the unicycle-like mobile robot are introduced as well as the laser range-finder employed in the experiments. Section 3 begins with a solution for the positioning problem with complete stability analysis and concludes with the design of the proposed switching parking controller. The addition of the obstacle-avoidance behavior is treated in Section 4. Then, in Section 5 it is considered the robot–environment interaction with the stability proposal and some preliminary laboratory experimental results. Next, in Section 6 experimental results in office settings are presented and analyzed. Finally in Section 6 the conclusions are stated.

2. Mobile robot

In this paper it is considered the wheeled mobile robot of unicycle type shown in Fig. 2, in which the state variables are x and y (the coordinates of the middle point of the front wheels axle) and θ (angle of the vehicle with the world X-axis [^wX]). A rear wheel turns freely and balances the rear end of the robot above the ground. The kinematics of the robot can be modeled by

$$\dot{\chi} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = f(\chi,u) = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \upsilon \\ \omega \end{bmatrix} \quad (1)$$

where $u = [\upsilon \quad \omega]^T$ is the control input vector: υ and ω are the forward and the angular robot velocity, respectively. The robot is equipped with a 181-beams laser range-finder. With reference to Fig. 2, the lateral beams from 0° to 15° (and from 165° to 180°) are used to estimate the robot/obstacle distance and the obstacle contour angle with respect to the global framework, while all beams are used to define a safety-zone, whose purpose is to detect possible robot-obstacle collisions. The guard-zone is rectangular and is defined by two parameters: the desired lateral d_{lat} and frontal d_{front} distances. The minimum lateral value for a Pioneer IIIDX is about 330 mm.

3. Parking problem

This section presents a switching controller approach to address the parking problem in Cartesian coordinates. This strategy is based on the consideration of asymptotically stable subsystems that solve specific navigation actions, namely the positioning control in Section 3.1 and the heading control in Section 3.2, and then designing a simple switching controller including both subsystems, presenting a solution for the parking problem and discussing stability at the switching times in Section 3.3.

3.1. Positioning control

Let us consider a continuous controller similar to [20], where it is guaranteed the achievement of a desired destination point $[x_d \quad y_d \quad \theta]^T$ without specifying the desired final heading θ_d (Fig. 3). We continue with the description of this controller by defining Cartesian errors as

$$\tilde{x} = x_d - x \quad (2.a)$$

$$\tilde{y} = y_d - y \quad (2.b)$$

and computing the control states: distance to the goal point d and heading error $\tilde{\theta}$:

$$d = \sqrt{\tilde{x}^2 + \tilde{y}^2} \quad (3.a)$$

$$\tilde{\theta} = \theta_{dP} - \theta = a\tan 2(\tilde{y},\tilde{x}) - \theta \quad (3.b)$$

where $\tilde{\theta}$ is the orientation error between the initial and the final point. Hence, the time-variation of these control states is ruled by

$$\dot{d} = -\upsilon\cos(\tilde{\theta}) \quad (4.a)$$

$$\dot{\tilde{\theta}} = \frac{\upsilon}{d}\sin(\tilde{\theta}) - \omega \quad (4.b)$$

where υ is the robot translational velocity and d is the distance between the initial and the target points.

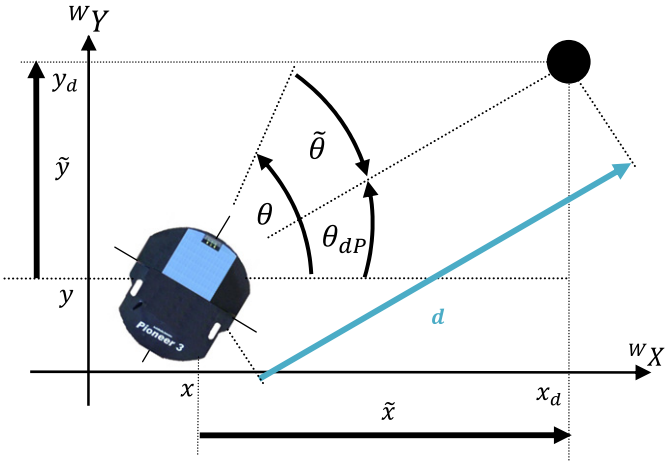

Fig. 3. Controller for positioning a non-holonomic mobile robot.

To prove the asymptotic convergence of the orientation error to zero it is proposed the following control law for the angular velocity:

$$\omega = \frac{\upsilon}{d}\sin(\tilde{\theta}) + K_{\tilde{\theta}}\tanh(k_\theta\tilde{\theta}) \tag{5}$$

In (5), the first term cannot be saturated, since it includes a division by d, whereas the second term saturates at the value of the constant $K_{\tilde{\theta}} > 0$, while $k_\theta > 0$ is other design constant chosen to increase the angular velocity for small errors. Note that due to the division by d, the value for (5) can be arbitrarily large and for this reason the robot must be stopped within a given neighborhood of the desired target point. The closed-loop equation, obtained by replacing (5) into (4.b) is

$$\dot{\tilde{\theta}} = -K_{\tilde{\theta}}\tanh(k_\theta\tilde{\theta}) \tag{6}$$

Next, it is considered the following Lyapunov function candidate and its time-derivative along trajectories:

$$V_{\tilde{\theta}} = \frac{\tilde{\theta}^2}{2} \tag{7}$$

$$\dot{V}_{\tilde{\theta}} = \tilde{\theta}\dot{\tilde{\theta}} = -K_{\tilde{\theta}}\tilde{\theta}\tanh(k_\theta\tilde{\theta}) < 0 \tag{8}$$

From these, it can be concluded the asymptotic stability of the angular error, i.e. the robot will head towards the goal point even with null linear velocity. It must now be proved that the same occurs for the distance d to the goal point. Then, it is analyzed the equilibrium point at $d=0$, defining the following control law for the linear velocity:

$$\upsilon = \frac{d}{1+|d|}\upsilon_{max}\cos(\tilde{\theta}) \tag{9}$$

where υ_{max} is the constant that indicates the desired maximum robot linear velocity. Then, the closed-loop equation becomes

$$\dot{d} = -\frac{d}{1+|d|}\upsilon_{max}\cos^2(\tilde{\theta}) \tag{10}$$

The stability of the equilibrium $d=0$ is considered with the Lyapunov candidate function (11.a) and its time derivative along trajectories (11.b):

$$V_P = \frac{d^2}{2} \tag{11.a}$$

$$\dot{V}_P = d\dot{d} = -d\upsilon\cos(\tilde{\theta}) \tag{11.b}$$

Then, it is straightforward that

$$\dot{V}_P = -\frac{d^2}{1+|d|}\upsilon_{max}\cos^2(\tilde{\theta}) < 0 \tag{12}$$

With the previous demonstration for $\tilde{\theta}(t)\rightarrow^{t\rightarrow\infty}0$, it is immediately concluded the asymptotic stability of this positioning controller, i.e. $d(t)\rightarrow^{t\rightarrow\infty}0$. With reference to Fig. 4 some simulation results can be seen, the considered initial point was $[x_i \quad y_i \quad \theta_i]^T = [3m \quad 3m \quad 45^o]^T$ whereas the target point was set up at $[x_d \quad y_d \quad \theta]^T = [0m \quad 0m \quad \theta]^T$. As can be noted, this controller allows robot backward motions and as the target point is closer the heading error computation is affected. In the extreme case of the robot just on the target point, this heading error cannot be determined due to (4.b).

3.2. Heading control

A heading controller can be obtained from the previously presented control actions considering $\upsilon=0$ in (5) and changing the desired heading position to a new (constant) orientation angle θ_{dH} (Fig. 5). This way, the analysis of the controller could be summarized as follows:

$$\tilde{\theta} = \theta_{dH} - \theta \tag{13}$$

$$\dot{\tilde{\theta}} = -\omega \tag{14}$$

In consequence, the control commands are

$$\upsilon = 0 \tag{15.a}$$

$$\omega = K_{\tilde{\theta}}\tanh(k_\theta\tilde{\theta}) \tag{15.b}$$

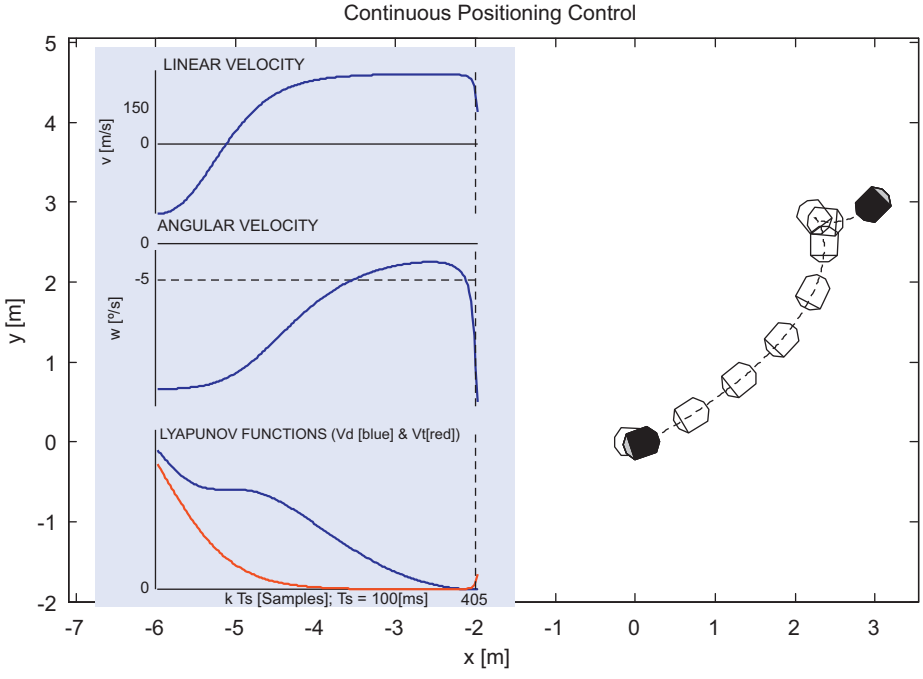

Fig. 4. Continuous parking controller without final orientation.

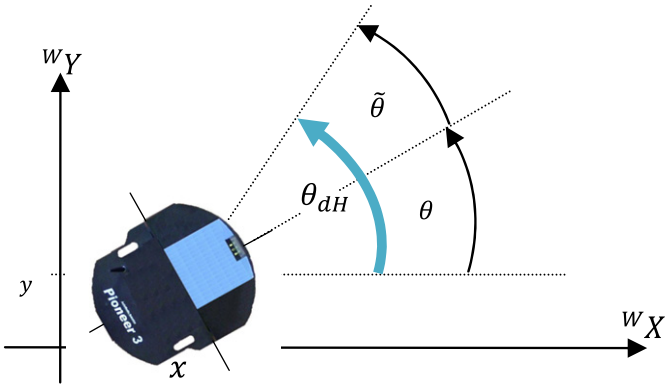

Fig. 5. Controller for angular position.

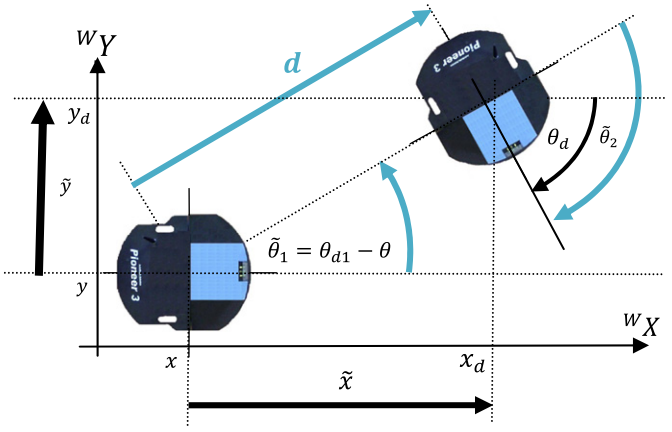

Fig. 6. Parking controller: angles involved.

noting that the angular velocity saturates at the value of the constant $K_{\tilde{\theta}} > 0$. This way it is obtained the same following closed-loop Eq. (6), and by considering a similar analysis as used in (7), it can be concluded (as expected) the control system asymptotic stability at the origin, i.e. $\tilde{\theta}(t) \rightarrow^{t\rightarrow\infty} 0$.

3.3. Switching parking controller

The theory of hybrid, switched control systems, i.e. systems that comprise a number of continuous subsystems and a discrete system that switches between them under certain logic rules, has received notable attention in the control theory community [16,17] since they provide a natural and convenient unified framework. In general, a hybrid switched system can be represented by the differential equation:

$$\dot{\chi}(t) = f'_{\sigma_i(t)}(\chi, t, u_{\sigma_i}) \tag{16}$$

where $f'_{\sigma_i(t)}$ is a collection of n distinct functions, $\sigma_i \in \{1,2,\ldots,n\}$ is the switching signal and $\mu_{\sigma i}$ explicitly denotes the dependence of the control input u on the switching signal σ_i, which is a discrete signal switching among discrete values in $\{1,2,\ldots,n\} \subseteq Z_+$. The value $\sigma_i(t)$ determines which function $f'_{\sigma_i}(\chi, u_{\sigma i})$ governs system behavior at time t.

Now we present our switching control system composed by the controllers described in previous sections. The switching signal is σ_1 and whenever $\sigma_1 = 1$, the controller for distance correction (positioning) is active, whereas when $\sigma_1 = 0$ or $\sigma_1 = 2$ the controller for angular position is active. The errors are redefined accordingly to Fig. 6.

In Fig. 6, $\tilde{\theta}_1 = \theta_{d1} - \theta$ is the initial orientation error, $\theta_{d1} = a\tan 2(\tilde{y}, \tilde{x})$ is the constant angle between both points of interest, θ is the robot heading and $\tilde{\theta}_2 = \theta_d - \theta_{d1}$ is the final orientation error. Next, based on the controllers described in Sections 3.1 and 3.2 we build the switching system of Fig. 7, aggregating an automaton to rule the switching between these controllers. Its logic is based on three simple stages:

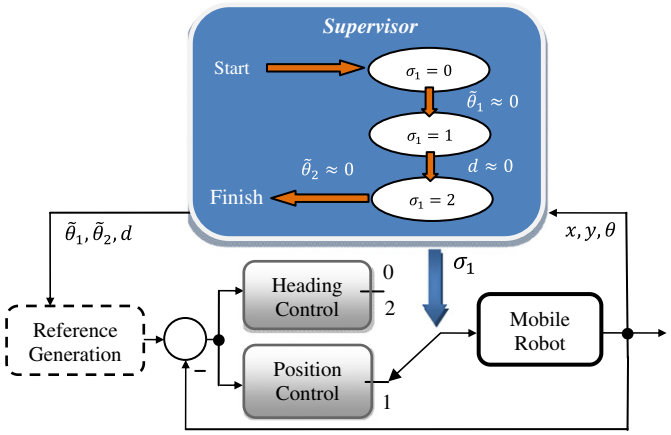

Fig. 7. Block diagram of the supervisor.

The robot heading is directed towards the destination point, i.e. $\tilde{\theta}_1 \rightarrow 0$.
The robot achieves the final point regardless of its orientation, i.e., $d \rightarrow 0$.
The robot adjusts its orientation towards the desired final heading θ_d, i.e. $\tilde{\theta}_2 \rightarrow 0$.

Hence, by switching according to this logic the robot goes straight to the target point activating the *heading controller* before moving towards the target point and after reaching it. In Fig. 7 the *Position Control* implements the control actions: Eq. (9) for the robot translational velocity and Eq. (15.b) for the robot angular velocity. This strategy preserves the asymptotic stability property if it is assumed that $\theta_d = const.$ in (3.b). This assumption is logical since the robot is already orientated towards the goal point, and, at the same time, avoids the division by d in the robot angular velocity.

3.3.1. Stability considerations at switching times

It is well-known that a switched system is stable if all individual subsystems are stable and the switching is sufficiently slow, so as to allow the transient effects to dissipate after each switch [16]. These facts are based most on the stability derivations from the theory of impulse effect systems [21,22] and randomly switched systems [28]. In our case, as each controller is stable and has of all the time necessary to achieve its control objectives, the supervisor can switch among the above mentioned controllers without affecting the stability of the system. In Fig. 8, it is shown the performance of our switching parking controller in order to allow a comparison with the conventional continuous controller similar to [20] of Fig. 7. The main difference is in the possibility to choose the final orientation. Besides, avoiding robot backward motions two main advantages are achieved: (1) the robot is ever "looking at" the target point, which is a good strategy for robots with a unique laser range-finder in its front and (2) the robot will perform the optimal trajectory to the target point.

4. Switching parking control with obstacles

Given the robot at an initial position, it must arrive to the destination posture avoiding the obstacles in between. The

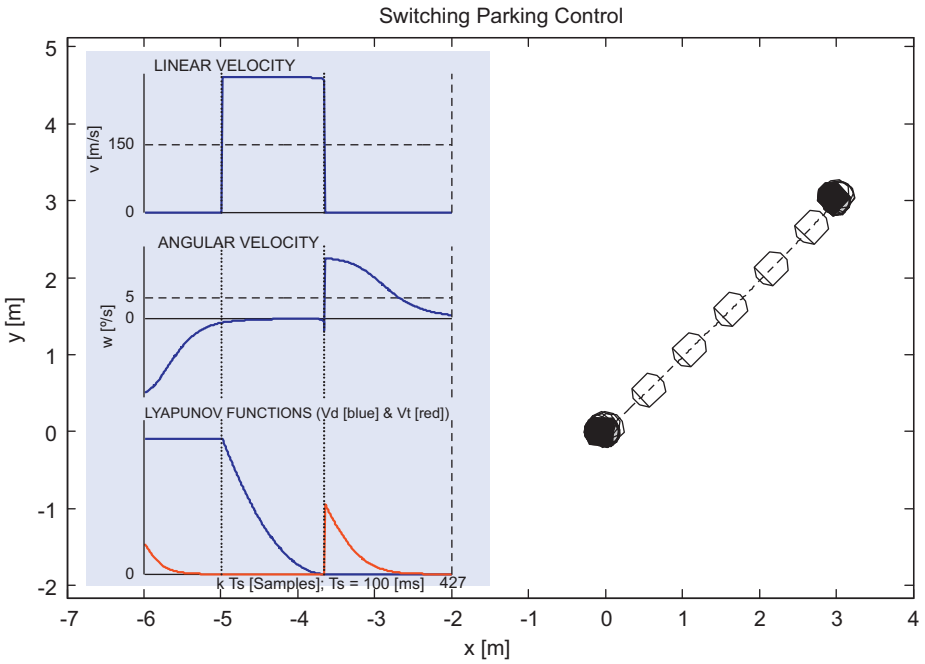

Fig. 8. Right: switching parking controller with final orientation.

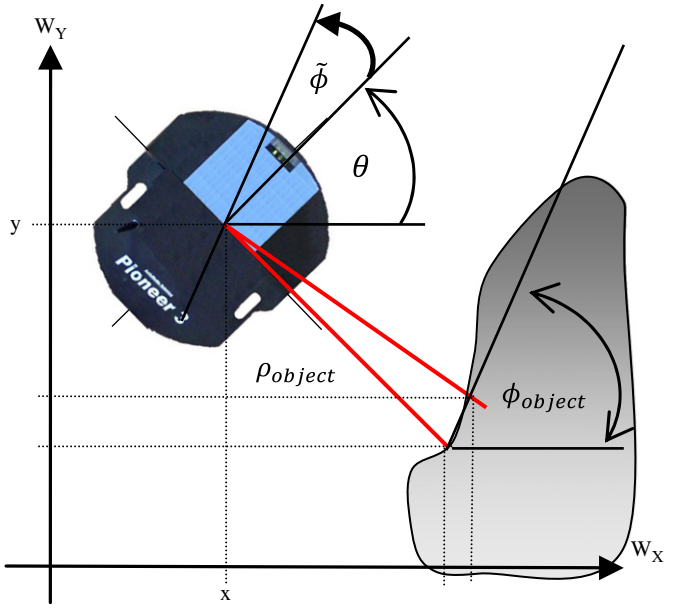

Fig. 9. Control states for the CF-controller.

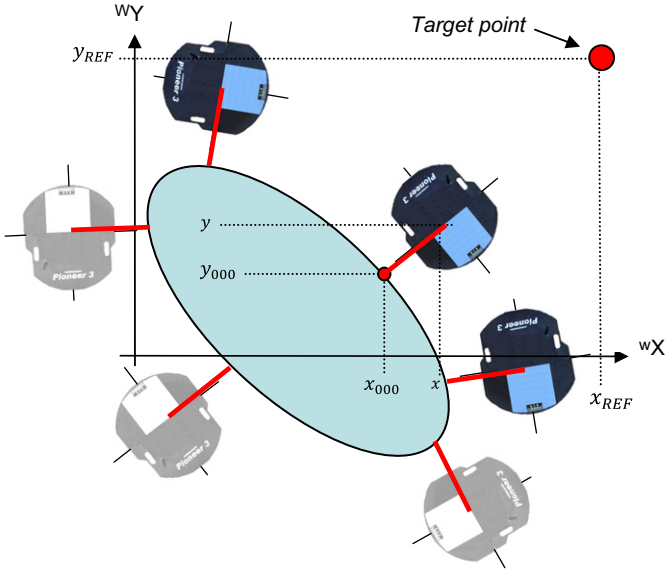

Fig. 10. Obstacle avoided detection example for the III quadrant (considering the goal point (x_{REF},y_{REF}) as the origin of a new Cartesian coordinate plane and the robot position point (x,y) as a point in space with some displacement from the origin) and for the robot following an obstacle (here with oval shape) at its right side. Black robots indicate the zone where the obstacle was avoided and the grey robots indicate the zones where the obstacle was not yet avoided. Similar graphs can be constructed for the other quadrants and for the robot following the obstacle at its left side.

uncertainty about obstacles' shapes and positions on the scenario generally leads to the apparition of deadlocks and local minima. Many papers were devoted to the solution of such algorithms. In [11] this problem is considered for path planning. The authors present the *Bug* algorithms that allow the robot to achieve a target position following the boundary of the obstacles always at the same side of the robot. The so-called *Bug1* algorithm is based on following the obstacles boundary; this happens until the closest point to the target point along the boundary is detected. Finally, in the *Bug2* algorithm, it is defined a main line connecting the initial and the target point and it is considered in order to leave the obstacle. More recently, in [23] the authors propose a switching multi-controller approach using a *look-ahead* controller several times until convergence is reached.

In this paper we propose a hybrid switching system composed by the parking controller of Section 3 and a contour-following controller, which is considered in detail in [15]. This is not a map dependent strategy, the robot only knows the initial and the target points, and it is equipped with odometry and distance (a laser range-finder) sensors in order to properly determine the obstacle position and to be able to follow its contour. These reactive approaches are characterized by its low computational cost (computing time and memory use), making possible to disregard

this aspect in the algorithm implementation. It is also assumed that there exists a feasible path to get the goal point (narrow corridors are detected as obstacles, whereas the minimal corridor narrowness depends on the selection of the robot guard-zone). The algorithm that rules our strategy could be summarized as follows:

(1) Compute the distance to the target point as a threshold value (*THR*).
(2) Parking controller (Section 3.3) is active in order to get the target position until one of the following occurs:
 a) Target is reached. Procedure stops.
 b) An obstacle is encountered go to Step 3.
(3) Determination of the robot side at which the obstacle will be followed (Section 4.1). Go to Step 4
(4) Follow the obstacle contour until one of the following occurs
 a) Target is reached. Procedure stops.
 b) The obstacle was surpassed (Section 4.2) at a distance which is less than the threshold value. Define a new initial position for the parking controller and go to Step 1.

Note that the robot will keep following the contour of the obstacle until the distance to the target point is less than the previously taken *THR* value. This solution does not define a global optimal path due to the unknown environment.

4.1. Contour-following controller [15]

This controller allows the robot to follow the discontinuous contour of any object at a desired constant distance ρ_{des}, considering a set of stable controllers providing the switching rule among them. In [15] it is proved that the control system stability for the control states $\begin{bmatrix}\tilde{\rho} & \tilde{\phi}\end{bmatrix}^T$ is

$$\tilde{\rho} = \rho_{des} - \rho_{object} \tag{17}$$

the distance error between the robot and the object, and

$$\tilde{\phi} = \phi_{object} - \theta \tag{18}$$

the orientation error between the robot and the measurement of the object orientation at this point (Fig. 9). The common Lyapunov function considered in [15] is

$$V_{CF} = \frac{\tilde{\phi}^2}{2} + \int_{0\tilde{\rho}} \kappa_{\tilde{\rho}}(\varsigma)\varsigma d\varsigma \tag{19}$$

Therefore, based on the results presented in [15] it will be assumed that the robot is able to asymptotically achieve the desired robot-obstacle distance ρ_{des}. Furthermore, this result is fulfilled for any obstacle shape, even for non-smooth contours.

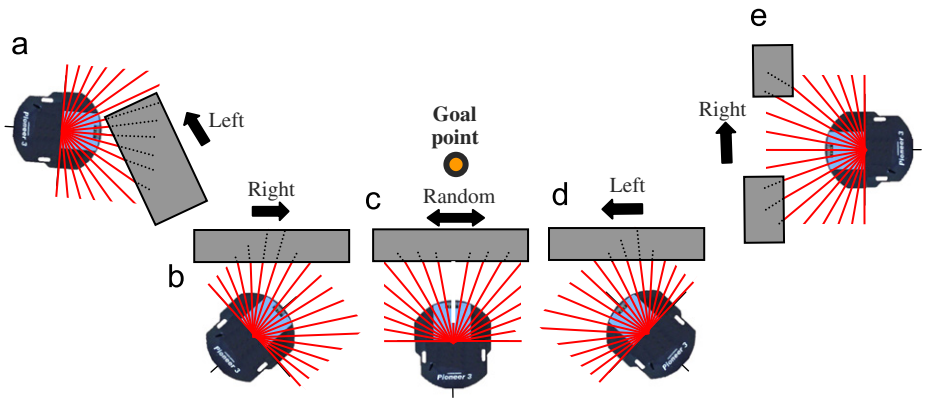

Fig. 11. Invasion to the safety-zone is stronger at the robot right-side in (a) and (d); in (c) the invasion is equal at both sides so the side to follow is selected randomly. Finally in (b) and (e) the robot turns right in order to follow the obstacle at its left side. Note that the robot head is always in the goal direction.

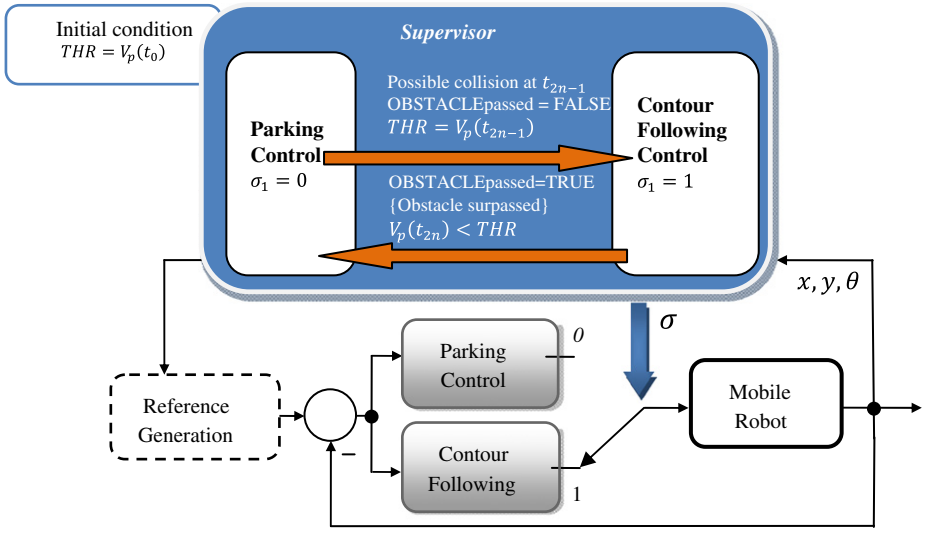

Fig. 12. Main control system block diagram. At instants t_{2n-1} the obstacle is encountered activating the contour-following mode. The leaving condition is activated at $t = t_{2n}$ when the obstacle was surpassed and the robot is closer to the target point (compared to the last surpassed obstacle).

4.2. Obstacle avoided detection

This algorithm is intended to detect an instant at which an obstacle could be considered as *surpassed* and it is useful to determine a leaving condition for the CF behavior. Such algorithm requires the knowledge of the current robot position(x,y), the desired final position (x_{REF},y_{REF}) and the position (x_{000},y_{000}) – or (x_{180},y_{180}) – of the laser beam at 0° – or 180° – which indicates the position at the right- or left-side of the robot. These points can be appreciated in Fig. 10. Then the problem is divided into four quadrants depending on the relation between the current and the final points. A flag variable *OBSTACLEpassed* is defined; the value TRUE for this variable indicates that the obstacle was actually surpassed. As an example for the case in which $x_{REF} > x$ and $y_{REF} > y$,

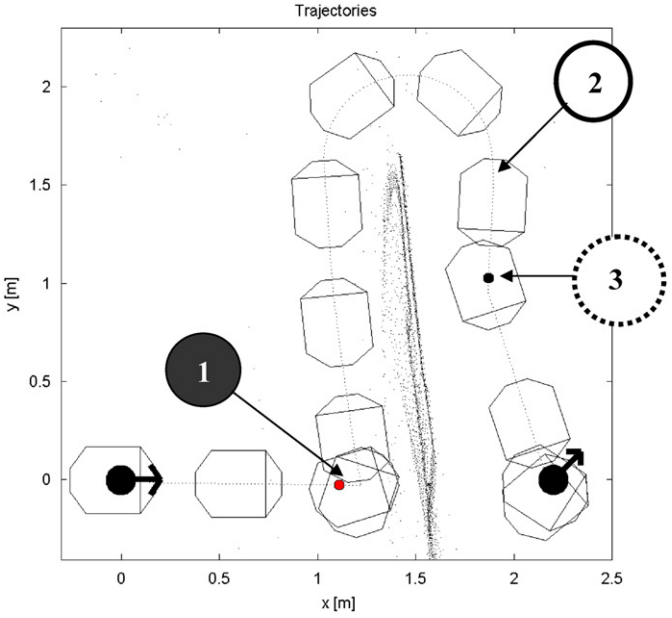

Fig. 13. Obstacle between the initial and the final point.

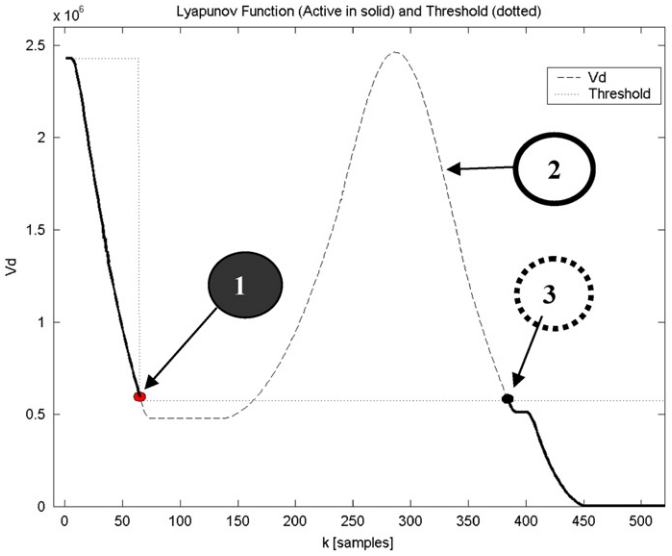

Fig. 14. Parking controller Lyapunov function. $<1>$: obstacle detected; $<2>$: obstacle avoided; and $<3>$: switching to the parking controller.

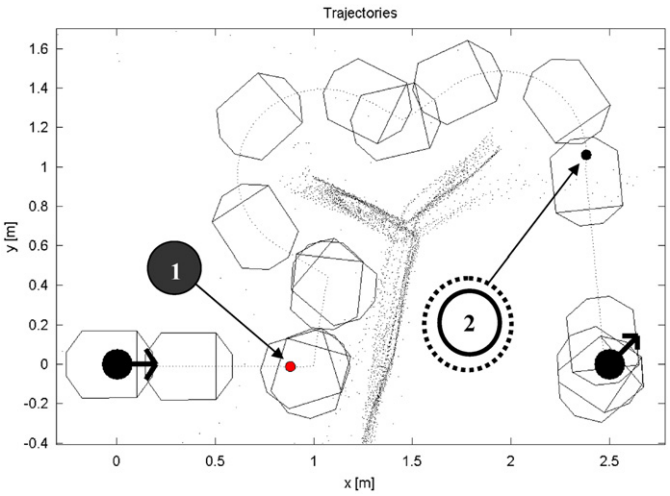

Fig. 15. Avoiding a trap situation.

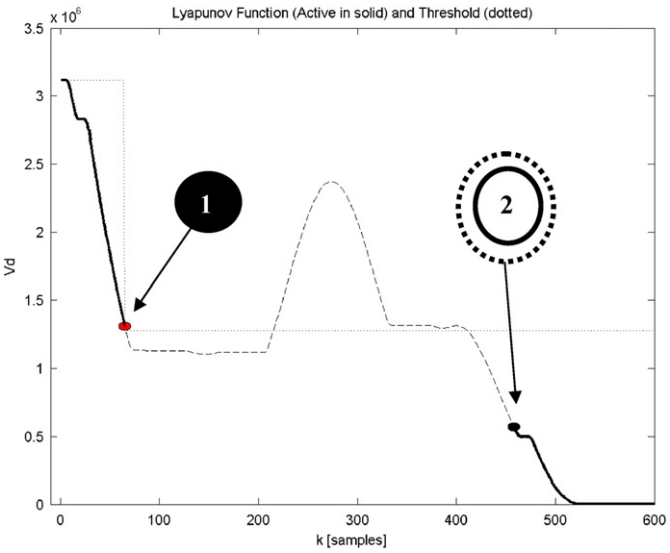

Fig. 16. Parking controller Lyapunov function. $<1>$: obstacle detected; $<2>$: obstacle avoided with a V_d value smaller than the threshold (direct switching case).

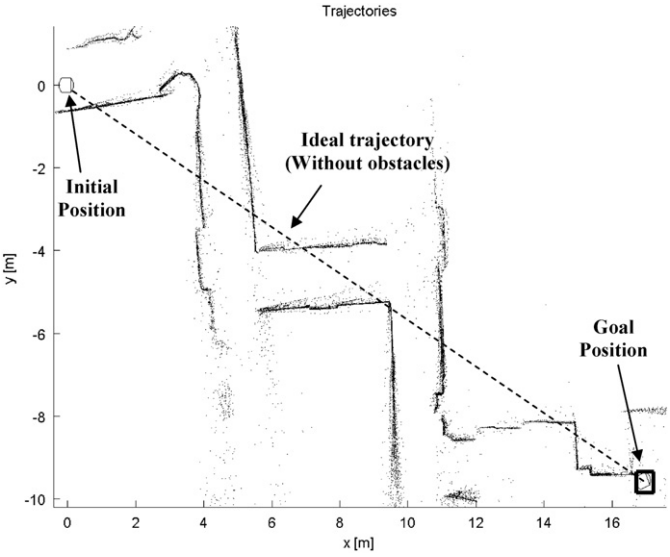

Fig. 17. Large-scale experiment setting (20 m long).

see Fig. 10, the algorithm is

$$OBSTACLE_{passed} = FALSE$$
$$if((x_{REF} > x) AND (y_{REF} > y) AND (x_{000} < x) AND (y_{000} < y)) \quad (20.a)$$
$$OBSTACLE_{passed} = TRUE$$

and for the other three quadrants: the equations become: (i) for $x_{REF} > x$ and $y_{REF} < y$, see (20.b); (ii) for $x_{REF} < x$ and $y_{REF} < y$, see (20.c), and finally, (iii) for $x_{REF} < x$ and $y_{REF} > y$, see (20.d):

$$OBSTACLE_{passed} = FALSE$$
$$if((x_{REF} > x) AND (y_{REF} < y) AND (x_{000} < x) AND (y_{000} > y)) \quad (20.b)$$
$$OBSTACLE_{passed} = TRUE$$

$$OBSTACLE_{passed} = FALSE$$
$$if((x_{REF} < x) AND (y_{REF} < y) AND (x_{000} > x) AND (y_{000} > y)) \quad (20.c)$$
$$OBSTACLE_{passed} = TRUE$$

$$OBSTACLE_{passed} = FALSE$$
$$if((x_{REF} < x) AND (y_{REF} > y) AND (x_{000} > x) AND (y_{000} < y)) \quad (20.d)$$
$$OBSTACLE_{passed} = TRUE$$

4.3. Right/left robot side to follow selection

As the obstacle avoidance problem is treated using a CF controller, it is important to select the side of the robot that will avoid the obstacle. To this aim, the safety-zone defined by the laser range-finder is employed in such a way that, analyzing the obstacle invasion according to Fig. 11, it is decided if the robot will avoid the obstacle to its left or to its right side. For brevity this algorithm is not explained in detail, but an intuitive approach is given in Fig. 11, where it can be seen how this algorithm depends on the safety-zone area occupied by the obstacle. It must be observed that this decision could affect the path length. But a wrong side to follow

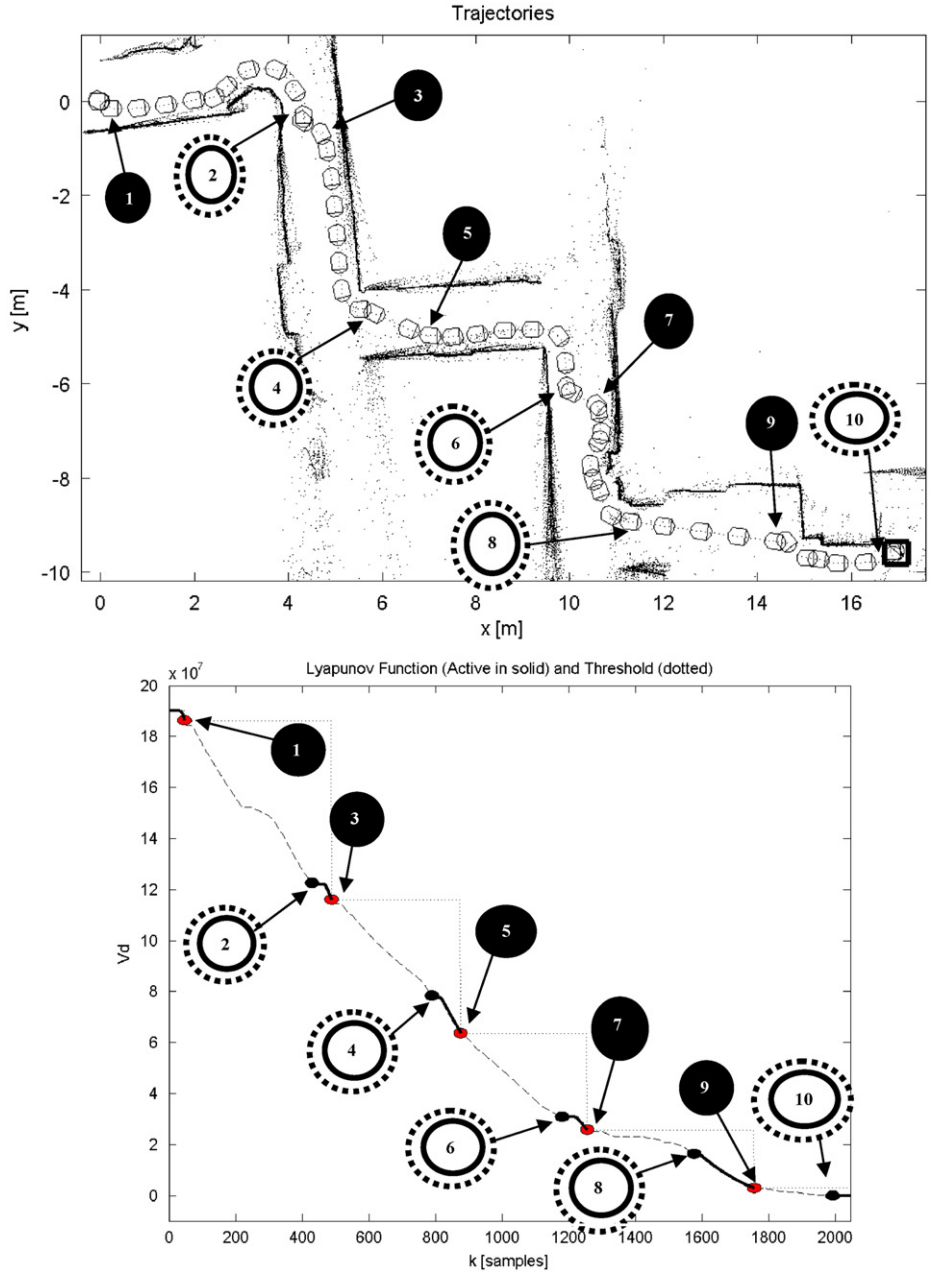

Fig. 18. $<(1,3,5,7,9>$ obstacle detected; $<2,4,6,8,10>$: obstacle avoided. Right picture: parking controller Lyapunov function.

could be practically detected and avoided in a second instance as e.g. in [24].

4.4. Block diagram

The block diagram of the control system is shown in Fig. 12 including both the parking controller (described in Section 3.1) and the contour following (CF) controller (described in Section 3.1). It is firstly defined a threshold value (THR) as the value of (11.a) at $t=t_0$, i.e., $THR=V_p(t_0)$. The switching signal is σ, whereas when $\sigma=0$, the robot is approaching the goal point using the parking controller and will only switch to the CF controller if an obstacle is detected at the instant $t=t_{2n-1}$,$n=1,2,\ldots$, updating the threshold value $THR=V_p(t_{2n-1})$. The CF controller allows the robot to follow the discontinuous contour of the obstacle at a desired constant distance, and then, at the instant $t=t_{2n}$ the supervisor selects the parking controller again provided that: (1) the obstacle was successfully surpassed (accordingly to the algorithm of Section 4.2) and (2) the robot-to-target distance is less than the previously defined threshold value, i.e. $V_p(t_{2n}) \leq THR$.

4.5. Stability analysis

In order to prove the asymptotic stability of this switching control system we considered a multiple Lyapunov functions approach (MLF) [16,17] associating a Lyapunov function to each controller (one for the parking controller and other for the contour following controller) and designing a logic that guarantees that the sequence of values for these functions is decreasing. Next, we state some useful and reasonably assumptions.

Assumption I. There are a finite number of obstacles, and each of them has finite length.

Assumption II. The performance of the main control system could be properly evaluated by (11.a).

Remark. Here, we relax the meaning of performance of the main controller to only "arrive at the goal point" being such information provided by the Lyapunov function (11.a), which depends only on the robot-to-goal distance d. This way, other states such as the orientations $\tilde{\theta}_1$ and $\tilde{\theta}_2$, and the CF-error $\tilde{\rho}$, are ignored when talking about the main controller's performance.

Assumption III. Perfect functioning of the CF controller assumed.

Remark. It will be assumed that the CF controller when active reduces to zero the error distance between the robot and the desired robot-to-obstacle distance (as derived from its stability property), i.e., $\tilde{\rho} \to 0$.

Proposition I. Guaranteeing that the discrete sequence associated to (11.a) – considering only the instants at which the positioning controller of Section 3.1 is activated – is decreasing, the overall switching controller of Fig. 11 is stable.

Proof. In order to prove asymptotic stability of a switching system composed by two subsystems considering multiple Lyapunov functions theory [25], both discrete sequences (associated to each subsystem) must be decreasing. In our case, the one associated to the parking controller and the one associated to the CF controller. Under Assumption III, the sequence associated to the activations of the CF controller is always decreasing. So, the stability property for the main switching controller depends only on the discrete sequence associated to the parking controller. □

Theorem I. The switching controller compound by the parking controller of Section 3.3 and the CF-controller of Section 4.1 switches according to

(i) Switching from the parking controller to the CF-controller ($\sigma=1$) at the switching instants t_{2n-1} with $n=1,2,\ldots$ is allowed to occur only if an obstacle is detected within a given robot neighborhood.
(ii) Switching from the CF-controller to the parking controller ($\sigma=0$) at switching instants t_{2n} with n=1,2,… is allowed to occur only if the obstacle has been surpassed according to the algorithm of Section 3.2 and the following condition is fulfilled:

$$V_P(t_{2n}) < V_P(t_{2n-1}) \tag{21}$$

is globally uniformly stable, recalling that V_p is the Lyapunov function associated to the Parking controller.

Proof. Condition (21) secures that the sequence associated to V_P will be decreasing as required, meanwhile if Assumption III holds, the overall switching system will be stable. In other words, as any obstacle of finite length detected at a distance $d(t_{2n-1})$ from the

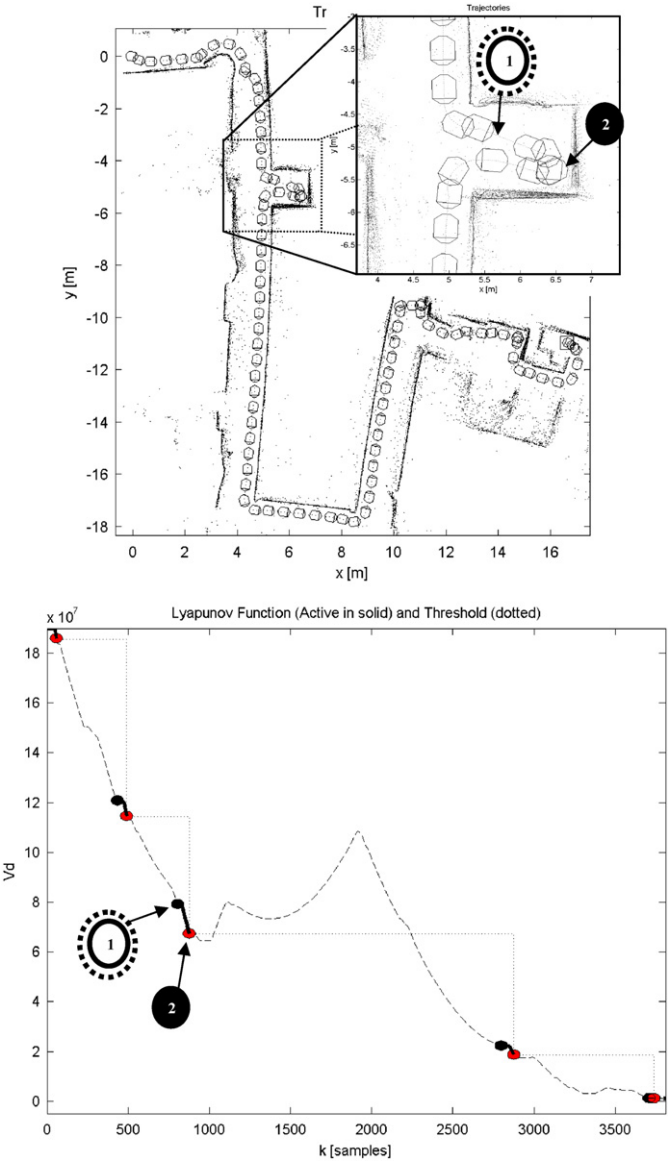

Fig. 19. The same experimental setting of Fig. 16, but blocking the corridor: <1>: obstacle avoided; <2>: obstacle detected. Right picture: parking controller Lyapunov function.

target point could be surpassed at a distance $d(t_{2n}) < d(t_{2n-1})$, then, condition (21) is fulfilled.

5. Experimental results

The experiments were carried out using a Pioneer IIIDX mobile robot equipped with an onboard Pentium III computer, a laser range-finder and internal odometry sensors. The robot does not have any camera or map information. The implementation of the overall control system needs 10 ms, which represents only a tenth of the robot sample time. In the first experiment (Figs. 13 and 14), it can be seen how the obstacle is detected in $<1>$ (red dot) and the value of V_P at this instant is considered as the new threshold value. Next, in $<2>$ the obstacle is considered as surpassed according to the algorithm of Section 3.1. However, since the value of V_P at this instant is still greater than the previous threshold the robot keeps following the obstacle until, at the instant $<3>$, at which the value of V_P is less than the threshold and the system switches to the parking controller (securing that V_P will be decreasing).

In the second experiment, it is considered a trap situation. As shown in Figs. 15 and 16, the obstacle is detected at the instant $<1>$, and the CF controller is activated, setting the corresponding *THR* value, then in $<2>$ the obstacle is surpassed and the value for the Lyapunov function (11.a) is less than the *THR* value, so the system switches again to the parking controller.

Note in both previous Figs. 14 and 16, the solid line on the Lyapunov function, which denotes that the parking controller is active, is always a decreasing function, in spite of having increasing parts at the moments where the controller is not active.

The third experiment, Fig. 17, proposes an autonomous navigation in a larger office setting. The environment is completely unknown and the goal position is about 20 m far from the initial point and the robot should cross several doors and corridors in order to get the final point. The path denoted as *idea trajectory* is the path expected to be followed by the robot if there were no obstacles in the trajectory.

From Fig. 18, it can be appreciated the resulting controller performance and the associated Lyapunov functions time-evolution. As can be seen, the task is easily solved by considering the proposed strategy. It could be remarked that the obstacles were found always at a distance less than the corresponding *THR* value and consequently the Lyapunov function is decreasing. For this reason, in order to obtain a more complicated test for the switching control system, one of the corridors was blocked in the fourth experiment, forcing the trap situation shown in Fig. 19.

Based on these experimental results, it could be interesting to compare our approach with other well-known algorithms which also allow the robot to achieve a target position while avoiding obstacles. These classical navigation algorithms belong to the *Bug* family [11], which combines local planning with global information (the target position related to the world coordinate system). Among several *Bug* algorithms [29–33], the performance (regarding path travelled) of our strategy is comparable to the *DistBug* [29] approach. From results reported in [34,35] and also from the comparison results presented in [29] it can be concluded that the *DistBug* algorithm's performance is notoriously better than the performance presented by the other cited strategies and perfectly comparable with ours. In fact, both strategies appear to have similar performances comparing the trajectories found. The difference (at the algorithm level) lies in the leaving condition, i.e., the instant at which the robot abandon the obstacle boundary, since the *DistBug* considers an auxiliary distance in free space *F*, which is used in the determination of the leaving condition. The authors consider that this distance in free space is in fact reachable by the robot, and based on this assumption, the obstacle's boundaries can be abandoned. Instead of this, we propose to leave the obstacle's boundaries when there have been surpassed at an appropriated real distance to the destiny point in spite of an assumed reachable distance. The length of the path found by each algorithm could be larger or shorter depending on the configuration of the obstacles as can be observed in the examples shown in Fig. 20. In Fig. 20a it is shown a two-obstacles setting, where the *DistBug* approach achieves a slightly shorter path. Next, in Fig. 20b and d both strategies find the same path. Fig. 20c shows a scenario where the assumption made by the *DistBug* algorithm produces a considerably larger path. It must also be noted that different from these related papers, our work not only deals with the switching rules but also includes a full description of the autonomous navigation control algorithm with formal stability analysis.

6. Conclusions

In this paper, we have presented a fully described switching controller that solves the problem of parking a wheeled mobile

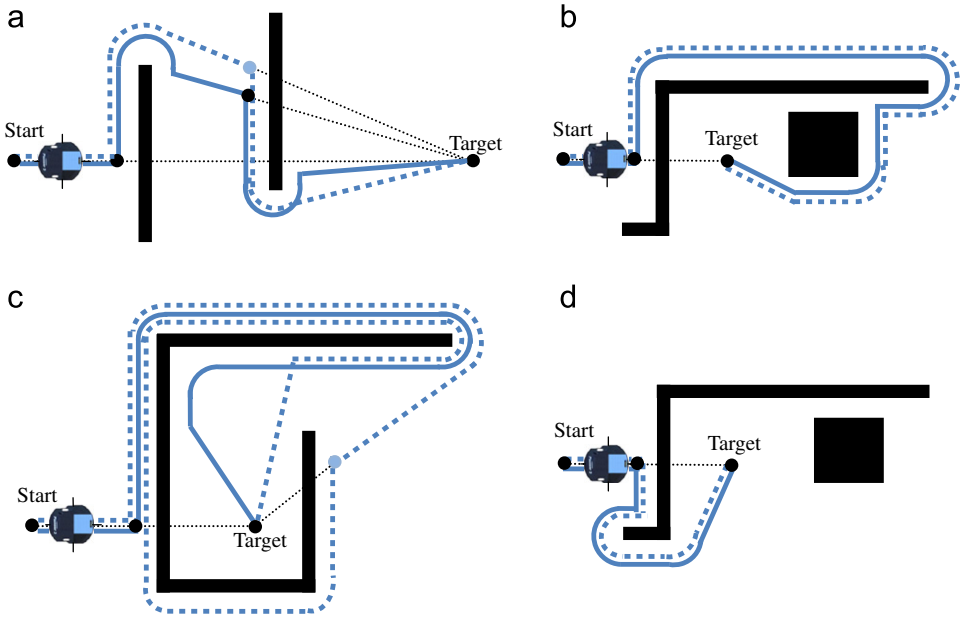

Fig. 20. Paths found using the proposed algorithm (continuous path) and the one presented in [29] (dotted path).

robot (positioning with final orientation) avoiding unknown obstacles. The strategy has been based on the reactive contour following the obstacles, and to this aim two complementary algorithms have been included: one that allows the robot to detect when an obstacle was or was not avoided (a leaving condition of the contour following controller), and other that selects the side to avoid the obstacle. The presented switching controllers: the parking controller and the parking/contour following controllers, include the stability analysis at the switching instants. In spite of the complexity of the problem due mainly for the uncertainties on the robot environment, the asymptotic stability is proved provided some common sense assumptions are fulfilled: such as the presence of static isolated obstacles and the possibility to achieve the target position. Furthermore, the performance of this strategy has shown good results though several experiments in real settings. Nevertheless, adding knowledge about the robot environment, as e.g. local map information, would improve its performance. Potential applications for this approach can be found in many areas such as exploration, map construction and search/rescue tasks.

References

[1] Aicardi M, Casalino G, Bicchi A, Balestrino A. Closed loop steering of unicycle like vehicles via Lyapunov techniques. IEEE Robotics and Automation Magazine 1995;2(1):27–35.
[2] Arkin RC. Behavior-based robotics. MIT Press; 1998.
[3] Brockett R. Differential Geometric Control Theory. Birkhauser; 1983 pp. 181–91.
[4] Fierro R, Lewis FL. Practical point stabilization of a nonholonomic mobile robot using neural networks. In: Proceedings of the conference on decision and control; 1996. pp. 1722–27.
[5] de Wit C, Sordalen OJ. Exponential stabilization of mobile robots with non-holonomic constraints. Transactions on Automatic Control 1992;37: 1792–7.
[6] Tayebi A, Rachid A. A unified discontinuous state feedback controller for the path-following and the point-stabilization problems of a unicycle-like mobile robot. In: Proceedings of the IEEE international conference on control applications, vol. 6; 1997. pp. 31–35 .
[7] Lee D. Passivity-based switching control for stabilization of wheeled mobile robots. In: Proceedings of the robotics science and systems. Available on line at the author web page: 〈http://web.utk.edu/~djlee/papers/RSS07.pdf〉; 2007.
[8] Wang M, Liu J. Autonomous robot navigation using fuzzy logic controller. International Conference on Machine Learning and Cybernetics 2004:26–29.
[9] Carelli R, Oliveira Freire E. Corridor navigation and wall-following stable control for sonar-based mobile robots. Robotics and Autonomous Systems 2003;45:235–47.
[10] Bicho E. Detecting, representing and following walls based on low-level distance sensors. In: International symposium on neural computing; 2000.
[11] Lumelsky V, Stepanov AA. Dynamic path planning for a mobile automaton with limited information on the environment. IEEE Transactions on Automatic Control 1986;31:11.
[12] Grisetti G, Stachniss C, Burgard W. Improved techniques for grid mapping with Rao-balckwellized particle filters. IEEE Transactions on Robotics 2007;23:1.
[13] Calisi D, Farinelli A, Iocchi L, Nardi D. Autonomous navigation and exploration in a rescue environment. In: International workshop on safety, security and rescue robotics; 2005.
[14] Betke M, Gurvits L. Mobile robot localization using landmarks. IEEE Transactions on Robotics and Automation 1997;13:2.
[15] Toibero JM, Roberti F, Carelli R. Stable switching contour-following control of wheeled mobile robots. ROBOTICA 2008;27:1–12.
[16] Liberzon D. Switching in systems and control. Birkhauser; 2003.
[17] Branicky MS. Stability of hybrid systems: state of the art. In: Proceedings of the IEEE conference on decision & control; 1997. pp. 120–25.
[18] Toibero M, Carelli R, Kuchen K. Switching control of mobile robots for autonomous navigation in unknown environments. International Conference on Robotics and Automation 2007:1974–79.
[19] Burgard W, Moors M, Stachniss C, Schneider F. Coordinated multi-robot exploration. IEEE Transactions on Robotics 2005;21:3.
[20] Secchi H, Carelli R, Mut V. Design of stable algorithms for mobile robots control with obstacle avoidance. In: Proceedings of the 14th IFAC World Congress; 1999. pp.185–90.
[21] Pavlidis T. Stability of systems described by differential equations containing impulses. IEEE Transactions on Automatic Control 1967;12(1):79–83.
[22] Ye H, Michel AN, Hou L. Stability analysis of systems with impulse effects. IEEE Transactions on Automatic Control 1998;43(12):1719–23.
[23] Mansard N, Chaumette F. Task sequencing for high-level sensor based control. IEEE Transactions on Robotics 2007;23(1):60–72.
[24] Balch T, Arkin R Avoiding the past: a simple but effective strategy for reactive navigation. In: Proceedings of the IEEE international conference on robotics and automation; 1993.
[25] Branicky MS. Multiple Lyapunov functions and other analysis tools for switched and hybrid systems. IEEE Transactions on Automatic Control 1998;43(4):475–83.
[26] Yang X, Moallem M, Patel R. A layered goal-oriented fuzzy motion planning strategy for mobile robot navigation. IEEE Transactions on Systems, Man and Cybernetics—Part B: Cybernetics 2005;35(5):1214–24.
[27] Tovar B, Murrieta-Cid R, La Valle S. Distance optimal navigation in an unknown environment without sensing distances. IEEE Transactions on Robotics 2007;23(3):506–19.
[28] Chatterjee D, Liberzon D. Stability analysis and stabilization of randomly switched systems. In: Proceedings of the IEEE conference on decision & control; 2006.
[29] Kamon I, Rivlin E. Sensory-based motion planning with global proofs. IEEE Transactions on Robotics and Automation 1997;13(6):814–22.
[30] Magid E, Rivlin E. CAUTIOUSBUG: a competitive algorithm for sensory-based robot navigation. In: Proceedings of the 2004 IEEE/RSJ international conference on intelligent robots and Systems, Sendai, Japan; 2004.
[31] Kamon I, Rimon E, Rivlin. E. Tangentbug: a range-sensor-based navigation algorithm. The International Journal of Robotics Research 1998;17(9):934–53.
[32] Sankaranarayanan A, Vidyasagar M. Path planning for moving a point object amidst unknown obstacles in a plane: a new algorithm and a general theory for algorithm development. In: Proceedings of the 39th conference on decision and control, Honolulu, Hawai; 1990.
[33] Taylor K, LaValle SM. I-Bug: an intensity-based bug algorithm. In: Proceedings of the IEEE international conference on robotics and automation; 2009.
[34] Yufka A, Parlaktuna O. Performance comparison of bug algorithm for mobile robots. In: Proceedings of the 5th international advanced technologies symposium, Karabuk, Turkey; 2009.
[35] Noborio H, Fujimora K, Horiuchi Y. A comparative study of sensor-based path-planning algorithms in an unknown maze. In: Proceedings of the 2000 IEEE/RSJ international conference on intelligent robots and systems; 2000.