Lionel Lapierre
Rene Zapata
Pascal Lepinay
Université Montpellier II
161, rue Ada 34392 Montpellier Cedex 5
{lapierre, zapata, lepinay}@lirmm.fr

Combined Path-following and Obstacle Avoidance Control of a Wheeled Robot

Abstract

This paper proposes an algorithm that drives a unicycle type robot to a desired path, including obstacle avoidance capabilities. The path-following control design relies on Lyapunov theory, backstepping techniques and deals explicitly with vehicle dynamics. Furthermore, it overcomes the initial condition constraints present in a number of path-following control strategies described in the literature. This is done by controlling explicitly the rate of progression of a "virtual target" to be tracked along the path; thus bypassing the problems that arise when the position of the path target point is simply defined as the closest point on the path. The obstacle avoidance part uses the Deformable Virtual Zone (DVZ) principle. This principle defines a safety zone around the vehicle in which the presence of an obstacle induces an "intrusion of information" that drives the vehicle reaction. The overall algorithm is combined with a guidance solution that embeds the path-following requirements in a desired intrusion information function, which steers the vehicle to the desired path while the DVZ ensures minimal contact with the obstacle, implicitly bypassing it. Simulation and experimental results illustrate the performance of the control system proposed.

KEY WORDS—

1. Introduction

Real-time obstacle avoidance coupled with accurate path- following control is one of the major issues in the field of mobile robotics (Ogren 1989, 2003). The underlying problem to be solved can be divided into three areas:

The International Journal of Robotics Research
Vol. 00, No. 00, Xxxxxxxx 2007, pp. 000–000
DOI: 10.1177/0278364906076790
©2007 SAGE Publications
Figures 4–7 appear in color online: http://ijr.sagepub.com

- path-following control of nonholonomic systems,
- obstacle avoidance strategy,
- coupling between the above two areas.

1.1. Path-following control of a nonholonomic system

The general underlying assumption in path-following control is that the vehicle's forward velocity tracks a desired speed profile, while the controller acts on the vehicle orientation to drive it to the path. See Micaeli and Samson (1993), and Samson and Ait-Abderrahim (1991) for pioneering work in the area as well as Canudas de Wit et al. (1993), Jiang and Nimeijer (1999), and Soetanto et al. (2003) and references therein. Path-following systems for marine vehicles have been reported by Encarnação et al. (2000), Encarnação and Pascoal (2000), and Lapierre et al. (2003). The main contribution of the method in Lapierre et al. (2003) is threefold:

i) it extends the results obtained by Micaeli and Samson (1993) and Soetanto et al. (2003) – for kinematic wheeled robots – to a more general setting, in order to deal with vehicle dynamics and parameter uncertainty,

ii) it overcomes stringent initial condition constraints that are present in a number of path-following control strategies in the literature. This is done by controlling explicitly the rate of progression of a *virtual target* to be tracked along the path, thus bypassing the problems that arise when the position of the target point on the path is simply defined by the projection of the actual vehicle on the path. This procedure avoids the singularities that occur when the vehicle is located at the current center of curvature of the path virtual target location (where the closest point is not unique), and allows for global convergence of the vehicle to the desired path. The path is a general curve, described by its curvature as a function of the curvilinear abscissa; the singularity occurs when the robot is located at the centre of the path curvature where the virtual target is located. This is in contrast with the results described by Micaeli and Samson (1993) for example, where only local convergence is proven. To the best of the authors' knowledge, the idea of exploring the extra degree of freedom that comes from controlling the motion of a virtual target along a path appeared for the first time in the work of Aicardi et al. (1995) for the control of wheeled robots. This idea was later extended to the control of marine craft in the work of Aicardi et al. (2001). However, none of these addresses the issues of vehicle dynamics and parameter uncertainty. Furthermore, the methodologies adopted by Aicardi et al. (1995, 2001) for control system design build on entirely different techniques that required the introduction of a nonsingular transformation in the original error space.

iii) It combines path-following control and obstacle avoidance.

In this paper, controller design builds on the work reported in Micaeli and Samson (1993) on path-following control and relies heavily on Lyapunov-based and backstepping techniques (Krstić et al. 1995; Fierro and Lewis 1997) in order to extend kinematic control laws to a dynamic setting. In Soetanto et al. (2003) parameter uncertainty is considered within a framework by augmenting Lyapunov function candidates with terms that are quadratic in the parameter error. For the sake of clarity, in this paper, we do not extend the dynamic control with the parameter uncertainty robust scheme.

1.2. Obstacle Avoidance

The obstacle avoidance strategy is another major issue affecting applications in the field of mobile robotics, and comprises two different issues:

- obstacle detection,
- computation of the system reaction.

Obstacle detection requires the system to be equipped with sensor devices able to estimate the distance between the robot and the obstacles. Two different types of sensors are generally used:

- A belt of proximity sensors (ultrasound, infrared, ...) mounted on the vehicle allowing the acquisition of a discrete estimation of the free space around the robot.
- A rotating laser beam coupled with a vision system, or a stereo vision system, permitting continuous estimation of the polar signature of the free space region around the vehicle.

System reaction can be at the navigation system level (path replanning) or directly in the controller as a *reflex behavior*. Reaction quantification is generally made according to an arbitrary positive potential field function attached to obstacles, which repels the robot, and an attractive field located on the goal. The main difficulty with this method is the design of an artificial potential function without undesired local minima. Elnagar and Hussein (2002) propose to model the potential field using Maxwell's equations that completely eliminate the local minima problem, with the condition that an *a priori* knowledge of the environment is available. These methods are generally computationally intensive. Iniguez and Rossel (2002) propose a hierarchical and dynamic method that works on a no regular grid decomposition, simple and computationally efficient, both in time and memory. Ge and Cui (2000) address the problem of the adaptation of the potential field magnitude, in order to avoid the problematic situation where the goal is too close to an obstacle, inducing an inefficient attractive field, termed *Goals Non-reachable with Obstacles Nearby*. The work of Louste (1999) tackles the problem of coupling a path-planning method, based on viscous fluid propagation, with nonholonomic robot kinematics restrictions. Another approach, based on a *reflex behavior* reaction, uses the *Deformable Virtual Zone* (DVZ) concept, in which a robot kinematic dependent *risk zone* is located surrounding the robot. Deformation of this zone is due to the intrusion of proximity information. The system reaction is made in order to reform the *risk zone* to its nominal shape, implicitly moving away from obstacles (Zapata 2004).

1.3. Coupling Obstacle Avoidance and Path Following

Two different methods are generally proposed in the literature for obstacle avoidance algorithms, coupled with a nominal objective, defined as path following, trajectory tracking or point stabilization:

- path replanning,
- reactive path following.

The path replanning strategy requires the system to design a safe path between obstacles that ensures the vehicle avoids obstacles and finally reaches the desired goal. The advantage of this method is that, under the assumption that an accurate map of the environment is available, it designs a global solution that guarantees the system reaches the goal, if this solution exists. Moreover, since the system is necessarily equipped with proximity sensors, the requirements for an environmental map construction are met. This allows the use of SLAM (Simultaneous Localization and Mapping) or CML (Concurrent Localization and Mapping) techniques for accurate navigation systems, in which the path replanning method naturally takes place. The drawback is the necessary computational time, which may cause the system to stop in front of an uncharted obstacle. Finally the control objective is to follow the computed path (Elnagar and Hussein 2002; Iniguez and Rossel (2002); Rimon and Koditschek 1992).

Reactive path following, which acts directly at the control level, is generally considered as a behavior-based approach, and includes fuzzy or neural systems, Althaus and Christensen (2002) and Arkin (1998). Several behaviors are defined (avoid-obstacle, move to goal, ...) and a switching scheme is designed to switch between these behaviors.

1.4. Outline of this paper

We have investigated the coupling between the path-following algorithm described in Soetanto et al. (2003) and a DVZ-based reactive obstacle avoidance control. The proposed method consists in a guidance solution that embeds the path-following requirements in a desired proximity function (with respect to the obstacles) that drives the robot to contour the obstacles while guaranteeing the path-following convergence requirements when there is no obstacle. This approach is based on the derivation of a Lyapunov function that guarantees asymptotic convergence to the path without obstacles, and the boundedness of a variable called the *intrusion ratio*, that captures the surrounding obstacles proximity and the current robot situation with respect to the path. The combination of path following with a reactive – local – obstacle avoidance strategy has a natural limitation in the situation that both controllers yield antagonistic system reactions. In the proposed method, this situation leads to a local minimum called the *corner situation*, where a heuristic switch between controllers is necessary. The advantage of the method is that outside this very specific *corner situation*, no switch is required.

The paper is organized as follows: Sections 2 and 3 review the basic algorithms we use for path following and obstacle avoidance. Section 4 shows our solution combining both controllers and Section 5 illustrates the results obtained. Finally, Section 6 presents conclusions and suggests further work.

2. Path-following Algorithm

2.1. Path Following. Problem Formulation

This section reviews a solution to the problem of steering a unicycle type vehicle along a desired path.

The following assumptions are made regarding the robot (see Figure 1). The vehicle, of width $2L$, has two identical parallel, non-deformable rear wheels of radius R, which are controlled by two independent motors, and a passive front wheel. It is assumed that the plane of each wheel is perpendicular to the ground and that the contact between the wheels and the ground is pure rolling and non-slipping, i.e. the velocity of the center of mass of the robot is orthogonal to the rear wheels axis. It is further assumed that the masses and inertias of the wheels are negligible and that the center of mass of the mobile robot is located in the middle of the axis connecting the rear

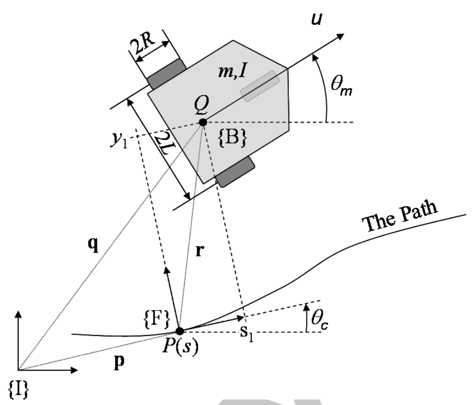


Fig. 1. Path following: frames definition and problem pose description.

wheels. The wheels control provides a forward force F and an angular torque N applied at the vehicle's center of mass. The vehicle mass and moment of inertia are denoted m and I, respectively. Let u and r denote the forward and rotational velocity of the vehicle.

2.1.1. Kinematic equations of motion. The Serret–Frenet Frame

The solution to the problem of path following derived in Micaelli and Samson (1995) admits an intuitive explanation: a path-following controller should look at (i) the distance from the vehicle to the path and (ii) the angle between the vehicle velocity vector and the tangent to the path, and reduce both to zero. This motivates the development of the kinematic model of the vehicle in terms of a Serret–Frenet frame {F} that moves along the path, with abscissa denoted s_1 and ordinate y_1; {F} plays the role of the body axis of a "virtual target vehicle" that should be tracked by the "real vehicle". Using this set-up, the abovementioned distance and angle become the coordinates of the error space in which the control problem is formulated and solved. In this paper, motivated by the work in Micaelli and Samson (1995), a Frenet frame {F} that moves along the path to be followed is used with a significant difference: *the Frenet frame is not attached to the point on the path that is closest to the vehicle*. Instead, the origin of {F} along the path is made to evolve according to a conveniently defined function of time, effectively yielding an extra controller design parameter. As will be seen, this seemingly simple procedure allows one to lift the stringent initial condition constraints that arise with the path-following controller described in Micaelli and Samson (1993). The notation that follows is now standard.

Consider Figure 1, where P is an arbitrary point on the path to be followed and Q is the center of mass of the moving vehicle. Associated with P, consider the corresponding

Serret–Frenet frame {F}. The signed curvilinear abscissa of P along the path is denoted s. Clearly, Q can either be expressed as $\mathbf{q} = (X, Y, 0)$ in a selected inertial reference frame {I} or as $(s_1, y_1, 0)$ in {F}. Stated equivalently, Q can be given in (X, Y) or (s_1, y_1) coordinates (see Figure 1). Let the position of point P in {I} be vector $\mathbf{p}$. Let

$$\mathbf{R} = \begin{bmatrix} \cos\theta_c & \sin\theta_c & 0 \\ -\sin\theta_c & \cos\theta_c & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

be the rotation matrix from {I} to {F}, parameterized locally by the angle θ_c. Define $\omega_c = \dot{\theta}_c$. Then,

$$\begin{cases} \omega_c = \dot{\theta}_c = c_c(s)\dot{s} \\ \dot{c}_c(s) = g_c(s)\dot{s} \end{cases} \qquad (1)$$

where $c_c(s)$ and $g_c(s) = \frac{dc_c(s)}{ds}$ denote the path curvature and its derivative, respectively. The velocity of P with respect to {I} can be expressed in {F} to yield

$$\left(\frac{d\mathbf{p}}{dt}\right)_F = \begin{bmatrix} \dot{s} \\ 0 \\ 0 \end{bmatrix}.$$

It is also straightforward to compute the velocity of Q in {I} as

$$\left(\frac{d\mathbf{q}}{dt}\right)_I = \left(\frac{d\mathbf{p}}{dt}\right)_I + \mathbf{R}^{-1}\left(\frac{d\mathbf{r}}{dt}\right)_F + \mathbf{R}^{-1}(\omega_c \times \mathbf{r})$$

where $\mathbf{r}$ is the vector from P to Q. Multiplying the above equation on the left by $\mathbf{R}$ gives the velocity of Q in {I} expressed in {F} as

$$\mathbf{R}\left(\frac{d\mathbf{q}}{dt}\right)_I = \left(\frac{d\mathbf{p}}{dt}\right)_F + \left(\frac{d\mathbf{r}}{dt}\right)_F + \omega_c \times \mathbf{r}$$

Using the relations

$$\left(\frac{d\mathbf{q}}{dt}\right)_I = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ 0 \end{bmatrix},$$

$$\left(\frac{d\mathbf{r}}{dt}\right)_F = \begin{bmatrix} \dot{s}_1 \\ \dot{y}_1 \\ 0 \end{bmatrix},$$

and

$$\begin{aligned} \omega_c \times \mathbf{r} &= \begin{bmatrix} 0 \\ 0 \\ \dot{\theta} = c_c(s)\dot{s} \end{bmatrix} \times \begin{bmatrix} s_1 \\ y_1 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} -c_c(s)\dot{s}y_1 \\ c_c(s)\dot{s}s_1 \\ 0 \end{bmatrix}, \end{aligned}$$

equation (2) can be rewritten as

$$\mathbf{R}\begin{bmatrix} \dot{X} \\ \dot{Y} \\ 0 \end{bmatrix} = \begin{bmatrix} \dot{s}(1 - c_c(s)y_1) + \dot{s}_1 \\ \dot{y}_1 + c_c(s)\dot{s}s_1 \\ 0 \end{bmatrix}.$$

Solving for $\dot{s}_1$ and $\dot{y}_1$ yields

$$\begin{cases} \dot{s}_1 = \begin{bmatrix} \cos\theta_c & \sin\theta_c \end{bmatrix} \begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} - \dot{s}\,(1 - c_c y_1) \\ \dot{y}_1 = \begin{bmatrix} -\sin\theta_c & \cos\theta_c \end{bmatrix} \begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} - c_c\dot{s}s_1. \end{cases} \qquad (2)$$

At this point it is important to notice that in Micaelli and Samson (1993), $s_1 = 0$ for all t, since the location of point P is defined by the projection of Q on the path, assuming the projection is well defined. One is then forced to solve for $\dot{s}$ in the equation above. However, by doing so $1 - c_c y_1$ appears in the denominator, thus creating a singularity at $y_1 = \frac{1}{c_c}$. As a result, the control law derived in Micaelli and Samson (1993) requires that the initial position of Q be restricted to a tube around the path, the radius of which must be less than $\frac{1}{c_{c,\max}}$, where $c_{c,\max}$ denotes the maximum curvature of the path. Clearly, this constraint is very conservative since the occurrence of a large $c_{c,\max}$ in only a very small section of the path will impose a rather strict constraint on the initial vehicle position, no matter where it starts with respect to that path.

By making s_1 not necessarily equal to zero, a virtual target that is not coincident with the projection of the vehicle on the path is created, thus introducing an extra degree of freedom for controller design. By specifying how fast the newly defined target moves, the occurrence of a singularity at $y_1 = \frac{1}{c_c}$ is removed. The velocity of the unicycle in the {I} frame satisfies the equation

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} = u \begin{bmatrix} \cos\theta_m \\ \sin\theta_m \end{bmatrix} \qquad (3)$$

where θ_m and u denote the yaw angle of the vehicle and its body-axis speed, respectively. Substituting (3) in (2) and introducing the variable $\theta = \theta_m - \theta_c$ gives the kinematic model of the unicycle in (s, y) coordinates as

$$\begin{cases} \dot{s}_1 = -\dot{s}\,(1 - c_c y_1) + u\cos\theta \\ \dot{y}_1 = -c_c\dot{s}s_1 + u\sin\theta \\ \dot{\theta} = \omega_m - c_c\dot{s} \end{cases} \tag{4}$$

where $\boldsymbol{\omega}_m = \dot{\theta}_m$.

2.1.2. Dynamics. Problem Formulation

The dynamical model of the unicycle is obtained by augmenting (4) with the equations

$$\begin{cases} \dot{u} = \frac{F}{\mathcal{M}} \\ \dot{\omega} = \dot{\omega}_m - c_c\ddot{s} - g_c\dot{s}^2 \end{cases} \tag{5}$$

where $\dot{\omega}_m = \frac{N}{\mathcal{I}}$ and $\mathcal{M}$ and $\mathcal{I}$ are the mass and the moment of inertia of the unicycle, respectively. Let F_{PF} and N_{PF} be the path-following control inputs, the forward force and the angular torque, respectively.

With the above notation, the problem under study can be formulated as follows:

Given a desired speed profile $u_d(t) > u_{\min} > 0$ for the vehicle speed u, derive a feedback control law for F_{PF} and N_{PF} to drive y_1, θ, and $u - u_d$ asymptotically to zero.

2.2. Nonlinear Controller Design

This section introduces a nonlinear closed loop control law to steer the dynamic model of a wheeled robot described by (4) and (5) along a desired path. Controller design builds on previous work by Micaeli and samson (1993) on path-following control and relies heavily on backstepping techniques.

2.2.1. Kinematic Controller design

The analysis that follows is inspired by the work in Samson and Ait-Abderrahim (1991) and Micaelli and Samson (1993) on path-following control for kinematic models of wheeled robots. Recall from the problem definition in the previous section that the main objective of the path-following control law is to drive y_1 and θ to zero. Starting at the kinematic level, these objectives can be embodied in the Lyapunov function candidate (Micaelli and Samson 1993).

$$V_1 = \frac{1}{2}(s_1^2 + y_1^2) + \frac{1}{2\gamma}(\theta - \delta(y_1, u))^2 \tag{6}$$

where γ is a positive gain, and it is assumed that

A.1. $\delta(y_1, u)$ is a bounded differentiable function with respect to y_1 and $\delta(0, u) = 0$. This function will be called in the sequel 'approach angle'.

A.2. $y_1 u \sin\delta(y_1, u) \le 0, \forall y\ \forall u$.

A.3. $\lim_{t\to\infty} u(t) \neq 0$.

In the V_1 Lyapunov function adopted, the first term $\frac{1}{2}(s_1^2 + y_1^2)$ captures the distance between the vehicle and the path, which must be reduced to 0. The second term aims to shape the approach angle $\theta = \theta_m - \theta_c$ of the vehicle to the path as a function of the 'lateral' distance y_1 and speed u, by forcing it to follow a desired orientation profile embodied in the function δ. See Samson and Ait-Abderrahim (1991), where the use of a δ function of this kind was first proposed.

Assumption **A.1**. specifies that the desired relative heading vanishes as y_1 goes to zero, thus imposing the condition that the vehicle's main axis must be tangential to the path when the lateral distance y_1 is 0. Assumption **A.2**. provides an adequate reference sign definition in order to drive the vehicle to the path (turn left when the vehicle is on the right side of the path, and turn right in the other situation). Finally, Assumption **A.3**. states that the vehicle does not tend to a state of rest. The need for these conditions will become apparent in the development that follows. The derivative of V_1 can be easily computed to give

$$\begin{aligned} \dot{V}_1 &= s_1\dot{s}_1 + y_1\dot{y}_1 + \frac{1}{\gamma}(\theta - \delta)(\dot{\theta} - \dot{\delta}) \\ &= s_1\,(u\cos\theta - \dot{s}\,(1 - c_c y_1) - \dot{s}c_c y_1) \\ &+ \ y_1 u\sin\theta + \frac{1}{\gamma}(\theta - \delta)(\dot{\theta} - \dot{\delta}) \\ &= s_1\,(u\cos\theta - \dot{s}) + y_1 u\sin\delta + \frac{1}{\gamma} \\ &\cdot\ (\theta - \delta)\left(\dot{\theta} - \dot{\delta} + \gamma\, y_1 u\frac{\sin\theta - \sin\delta}{\theta - \delta}\right). \end{aligned}$$

Let the ideal (also called virtual) "kinematic control laws" for s and θ be defined as

$$\begin{cases} \dot{s} = u\cos\theta + k_1 s_1; \\ \dot{\theta} = \dot{\delta} - \gamma\, y_1 u\frac{\sin\theta - \sin\delta}{\theta - \delta} \\ \qquad - k_2(\theta - \delta). \end{cases} \tag{7}$$

where k_1 and k_2 are positive gains. Then,

$$\dot{V}_1 = -k_1 s_1^2 + y_1 u\sin\delta - k_2\frac{(\theta - \delta)^2}{\gamma} \le 0. \tag{8}$$

Note the presence of the term $y_1 u\sin\delta$ in the previous equation and how assumption **A.2**. is justified.

2.2.2. *Backstepping the Dynamics*

The above feedback control law applies to the kinematic model of the wheeled robot only. In what follows, using backstepping techniques, that control law is extended to deal with the vehicle dynamics. Notice how in the kinematic design the speed of the robot $u(t)$ was assumed to follow a *desired speed profile*, say $u_d(t)$. In the dynamic design this assumption is dropped, and a feedback control law must be designed so that the tracking error $u(t) - u_d(t)$ approaches zero. Notice also that the robot's angular speed ω_m was assumed to be a control input. This assumption will be lifted by taking into account the vehicle dynamics. Following Krstić et al. (1995) define the virtual control law for $\dot{\theta}$ (desired behaviour of $\dot{\theta}$ in (7)) as

$$\zeta = \dot{\delta} - \gamma\, y_1 u \frac{\sin\theta - \sin\delta}{\theta - \delta} - k_2(\theta - \delta) \tag{9}$$

and let $\epsilon = \dot{\theta} - \zeta$ be the difference between actual and desired values of $\dot{\theta}$. Replacing $\dot{\theta}$ by $\epsilon + \zeta$ in the computation of $\dot{V}_1$ gives

$$\begin{aligned} \dot{V}_1 &= -k_1 s_1^2 + y_1 u \sin\delta - k_2 \frac{(\theta-\delta)^2}{\gamma} \\ &+ \frac{(\theta-\delta)}{\gamma}\epsilon \end{aligned} \tag{10}$$

Augment now the candidate Lyapunov function V_1 with the terms $\epsilon^2/2$ and $(u - u_d)^2/2$ to obtain

$$V_2 = V_1 + \frac{1}{2}\left[\epsilon^2 + (u - u_d)^2\right] \tag{11}$$

with derivative

$$\begin{aligned} \dot{V}_2 &= -k_1 s_1^2 + y_1 u \sin\delta - \frac{1}{\gamma}(\theta-\delta)^2 \\ &+ \epsilon\left(\frac{1}{\gamma}(\theta-\delta) + \dot{\epsilon}\right) + (u-u_d)(\dot{u} - \dot{u}_d). \end{aligned}$$

Simple computations show that if

$$\begin{cases} \dot{\epsilon} = -\frac{1}{\gamma}(\theta-\delta) - k_3\epsilon \\ \dot{u} = \dot{u}_d - k_4(u - u_d), \end{cases} \tag{12}$$

where k_3 and k_4 are positive gains. Then

$$\begin{aligned} \dot{V}_2 &= -k_1 s_1^2 + y_1 u \sin\delta - \frac{1}{\gamma}(\theta-\delta)^2 \\ &- k_3\epsilon^2 - k_4(u-u_d)^2 \le 0. \end{aligned} \tag{13}$$

It is now straightforward to compute the control inputs F and N by solving the dynamics equation (5) to obtain

$$\begin{cases} N = \mathcal{I}\left(f_1(\cdot) - k_3\varepsilon\right) \\ F = \mathcal{M}\left(f_2(\cdot) - k_4(u - u_d)\right) \end{cases} \tag{14}$$

where

$$\begin{cases} f_1(\cdot) = \dot{\zeta} - \frac{1}{\gamma}(\theta-\delta) + c_c\ddot{s} + g_c\dot{s}^2 \\ f_2(\cdot) = \dot{u}_d. \end{cases}$$

The above control law makes $\dot{V}_2$ negative semi-definite. This fact plays an important role in the proof of convergence of the robot to the path.

2.2.3. *Choice of the Approach Angle $\delta(y_1, u)$*

As explained before, the choice of the $\delta(y_1, u)$ approach angle is instrumental in shaping the transient maneuvers during the path approach phase. Indeed, the approach angle is expressing the desired angle to approach the path. In Micaelli and Samsion (1993), the authors propose to use $\delta(y_1, u) = -sign(u)\tanh(y_1)$. This choice is natural in the sense that a large positive lateral distance between the robot and the path, will imply the desired relative heading robot/path to be $\pi/2$, that is the approach angle. As the robot approaches the path, and y_1 diminishes, the approach angle decreases as well. The previous choice is intuitively justified, however, it raises some subtle mathematical difficulties because $\delta(y_1, u)$ is not differentiable with respect to u at $u = 0$. Another choice is possible, $\delta(y_1, u) = -\tanh(y_1 u)$ for instance. This complicates the control derivation, and reduces the system performances in terms of convergence time. In this study, we avoid this problematic situation by imposing a forward velocity $u_d > u_{\min} > 0$ (cf. problem statement of Section 2.1.1, and the implicit respect of the assumption A.3). As we will see later, this condition will also be justified in the next section on obstacle avoidance (Section 3).

2.2.4. *Control Expression*

We now state the proposed solution for the problem exposed in Section 2.1.2.

Proposition 1: *Consider the kinematic and dynamic models described in (4) and (5). Let the approach angle $\delta(y_1, u)$ be*

$$\delta(y_1, u) = -sign(u)\theta_a \tanh(k_\delta y_1) \tag{15}$$

where $0 < \theta_a < \pi/2$ defines the asymptotic desired approach, with k_δ an arbitrary positive gain. Assume that $u_d(t) > u_{\min} > 0$ is a C^2 function. Suppose the path to be followed is parameterized by its curvilinear abscissa s and assume that for each s the variables θ, s_1, y_1, c_c and g_c are well defined (cf. the Remarks section below). Then, the dynamic control law

$$\begin{cases} N_{PF} = \mathcal{I}\left(f_1(\cdot) - k_3\varepsilon\right) \\ F_{PF} = \mathcal{M}\left(f_2(\cdot) - k_4(v - v_d)\right) \\ \dot{s} = u\cos\theta + k_1 s_1 \end{cases} \tag{16}$$

where the $f_1(.)$ and $f_2(.)$ are defines in equation (14) and k_1, k_2, k_3, k_4 and k_δ are arbitrary positive gains, drives y_1, s_1 and θ asymptotically to zero.

Proof: The key steps in the proof can be briefly described as follows: let V_2 be the Lyapunov function candidate expressed in equation (11). The kinematic control expression (16) yields $\dot{V}_2 \leq 0$. Since $\dot{V}_2$ is negative semi-definite and bounded below, V_2 is bounded and has a well defined limit. Therefore, s_1, y_1, θ and u are bounded, since δ and u_d are assumed to be bounded. Considering the previous equations, it is straightforward to show that $\dot{s}_1$, $\dot{y}_1$, $\dot{\theta}$ and $\dot{u}$ are bounded as well. Direct derivation allows one to compute the second derivative of V_2, and prove it is bounded. Therefore, $\dot{V}_2$ is uniformly continuous and application of Barbalat's lemma implies that $\dot{V}_2$ tends to 0 as t tends to ∞. Consider now the expression for $\dot{V}_2$ in (13). Since $\dot{V}_2$ is a sum of negative terms and tends to 0 as t goes to ∞, we conclude that s_1, y_1, $(\theta - \delta)$, ϵ, and $(u - u_d)$ tend to zero as well. That is, the robot asymptotically converges to the path.

Remarks: The proposed solution consists in a 'fox–rabbit' problem where the rabbit is really cooperative since it adjusts its own velocity to help the convergence, through the expression:

$$\dot{s} = u \cos\theta + k_1 s_1.$$

The first term $u\cos\theta$ is the projection of the vehicle (fox) onto the path where the virtual target (rabbit) is located. The second term is necessary to ensure the convergence of s_1 to 0, making the virtual target asymptotically converge to the closest point with respect to the vehicle current position. In the problem statement, the condition $u_d/u_{\min} > 0$ has been imposed (implicitly covering the assumption A.3). Then the vehicle will not stay at rest and will effectively converge to the path. The consequence of this is that the virtual target also will not stay at rest, since θ and s_1 are guaranteed to converge to zero, and $u > 0$, for all t. Then, the path-following problem, as described here can never degenerate to a pose regulation problem, for which Brockett's limitations are inevitable. The assumption that the variables θ, s_1, y_1, c_c and g_c are well defined implies that there is a path parameterization that allows one to compute the path curvature $c_c(s)$ and its spatial derivative $g_c(s)$, for all s denoting the current curvilinear abscissa of the virtual target. Moreover, this implies that the system is equipped with a sensor suite that allows computation of the robot situation in the Serret–Frenet frame, i.e. the variables θ, s_1 and y_1.

3. Obstacle Avoidance Algorithm

This section presents an obstacle avoidance algorithm based on the use of a continuous *Deformable Virtual Zone* (DVZ). The main idea is to define the robot/environment interaction as a DVZ surrounding the vehicle. Deformation of this *risk zone* Ξ is due to the intrusion of proximity information and thus controls the robot reactions. This DVZ characterizes the deformable zone geometry, and depends on the robot velocities (forward and rotational velocities, u and r). Briefly, the risk zone, disturbed by obstacle intrusion, can be reformed by acting on the robot velocities. For a complete exposition of the DVZ principle, the reader is referred to Zapata (2004).

3.1. DVZ Principle

General statement

A rigid body R representing a controlled robot is moving among obstacles in $\mathbb{R}^2$. The 2-dimensional vector $U = [u \;\; r]^T$ is the generalized momenta consisting of the translational and rotational velocities of the robot. It characterizes the motion of R. Furthermore, we will assume that the robot R can be controlled by the derivative $\phi = \dot{U}$ of this vector.

Informally, we define a *controlled DVZ*, Ξ_h, as any DVZ which depends on the vector U characterizing the motion of R. This functional dependence will appear more clearly in Section 3.2, after parameterization of the DVZ:

$$\Xi_h = \rho(U). \tag{17}$$

If we let $\mathcal{P} = (\Xi_h, \Xi)$ be an ordered pair of two DVZ of R (the first one being a controlled DVZ), we define the deformation Δ of the DVZ Ξ_h with respect to Ξ as the functional difference of Ξ and Ξ_h:

$$\Delta = \Xi - \Xi_h. \tag{18}$$

We assume that the robot can perceive distances in all directions of space. We also assume that the set of maximum distances that can be perceived by R and the set of actually perceived distances are also two DVZ surrounding R respectively named *Sensor boundary* and *Information boundary* (and respectively denoted Θ and Ψ). The deformation I of Θ with respect to Ψ is given by $I = \Psi - \Theta$.

Let Ξ_I, be a DVZ which depends on the *Sensor boundary* deformation I:

$$\Xi_I = \beta(I). \tag{19}$$

The deformation Δ of the DVZ Ξ_h with respect to Ξ_I can be written:

$$\Delta = \Xi_I - \Xi_h = \beta(I) - \rho(U). \tag{20}$$

For a given point M on R, the deformation vector $\Delta(M)$ depends on the intrusion of proximity information $I(M)$, in the rigid body workspace, and on the controlled DVZ Ξ_h.

Figure 2 shows these different DVZ.

Parameterization

Equation (20) is a functional equation that has to be parameterized in order to be differentiable and to lead to the "usable"

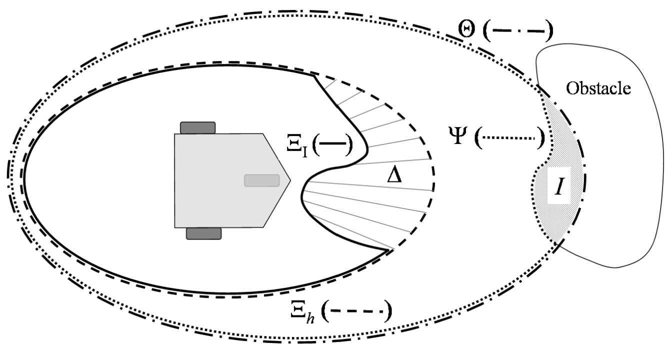
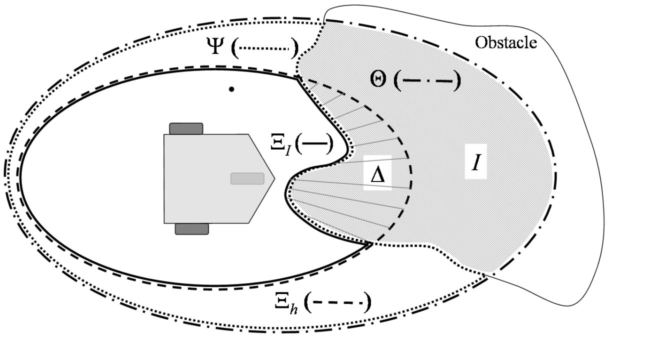


Fig. 2. DVZ principle: the different DVZ for collision avoidance.



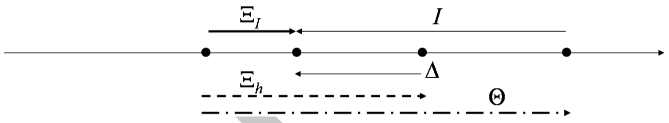


Fig. 3. Implementation of the function β (and projection).

equation (22). For instance, it can be spatially sampled along the directions of the real sensors. In this case, it becomes an s-dimensional vectorial equation where s is the number of proximity sensors. It can also be evaluated as a polar signature and therefore parameterized. In the case of simple mobile robots, the DVZ can be parameterized by simple shapes like ellipsoids (see 3.2).

Differentiation

By differentiating Equation (20) with respect to time, we get:

$$\dot{\Delta} = -\nabla_U[\rho]\,\phi + \nabla_I[\beta]\,\psi \tag{21}$$

where ∇_U and ∇_I are the differentiation operators with respect to the vectorial variables U and I. We note $\phi = \dot{U}$ and $\psi = \dot{I}$.

This equation can be rewritten as:

$$\dot{\Delta} = A\phi + B\psi. \tag{22}$$

Variations in Δ are controlled by a 2-fold input vector $u = [\phi \quad \psi]^T$. The first control vector ϕ, due to the robot controller, tends to minimize deformation of the DVZ. The second one, $\psi = \dot{I}$, is unknown and induced by the environment itself, the relative motion of obstacles with respect to the robot and their shapes. It can be seen as an uncontrolled input.

In the rest of the paper, we will assume that the function β relating the intrusion of information to the deformed DVZ Ξ_I is defined as follows:

$$\beta(I) = \begin{cases} \Theta + I & if\ \Theta + I < \Xi_h \\ 0 & otherwise \end{cases} \tag{23}$$

where the sign $<$ means the comparison of the sensor information ($\Psi = \Theta + I$) and the undeformed DVZ Ξ_h in each direction of measure (see Figure 3). This restriction does not change the principle of the DVZ and is just a simplification leading to confusion of the intrusion I and the deformation Δ (their derivatives are now identical).

3.2. The Controlled DVZ

In order to acquire an analytical expression for the polar signature of the controlled DVZ, expressed in the robot frame centered on R, we consider an elliptic shape. Let $P = [x \quad y]^T$ be a point on the ellipse with axis c_x and c_y. If we assume that the proper reference frame of the DVZ is translated from its center by a vector $a = [a_x \quad a_y]^T$, its equation in the robot frame is given by:

$$\left(\frac{x + a_x}{c_x}\right)^2 + \left(\frac{y + a_y}{c_y}\right)^2 = 1. \tag{24}$$

The coefficients a_x, a_y, c_x and c_y are chosen heuristically and depend on the first component of the control vector, i.e. the general momentum u, the translational velocity of the robot. We have chosen:

$$\begin{aligned} c_x &= \lambda_{cx} u^2 + c_x^{\min} \\ c_y &= \frac{\sqrt{5}}{3} c_x \\ a_x &= -(2/3) c_x \\ a_y &= 0. \end{aligned} \tag{25}$$

In the direction α with respect to the main axis of the ellipse, let $d_h(\alpha)$ be the length of the controlled DVZ Ξ_h, i.e. the norm of the vector $\overrightarrow{RP}$. We can write:

$$\overrightarrow{RP} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos(\theta_m) & \sin(\theta_m) \\ -\sin(\theta_m) & \cos(\theta_m) \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} \tag{26}$$

where $[X \quad Y]^T$ are the coordinates of point P in the absolute frame.

Using the definition of $d_h(\alpha) = \sqrt{x^2 + y^2}$, we can write:

$$\begin{bmatrix} x \\ y \end{bmatrix} = d_h(\alpha) \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \tag{27}$$

where $[v_1 \;\; v_2]^T = [\cos(\alpha) \;\; \sin(\alpha)]^T$ is the unit vector in the direction α, expressed in the robot frame. Substituting equation (27) and equation (26) in equation (24) leads to the quadratic equation:

$$A d_h(\alpha)^2 + B d_h(\alpha) + C = 0 \tag{28}$$

with:

$$\begin{aligned} A &= v_1^2 c_y^2 \cos^2(\alpha) + v_2^2 c_x^2 \sin^2(\alpha) \\ B &= 2a_x v_1 c_y^2 \cos(\alpha) + 2a_y v_2 c_x^2 \sin(\alpha) \\ C &= (a_x c_y)^2 + (a_y c_x)^2 - c_x^2 c_y^2. \end{aligned} \tag{29}$$

The solution of equation 28 is:

$$d_h(\alpha) = \frac{-B + \sqrt{B^2 - 4AC}}{2A} \tag{30}$$

This distance clearly depends on the form parameters of the controlled DVZ, i.e. the coefficients a_x, a_y, c_x and c_y. Less explicitly, it also depends on the orientation of the DVZ, i.e. on the attitude θ_m of the robot. In order to compute the derivative of $d_h(\alpha)$ with respect to this angle θ_m (as we will need below), we have to consider that the point P is fixed in the absolute frame, leading to the dependence of vector v on θ_m. We have:

$$\frac{dv}{d\theta_m} = \begin{bmatrix} \sin(\alpha) \\ -\cos(\alpha) \end{bmatrix} \tag{31}$$

Another way to say this is that the obstacle in this direction (if it exists) is ground-referenced (Figure 4).

3.3. The Deformed DVZ

Let $c(\alpha)$ be the distance between the sensor and the obstacle in the direction α, and $d(\alpha)$ the “distance” between the sensor and the deformed DVZ Ξ_I. $d(\alpha)$ is obtained by saturation of $c(\alpha)$ according to the following equation:

$$\begin{aligned} d(\alpha) &= c(\alpha) : if : c(\alpha) < d_h, \\ &= d_h(\alpha) :: elsewhere. \end{aligned} \tag{32}$$

3.4. The Deformation

In order to use the DVZ paradigm to control the robot collision avoidance and to combine this method with our path-following approach, we introduce the *intrusion ratio*:

$$\Upsilon = \int_{\alpha=0}^{2\pi} \frac{\Delta_\alpha}{d(\alpha)} d\alpha = \int_{\alpha=0}^{2\pi} \frac{d_h(\alpha) - d(\alpha)}{d(\alpha)} d\alpha \tag{33}$$

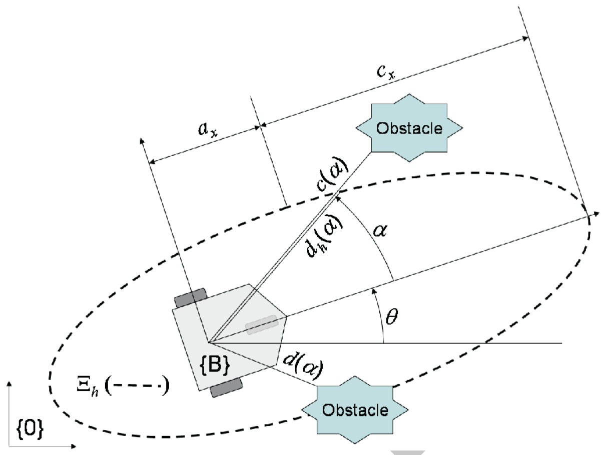


Fig. 4. Frames definition and problem pose description.

where Δ_α is the deformation of the DVZ in the direction α.

This quantifies the global amount of intrusion, normalized by the factor $d(\alpha)$. Note that a null distance robot/obstacle yields an infinite value of Υ. Thus a control that guarantees a bounded intrusion ratio at any time, ensures that the system avoids any obstacle.

The intrusion ratio Υ is a function of u (through the coefficients a_x, a_y, c_x and c_y), of θ (through equation 31) and of the absolute position of the detected obstacle ($[x_E \;\; y_E]$). We can write:

$$\Upsilon = \Upsilon(u, \theta_m, x_E, y_E). \tag{34}$$

3.5. System Jacobian Functions

The time derivation of the expression of the intrusion ratio yields

$$\dot{\Upsilon} = J_\Upsilon^u \dot{u} + J_\Upsilon^\theta r + F^{Rob.vel} u + F^{Obst.vel} \tag{35}$$

with

$$\left\{ \begin{aligned} J_\Upsilon^u &= \int_0^{2\pi} \tfrac{1}{d(\alpha)} \left(\tfrac{1}{2A} \left[-J_B^u + \tfrac{B J_B^u - 2C J_A^u - 2A J_C^u}{\sqrt{B^2 - 4AC}} \right] \right. \\ &- \left. \tfrac{-B + \sqrt{B^2 - 4AC}}{2A^2} J_A^u \right) d\alpha \\ J_\Upsilon^\theta &= \int_0^{2\pi} \tfrac{-1}{d(\alpha)} \left(\tfrac{1}{2A} \left[-J_B^\gamma + \tfrac{B J_B^\theta - 2C J_A^\theta}{\sqrt{B^2 - 4AC}} \right] \right. \\ &- \left. \tfrac{-B + \sqrt{B^2 - 4AC}}{2A^2} J_A^\theta \right) d\alpha \\ F^{Rob.vel} &= \int_0^{2\pi} \tfrac{d_h(\alpha)}{d^2(\alpha)} \cos \alpha \, d\alpha \\ F^{Obst.vel} &= \int_0^{2\pi} \tfrac{d_h(\alpha)}{d^2(\alpha)} (-[\dot{x}_E \cos(\theta_m + \alpha) \\ &+ \dot{y}_E \sin(\theta_m + \alpha)]) \, d\alpha \end{aligned} \right. \tag{36}$$

where

$$\begin{cases} J_A^u &= 2(c_y \frac{\partial c_y}{\partial u} \cos^2(\alpha) + c_x \frac{\partial c_x}{\partial u} \sin^2(\alpha)) \\ J_A^\theta &= 2\cos(\alpha)\sin(\alpha)(c_y^2 - c_x^2) \end{cases} \quad (37)$$

$$\begin{cases} J_B^u &= 2\left(\cos(\alpha)[c_y^2 \frac{\partial a_x}{\partial u} + 2a_x c_y \frac{\partial c_y}{\partial u}]\right. \\ &+ \left.\sin(\alpha)[c_x^2 \frac{\partial a_y}{\partial u} + 2a_y c_x \frac{\partial c_x}{\partial u}]\right) \\ J_B^\gamma &= 2\left(a_x c_y^2 \sin(\alpha) - a_y c_x^2 \cos(\alpha)\right) \end{cases} \quad (38)$$

and

$$\begin{aligned} J_C^u &= 2\left(a_x c_y \left[c_y \frac{\partial a_x}{\partial u} + a_x \frac{\partial c_y}{\partial u}\right]\right. \\ &+ a_y c_x \left[c_x \frac{\partial a_y}{\partial u} + a_y \frac{\partial c_x}{\partial u}\right] \\ &- \left. c_x c_y \left[c_x \frac{\partial c_y}{\partial u} + c_y \frac{\partial c_x}{\partial u}\right]\right) \end{aligned} \quad (39)$$

and $\dot{x}_E$, $\dot{y}_E$ define the obstacle absolute velocity. The assumption of static obstacles yields $F^{Obst.vel} = 0$. For the sake of simplicity, we consider only static obstacles.

3.6. Obstacle Avoidance Control Design

Consider the following Lyapunov function candidate $V_\Upsilon = \frac{I^2}{2}$. The derivation yields

$$\dot{V}_\Upsilon = \Upsilon(J_\Upsilon^u \dot{u} + J_\Upsilon^\theta r + F^{Rob.vel} u). \quad (40)$$

It is straightforward to see that the choice

$$\begin{cases} \dot{u} = -K_u J_\Upsilon^u \Upsilon - \frac{F^{Rob.vel}}{J_\Upsilon^u} u \\ r = -K_r J_\Upsilon^\theta \Upsilon \end{cases} \quad (41)$$

where K_u, K_r are arbitrary positive gains yields $\dot{V}_\Upsilon \leq 0 \forall t$. Note that tedious but straightforward computation (using for the sake of simplicity the guidance functions in (25)) shows that the term $\frac{F^{Rob.vel}}{J_\Upsilon^u}$ is a positive function that acts as a nonlinear damping term. This confirms the well posedness of the expression (41).

The control (41) is a hybrid kinematic/dynamic solution. A backstepping step is necessary for the torque control. Let $V_{OA} = \frac{1}{2}(r_d - r)^2$ be a Lyapunov function candidate, where $r_d = -K_r J_\Upsilon^\theta \Upsilon$. The choice $\dot{r} = \dot{r}_d - K_r(r - r_d)$ yields $\dot{V}_{OA} \leq 0$. Then the dynamic obstacle avoidance control is written as

$$\begin{cases} F_{OA} = \mathcal{M}\left(-K_u J_\Upsilon^u \Upsilon - \frac{F^{Rob.vel}}{J_\Upsilon^u} u\right) \\ N_{OA} = \mathcal{I}(\dot{r}_d - K_r(r - r_d)) \end{cases} \quad (42)$$

where K_r and K_u are arbitrary positive gains, $\mathcal{M}$ and $\mathcal{I}$ are the mass and moment of inertia of the vehicle, and $r_d = -K_r J_\Upsilon^\theta I$.

4. Combining Path-following and Obstacle Avoidance

4.1. Mathematical Inspiration

The combination of the two algorithms is solved as a guidance problem. The requirements are:

- the vehicle should remain far from the obstacle, i.e. in the presence of obstacles Υ has to be bounded at any time,
- when there is no obstacle, the vehicle has to asymptotically converge to the desired path.

To do so, we rewrite the obstacle avoidance control, adding a desired intrusion ratio $\Upsilon_d = K_d \tanh(\lambda_d(\theta - \delta))$, where K_d and λ_d are arbitrary positive gains, θ and δ are path-following variables defined in the previous section (equations (4) and (15)). Let $V_2 = \frac{(\Upsilon - \Upsilon_d)^2}{2}$ be a Lyapunov function candidate. It is straightforward to see that the choice

$$\begin{cases} r = -K_r(\Upsilon - \Upsilon_d)(J_\Upsilon^\theta - \Upsilon_d') + c_c \dot{s} + \dot{\delta} \\ \dot{u} = -K_u J_\Upsilon^u (\Upsilon - \Upsilon_d) - \frac{F^{Rob.vel}}{J_\Upsilon^u} u \end{cases} \quad (43)$$

implies that $\Upsilon - \Upsilon_d$ asymptotically converges to a bounded set defined as:

$$|\Upsilon - \Upsilon_d|_{t\to\infty} < \frac{J_\Upsilon^\theta (c_c \dot{s} + \dot{\delta})}{K_r (J_\Upsilon^\theta - \Upsilon_d')^2 + K_u (J_\Upsilon^u)^2} \quad (44)$$

where $\Upsilon_d' = \frac{2K_d \lambda_d}{1 + \lambda_d(\theta - \delta)^2}$.

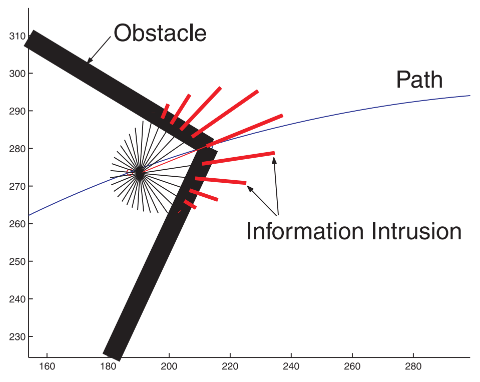
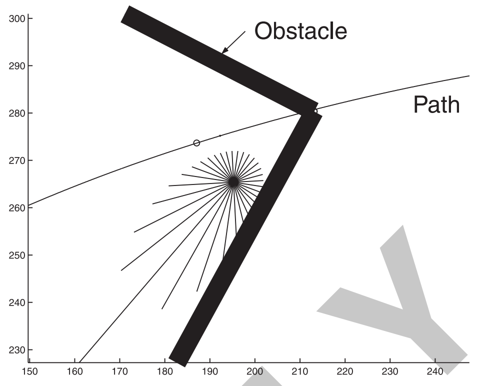
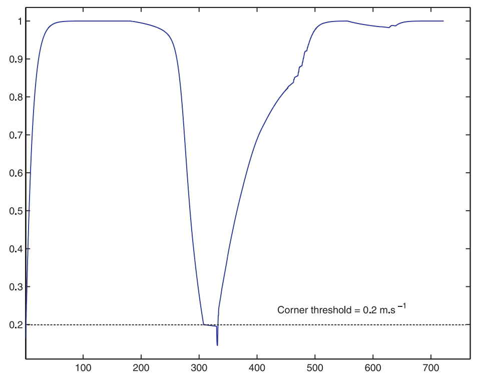
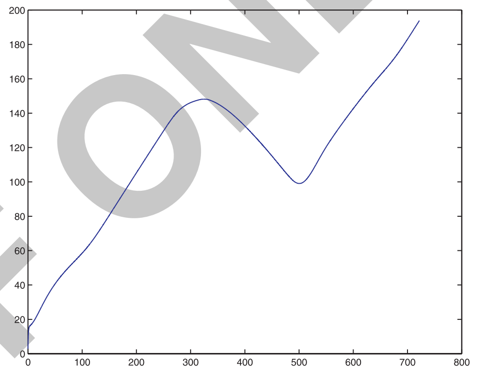
Note that the previous expression is bounded since $(J_\Upsilon^u)^2 > 0$ if $u \neq 0$, Υ_d' is bounded and if the quantity $c_c \dot{s} + \dot{\delta}$ is assumed to be bounded, which is covered by the assumption that the free space is connected.

Then, following the obstacle, the vehicle cannot be driven infinitely far from the desired path, i.e. $c_c \dot{s} + \dot{\delta}$ remains bounded. Moreover, if there is no obstacle, the Lyapunov function candidate degenerates to $V_{\Upsilon=0} = \frac{\Upsilon_d^2}{2}$, and the previous control choice yields $\dot{V}_{\Upsilon=0} = -K_r \Upsilon_d^2 \Upsilon_d'^2 \leq 0$, which induces the path-following asymptotic convergence requirements. The dynamic control corresponding to the previous solution is:

$$\begin{cases} F_{OA} = \mathcal{M}\left(-K_u J_\Upsilon^u (\Upsilon - \Upsilon_d) - \frac{F^{Rob.vel}}{J_\Upsilon^u} u\right) \\ N_{OA} = \mathcal{I}(\dot{r}_d - K_r(r - r_d)) \end{cases} \quad (45)$$

where $r_d = -K_r(\Upsilon - \Upsilon_d)(J_\Upsilon^\theta - \Upsilon_d') + c_c \dot{s} + \dot{\delta}$, K_u and K_r are arbitrary positive gains. At this stage one should note that this solution does not prevent the forward velocity u from being null, and thus a poor convergence rate around the path. We therefore propose another control version that avoids these drawbacks, but also loses any mathematical proof of convergence.

Table 1. Corner situation.

a) Corner situation	b) Extract. from Corner Sit.
	
c) Evolut. forward vel.	d) Evolut. of the 'rabbit'.
	

4.2. *Practical Solution*

The combination of path following and obstacle avoidance capabilities has a natural limitation in situations where both criteria induce antagonist system reactions.

For instance, the obstacle avoidance algorithm may demand that the forward velocity be reduced to zero or a negative value.

This situation occurs when the deformed DVZ Ξ is symmetric with respect to the forward velocity direction. Then the robot will orient itself to minimize the intrusion, and regulate its forward velocity to reshape the nominal DVZ Ξ_h in order to respect the intrusion requirement, i.e. $lim_{t\to\infty}\Upsilon = \Upsilon_d$. This situation is called a *corner situation*, as described in Table 1. To avoid this problematic local minimum, the following switching scheme was designed. Let Υ_l and Υ_r be the intrusion ratios on the left ($0 < \alpha < \pi$) and right ($\pi < \alpha < 2\pi$) side, respectively, and $u_{\min} > 0$ the lowest admissible forward velocity. The chosen corner situation detection is made with the following switching condition. The Boolean variable $CORNER$ is initialized to zero, then

$$
\begin{aligned}
&if\ [(\Upsilon_l\Upsilon_r > 0)\&(u < u_{\min})]\,\{CORNER &= 1\}\\
&if\ (\Upsilon_l\Upsilon_r = 0)\,\{CORNER &= 0\}.
\end{aligned} \quad (46)
$$

The reaction to this situation is to:

(i) reduce the forward velocity,

(ii) rotate until the obstacle is present only on one side, using the following controllers.

$$
\begin{aligned}
\dot{u}_{CORNER} &= -K_u u\\
r_{CORNER} &= r_C sign(\Upsilon_d)
\end{aligned} \quad (47)
$$

where K_u is a positive gain, and r_C the chosen rotational velocity to operate the corner extraction.

The overall control algorithm is written

$$
\begin{aligned}
&if\ (CORNER = 0)\\
\{N &= N_{OA} + f(\Upsilon)N_{PF};\ F = F_{OA} + f(\Upsilon)F_{PF}\}\\
&else\\
\{N &= \mathcal{I}(-K_r(r - r_{CORNER}))\,;\ F = \mathcal{M}\dot{u}_{CORNER}\}
\end{aligned} \quad (48)
$$

where the selection function is $f(\Upsilon) = \frac{1}{1+K_\Upsilon\Upsilon}$, with K_I a positive gain. Note that this algorithm causes the robot to travel with a positive forward velocity $u > u_{\min} > 0$, outside the *corner situation*. This guarantees the well posedness of the expressions (43), (44) and (45). Then the global control (48) is well posed in any situation. Table 1(c) describes the evolution of the forward velocity while encountering a *corner situation*. The increase in the intrusion ratio induces a decreasing forward velocity, down to $u_{\min} = 0.2 m.s^{-1}$, while the obstacle is present on the left and right side of the robot ($\Upsilon_l \Upsilon_r > 0$). The robot is then controlled according to (47), where r_{corner} is driven by the sign of Υ_d. This allows the system to be driven away from the path, thus guaranteeing the system skirts the obstacle. Table 1(d) shows the evolution of the virtual target ('rabbit') along the path, where its backward movement towards the closest point on the path is seen, while the robot is skirting the obstacle.

5. Results

The previous solution implicitly requires an estimation of $\dot{\Upsilon}$ in the computation of F_{OA}. The numerical derivation of the sensor information induces noise amplification and will not provide an accurate estimation of $\dot{\Upsilon}$. Therefore, we degrade the previous solution at a kinematic level, that does not require the estimation of $\dot{\Upsilon}$. Moreover, the considered robot (cf. Figure 5) accepts as control inputs the desired velocities u and r, and its rapid actuator response (combined with its light mass and inertia) allows implementation of a kinematic controller without degrading the overall performance.

$$\begin{cases} r = r_{OA} + f(\Upsilon) r_{pf} \\ u = u_{OA} + f(\Upsilon) u_{pf} \end{cases} \tag{49}$$

where

$$\begin{cases} r_{PF} = \dot{\delta} - K_r^{PF}(\theta - \delta) + c_c \dot{s} \\ u_{PF} = \int_0^t (\dot{u}_d - K_u^{PF}(u - u_d)) dt \\ \dot{s} = u \cos\theta + k_s s_1 \\ r_{OA} = -K_r^{OA}(\Upsilon - \Upsilon_d)(J_r - \Upsilon_d') + c_c \dot{s} + \dot{\delta} \\ u_{OA} = \int_0^t (-K_u^{OA} J_u^{\Upsilon}(\Upsilon - \Upsilon_d) - \frac{F^{rob.vel}}{J_u^{\Upsilon}} u) dt. \end{cases} \tag{50}$$

The DVZ guidance functions are chosen according to (25). The other functions are chosen according to the following definitions:

$$\begin{cases} \Upsilon_d = K_d \tanh \lambda_d (\theta - \delta) \\ f(\Upsilon) = \frac{1}{1 + K_{\Upsilon} \Upsilon} \end{cases} \tag{51}$$

Table 2. The chosen control parameters.

$K_r^{PF} = 0.1$	$K_u^{PF} = 1$	$K_r^{OA} = 0.01$	$K_u^{OA} = 0.1$
$K_d = 10$	$\lambda_d = 0.01$	$K_I = 100$	$u_d = 1$
$\lambda_{cx} = 100$	$c_x^{min} = 10$	$K_u^{CORNER} = 0.1$	$r_c = 1$

5.1. Simulation Results

We have implemented the algorithm on a unicycle simulator developed with Matlab. The control parameters are chosen according to Table 2. The results are displayed in Table 3. Table 2(a) shows the path and obstacles definitions. The robot is of the unicycle type, on which a 32-proximity-sensors belt is mounted, displayed as surrounding radial rays of length defined by the nominal DVZ. Table 2(b) indicates a corner situation, in which we see the reduction of the DVZ due to the decreased forward velocity. Table 2(c) shows the system after the switch from *corner situation* to nominal control. Finally, Table 3(d) displays the global trajectory the system has made.

5.2. Experimental Results

We have implemented the control algorithm of equations (49) on the Pekee robot from the Wany company. This robot is driven by two independent wheels and carries a 16-infrared-proximity- sensors belt, see figure 5, and is controlled via a Mitsubishi M16C family micro-controller. The complete algorithm, equations (49), was first implemented on an embedded PC daughterboard, and the test results are displayed in Figure 6 describing the odometric trajectory of the robot. Since the navigation system is based only on odometric information, the real trajectory is not significant in the demonstration of the validity of our approach. The robot clearly avoids the obstacle and returns to the nominal path without requiring switching control. This is valid in this particular situation where the *corner situation* does not occur. Some chattering behavior has to be filtered out by gain tuning according to the Pekee robot capabilities. Figure 7 shows the evolution of the forward velocity. The black dots denote the desired velocity when no obstacle is detected, the stars indicates the evolution of the forward velocity in the presence of obstacles. A second stage consisted in simplifying the algorithm complexity (including trigonometric look-up tables) in order to reduce the computational burden, and implementation of the solution directly on the M16C micro-controller onboard the vehicle. The tests displayed similar vehicle behavior, since the sensor (US and odometry) accuracy did not allow demonstration of the reactivity benefits of the low level implementation.

Table 3. Simulation results.

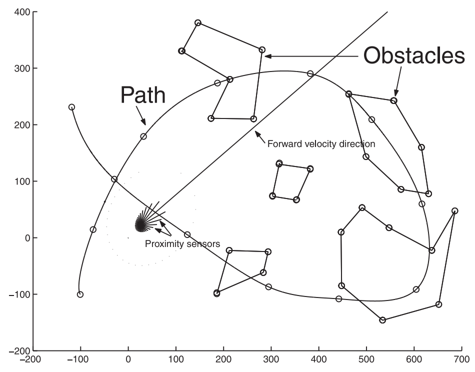
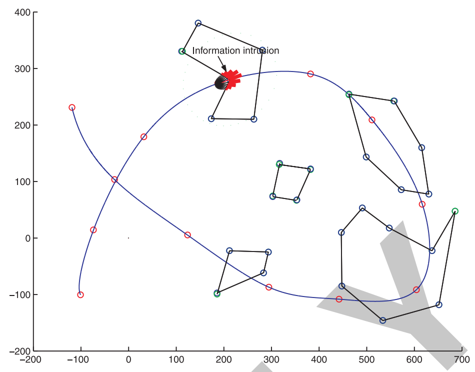
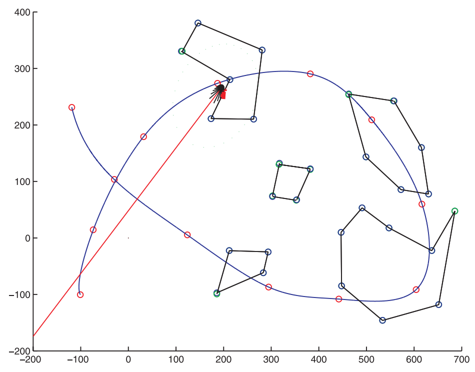
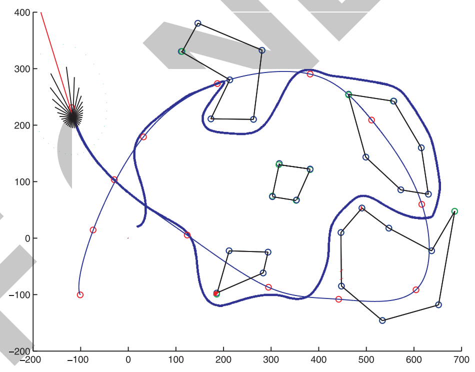
a) Path and Obst. Def.	b) Corner Situation
	
c) Extract. from Corner Sit.	**d) Final Trajectory**
	

Fig. 5. The Wany robot “Pekee”.

6. Conclusions

We have designed a combined path following and obstacle avoidance control law for a unicycle type robot based on the use of the DVZ concept and on a Lyapunov and backstepping design. Implementation of this solution on the Wany robot “Pekee” illustrates an interesting performance, avoiding unnecessary *hot* switches when the vehicle travels with an important forward velocity. With this system, we combine the reactivity of the DVZ principle with a path-following control without requiring any path replanning. The next step in this study is to intrinsically attribute the reactivity of the guidance system to the path-following virtual target, explicitly controlling the evolution of an added virtual state of the virtual target, along the y_1 direction (see Figure 1). Then the main robot control could be designed as a tracker of a cooperative virtual target; note that the system reactivity is then in charge of a new dynamic guidance system.

Acknowledgements

This work is supported by a post doctoral grant from the French national RNTL project WACIF. For further information on this project, please contact http://intranet.wacif.net/default.aspx .

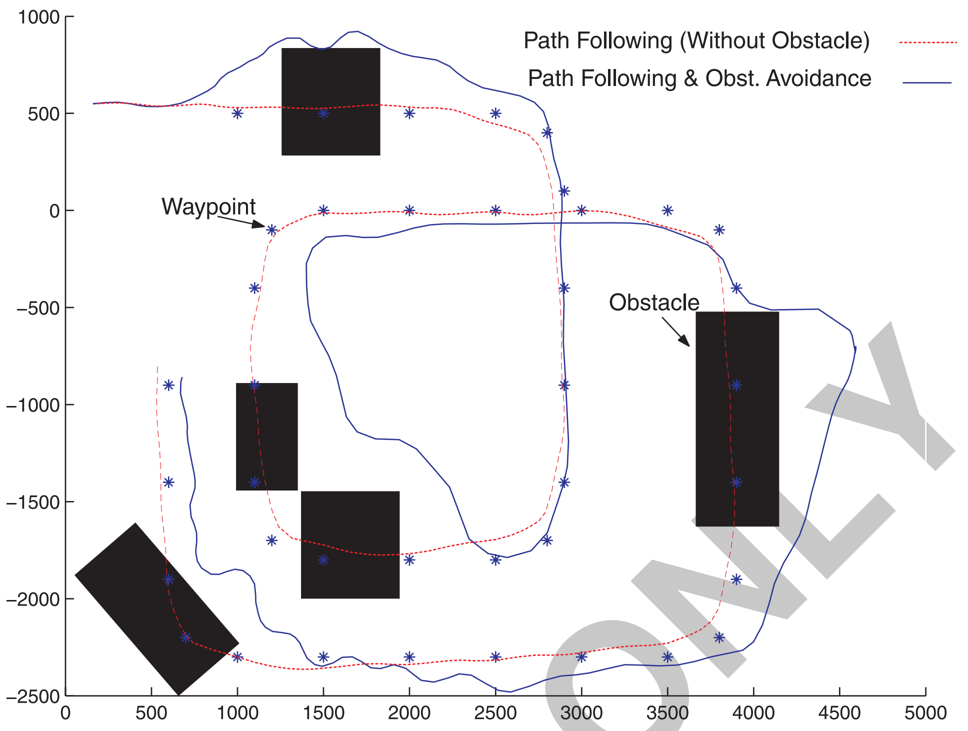


Fig. 6. Experimental results, trajectories.

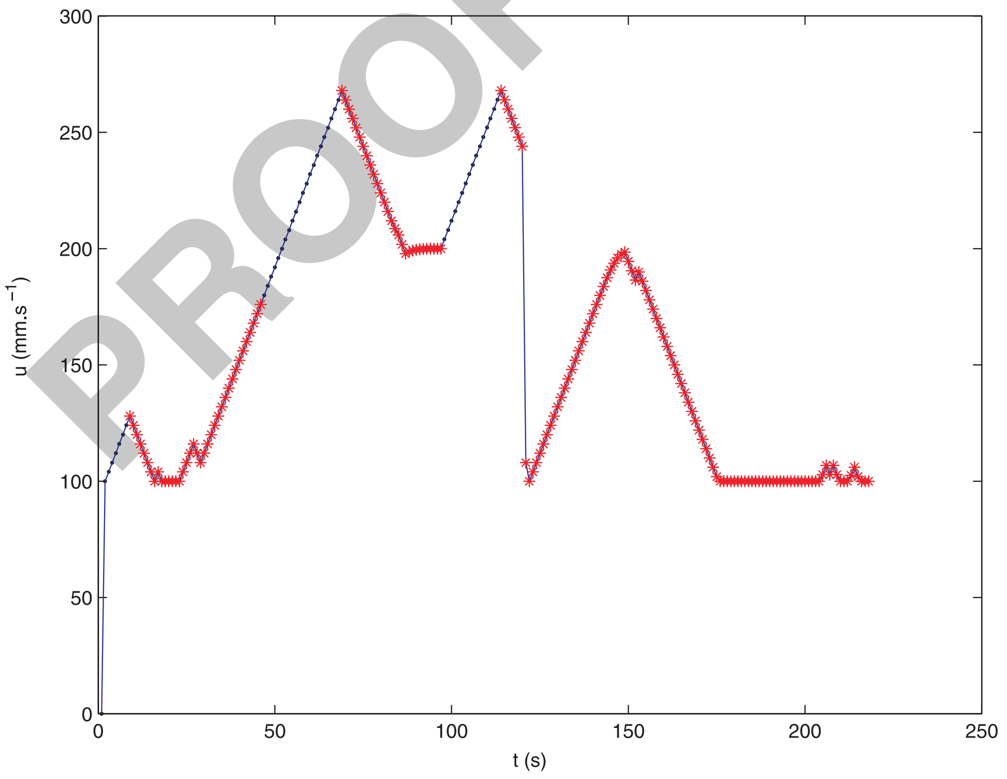


Fig. 7. Experimental results, forward velocity.

References

Aicardi, M., Casalino, G., Bicchi, A., and Balestrino, A. (1995). A closed loop steering of unicycle-like vehicles via Lyapunov techniques. *IEEE Robotics and Automation Magazine*, March, 27–35.

Aicardi, M., Casalino, G., Indiveri, G., Aguiar, P., Encarnacao, P. and Pascoal, A. (201). A planar path following controller for underactuated marine vehicles. *Proceedings of MED'2001*, Dubrovnik, Croatia, June.

Althaus, P. and Christensen, H. (2002). Behavior coordination for navigation in office environment, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland, October.

Arkin, R. C. (1998). *Behavior Based Robotics*. MIT Press, Cambridge, MA.

Canudas de Wit, C., Khennouf, H., Samson, C., and Sordalen, O. J. (1993). Nonlinear control design for mobile robots. In *Recent Trends in Mobile Robotics* (eds Y. F. Zheng), Vol 11, pp.121–156, World Scientific Series in Robotics and Automated Systems. 1993

Elnagar, A. and Hussein, A. (2002). Motion planning using Maxwell's equations. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland, October.

Encarnação, P., Pascoal, A., and Arcak, M. (2000). Path following for marine vehicles in the presence of unknown currents. *Proceedings of SYROCO'2000, 6th IFAC Symposium on Robot Control*, Vienna, Austria.

Encarnação, P. and Pascoal, A. (2000). 3D path following for autonomous underwater vehicle. *Proceedings of CDC'2000, 39th IEEE Conference on Decision and Control*, Sydney, Australia.

Fierro, R. and Lewis, F. (1997). Control of a nonholonomic mobile robot: backstepping kinematics into dynamics. *Journal of Robotic Systems*, **14**(3): 149–163.

Ge, S. and Cui, Y. (2000). Path planning for mobile robots using new potential functions. *Proceedings of the 3rd Asian Control Conference*, Shangai, China, July.

Iniguez, P. and Rossel, J. (2002). A hierarchical and dynamic method to compute harmonic functions for constrained motion planning. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Lausanne,Switzerland, October.

Jiang, Z. P. and Nijmeijer, H. (1999). A recursive technique for tracking control of nonholonomic systems in chained form. *IEEE Transactions on Robotics and Automation*, **44**(2): 265–279.

Krstić, M. I., Kanellakopoulos, and Kokotovic, P. (1995). *Nonlinear and Adaptive Control Design*. John Wiley & Sons, Inc., New York.

Lapierre, L., Soetanto, D., and Pascoal, A. (2003). Nonlinear path following with application to the control of autonomous underwater vehicle. *CDC 2003, 42nd IEEE Conference on Decision and Control*, Maui, Hawaii, USA, December.

Louste, C. (1999). *Conception d'une methode de planification pour robot mobile selon la methode des milieux continus appliquee aux fluides visqueux*. PhD thesis no. 6701, LIRMM, Montpellier, France.

Micaelli, A. and Samson, C. (1993). Trajectory-tracking for unicycle-type and two-steering-wheels mobile robots. Technical Report No. 2097, INRIA, Sophia-Antipolis, November.

Ogren, P. (1989). Real-time obstacle avoidance for fast mobile robots. *IEEE Transactions on Systems, Man and Cybernetics*, **19**(5): 1179–1187.

Ogren, P. and Leonard, N. (2002). A provable convergent dynamic window approach to obstacle avoidance. *IFAC World Congress*, Barcelona, Spain, July.

Ogren, P. (2003). *Formation and Obstacle Avoidance in Mobile Robot Control*. PhD thesis, Stockholm, Norway, June.

Rimon, E. and Koditschek, D. (1992). Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation*, **8**(5): 501–518.

Samson, C. and Ait-Abderrahim, K. (1991). Mobile robot control Part 1: Feedback control of a non-holonomic mobile robot. Technical Report No. 1281, INRIA, Sophia-Antipolis, France, June.

Soetanto, D., Lapierre, L., and Pascoal, A. (2003). Adaptive, nonsingular path following control of dynamic wheeled robot. *CDC'03, 42nd IEEE Conference on Decision and Control*, Maui, Hawaii, USA, December.

Zapata, R., Cacitti, A., and Lepinay, P. (2004). DVZ-based collision avoidance control of non-holonomic mobile manipulators. *European Journal of Automated Systems*, **38**(5): 559–588.