A PATH PLANNING STRATEGY FOR OBSTACLE AVOIDANCE

Guillaume Blanc, Youcef Mezouar and Philippe Martinet
LASMEA
UBP Clermont II, CNRS - UMR6602
24 Avenue des Landais, 63177 Aubiere, France

Keywords: Mobile robot, obstacle avoidance, path planning.

Abstract: This paper presents an obstacle avoidance module dedicated to non-holonomic wheeled mobile robots. Chained system theory and deformable virtual zone principle are coupled to design an original framework based on path following formalism. The proposed strategy allows to correct the control output provided by a navigation module to preserve the robot security while assuring the navigation task. First, local paths and control inputs are derived from the interaction between virtual zones surrounding the robot and obstacles to efficiently prevent from collisions. The resulting control inputs and the ones provided by the navigation module are then adequately merged to ensure the success of the navigation task. Experimental results using a cart-like mobile robot equipped with a sonar sensors belt confirm the relevance of the approach.

1 INTRODUCTION

Whether a mobile robot uses an internal representation of its environment or not, the navigation strategy that it applies can not ignored the risk to meet an unknown obstacle on its way while performing a navigation task. If such an event happens, the robot has to react for the best. On one hand it must preserve its security. On the other hand, it has to perform its main task.

If the robot navigates according to a geometric map, which models the free space as a subset of the configuration space, it is then possible to describe obstacles by their configurations into the map. In this context, a first approach to obstacle avoidance is to consider it as a path planning problem under geometric constraints (Laumond, 1998). The second main approach consists in confering on the robot a reflex behavior without explicitly planning any geometric path or trajectory. Typically, it can be done using the potential field method (Khatib, 1986; Barraquand et al., 1992). In this approach, the robot motions are under the influence of an artificial repulsive potential field pushing the robot away from the obstacles. This strategy does not necessarily require geometric models of the environment and of the obstacles since potential fields can be defined directly from sensor data (sensor-based reflex behavior). Generally, a sensor based approach consits in mapping the robot control inputs with sensor observations through a behavior model. The behavior model can be for instance inspired by neurosciences as proposed in (Braitenberg, 1984; Filliat et al., 1999) or would rather use sensor-based control formalism as proposed in (Samson et al., 1991). This last formalism has been intensively exploited in the context of mobile robotics. However, with such a strategy, it can be difficult to manage the induced kinematic constraints (nonholonomy) in presence of obstacles. This issue is elegantly addressed by Zapata *et al* through the DVZ (*Deformable Virtual Zone*) approach (Zapata et al., 1994). It consists in surrounding the robot with a virtual zone which can be deformed depending on two modes. The first one is called *controlled mode*. The shape of the zone is modified according to the internal state of the robot. The second mode is the *uncontrolled mode* of deformation. The shape of the zone is modified when an obstacle tries to come into it. The control inputs are computed in order to minimize the uncontrolled deformation. This paper focuses on an obstacle avoidance module (OAM) for a nonholonomic mobile robot. The environment and the obstacles are not supposed explicitly modeled. The navigation module is provided by an external algorithm as for instance the one proposed in (Blanc et al., 2005). The OAM thus aims to assist the navigation module by correcting the control inputs to

Blanc G., Mezouar Y. and Martinet P. (2006).
A PATH PLANNING STRATEGY FOR OBSTACLE AVOIDANCE.
In *Proceedings of the Third International Conference on Informatics in Control, Automation and Robotics*, pages 438-444
DOI: 10.5220/0001211604380444

prevent potential collision with obstacles.
In a more general point of view, the OAM can be considered as a security filter acting on the control inputs provided by an indepently designed navigation algorithm. The OAM acts according to the local context of obstacles and consequently, it can be classified as a reflex obstacle avoidance strategy. As it is done with the DVZ, a virtual observation zone surrounding the robot is defined. The shape of this observation zone varies with the instantaneous kinematic state of the robot. The goal is to protect this zone from obstacle intrusions. When an obstacle enters the observation zone, the shape of this zone is deformed to fit the shape of the obstacle. This new shape is considered as a path that should be followed for avoidance. The OAM provides a control vector which should be applied to follow the planned path, and merges it adequately with the one from the navigation module. In an original way, the avoidance strategy consists thus in a reflex action which is performed according to a local path following based control.

2 MODELING

The OAM consists in acting on the control inputs to thrust obstacles aside a zone which virtually surrounds the robot. The main idea is to consider the shape of entering obstacles as a path the robot has to follow at a given distance which depends on the shape of the virtual zone.
The proposed formalism is adequated to non-holonomic mobile robots. It takes advantage of the exact linearization of the mobile robot kinematic model through chained system (Samson, 1995). Without loss of generality, our framework is illustrate with a cart-like mobile robot.

2.1 Robot Model

The cart-like robot can be modeled as a unicycle driven by a control vector composed of two kinematic inputs : $\mathbf{U} = [u_1, u_2]$. An euclidian frame $\mathsf{F}_R \triangleq (O_R, \mathbf{X}_R, \mathbf{Y}_R, \mathbf{Z}_R)$ is attached to the robot. The origin O_R is the unicycle control point and is fixed on the wheels axle's middle point of the cart-like robot. u_1 and u_2 denotes respectively the longitudinal and the rotational velocities of the unicycle. As shown by Figure 1, $\mathbf{X}_R$ carries u_1 whereas u_2 is expressed around $\mathbf{Z}_R$.
The robot is supposed to roll without slipping. Whereas the avoidance strategy is based on a path following formalism, the robot kinematic model is expressed in a Frenet frame F_M. Let M be the orthogonal projection of O_R on a curve ζ. We assume that the distance from O_R to ζ always remains smaller than

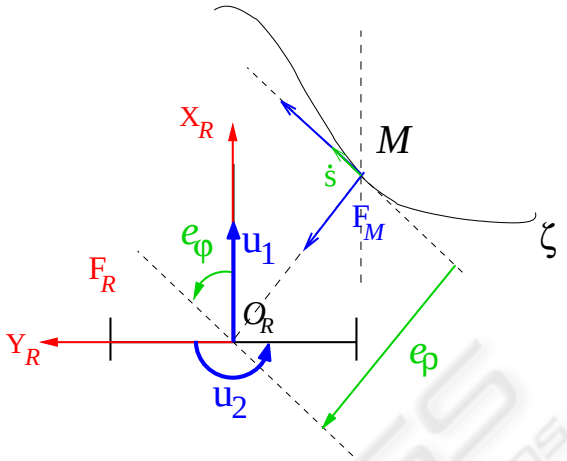

Figure 1: Robot kinematic model in a Frenet frame.

the minimal value of the curvature radius along ζ. In this way, M is unique. F_M is represented on Figure 1, where s is the curvilinear abscissa associated to M on ζ. e_ρ and e_φ are respectively the lateral and the angular deviations of the robot with respect to ζ. Let $c(s)$ be the curvature of ζ at M. The robot motions are described with respect to F_M by the following kinematics equations:

$$\begin{cases} \dot{s} = \dfrac{u_1 \cos e_\varphi}{1 - e_\rho c(s)} \\ \dot{e_\rho} = u_1 \sin e_\varphi \\ \dot{e_\varphi} = u_2 - \dot{s} c(s) \end{cases} \quad (1)$$

2.2 Observation Zone

Let us define a virtual zone, called *observation zone* and surrounding the robot, as a planar curve ζ_z lying on $\Pi \triangleq (O_R, \mathbf{X}_R, \mathbf{Y}_R)$. The polar coordinates of each point on Π will be denoted ρ for the module and θ for the argument.

Let now f_z be a continuous and derivable function:

$$\begin{array}{rccc} f_z & : & & \\ & [-\pi, \pi] & \mapsto & I\!R \\ & \theta & \longrightarrow & \rho = f_z(\theta) \end{array} \quad (2)$$

such that

$$\forall \theta \in [-\pi, \pi] \ , \ \rho > 0$$

f_z clearly defines a virtual zone that surrounds O_R in Π. This virtual zone is a closed curve ζ_z which marks off a set $\mathcal{Z}$ with $O_R \in \mathcal{Z}$ [1]:

$$\zeta_z = inf\{\rho > 0 \mid \rho = f_z(\theta)\} \quad (3)$$

[1] given a set $\mathcal{A}$, $inf\mathcal{A} = max\{t \in [0\ ,\ +\infty] \mid \forall a \in \mathcal{A}\ ,\ t < a\}$

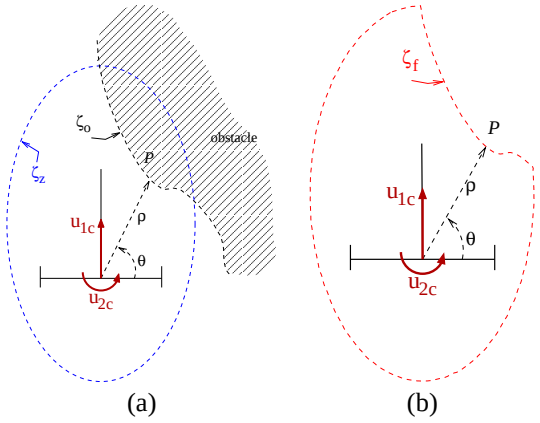

Figure 2: Illustration of the definition of: (a) the observation zone by ζ_z, the obstacle shape by ζ_o ; (b) the free zone by ζ_f.

We assume now that an obstacle is known in F_R as a polar function f_o in Π describing its shape. Let ζ_o be the corresponding curve:

$$\zeta_o = inf\{\rho > 0 \mid \rho = f_o(\theta)\} \quad (4)$$

ζ_o defines the set $\mathcal{O}$ which contains the points lying on the considered obstacle. If we assume that the robot did not collide with the obstacle then $O_R \notin \mathcal{O}$.

Figure 2(a) illustrates above definitions with an elliptic virtual zone and a single obstacle.

2.2.1 Uncontrolled Deformations

When an obstacle gets closer to the robot, it may physically penetrate into the virtual zone. In this case, $\mathcal{Z} \cap \mathcal{O} \neq \emptyset$. We define the *free set* $\mathcal{F}$ as:

$$\mathcal{F} = \mathcal{Z} - \mathcal{O}$$

Now consider the curve ζ_f:

$$\zeta_f = min(\zeta_z, \zeta_o) \quad (5)$$

ζ_f is a closed curve, always defined since ζ_z is also a closed curve and $\mathcal{F}$ is the inner set defined by ζ_f. Note that ζ_f can be considered as the result of the deformation of ζ_z by the uncontrolled intrusion of an obstacle as illustrated by Figure 2(b).

2.2.2 Controlled Deformations

Till now, ζ_z was only considered as a geometric curve which did not depends on time since f_z was fixed. However, it may be judicious to adapt the shape of ζ_z with the navigation context. Remembering that the robot is guided by a navigation module, ζ_z may be deformed in a controlled mode by taking into account the input control vector $\mathbf{U_c} = [u_{1_c}, u_{2_c}]$ provided by this module. Then the definition of f_z (refer to equation (2)) has to be modified: $\rho = f_z(\theta, \mathbf{U}_c)$.
For instance, an appropriated deformation should be to control a scale factor on ζ_z according to u_{1_c}, in order to extend the observation zone when the robot goes fast. If the robot goes slowly, it probably navigates in a cluttered or unknown space. In this case, it has to be reactive to obstacles and its motions should not be too much constrained so that it can carry on making progress.
If we now consider the robot as a volumic object instead of a single point, it appears necessary to constrain the controlled deformations of ζ_z with a low limit. This low limit can be seen as a *security zone* that includes the robot.

2.3 Security Zone

As the observation zone ζ_z (refer to equation (3)), the security zone is defined as a closed curve ζ_s. The associated function f_s is designed in order that ζ_s closely fits the area defined by the orthogonal projection of the robot volume on Π.
ζ_s is not deformable. From a practical point of view, it must be designed according to the minimal range that embedded sensor are able to provide accurately. No obstacle has normally to come within ζ_s, but if unfortunately it happens, the obstacle must be detected to adapt the control strategy. This also implies that ζ_s should be within ζ_z as illustrated in the Figure 3. The controlled deformations of ζ_z are thus limited:

$$\forall\theta \in [-\pi, \pi], \ f_z(\theta, \mathbf{U}_c) \geq f_s(\theta) \quad (6)$$

Seeing its functional interest, ζ_s plays an important rule for the control design.

3 CONTROL DESIGN

3.1 Control Strategy

The avoidance control law aims to drive the robot in order to get $\mathcal{F} = \mathcal{Z}$.
Let M be a point of ζ_f defined as:

$$M \triangleq min\{\zeta_f - \zeta_z\} \quad (7)$$

M is the orthogonal projection of O_R on the piece of curve ζ_f which is not in ζ_z. Note that if $\mathcal{F} = \mathcal{Z}$, $M = \emptyset$. In this case, we consider that the coordinates of M are not defined; otherwise, we note $(\rho_M, \theta_M, 0)$ the coordinates of M in F_R.
Now let us define the error e_ρ which models the deformation from ζ_z to ζ_f by the following signed distance (see Figure 4):

$$e_\rho = f_z(\theta_M, \mathbf{U}_c) - \rho_M \quad (8)$$

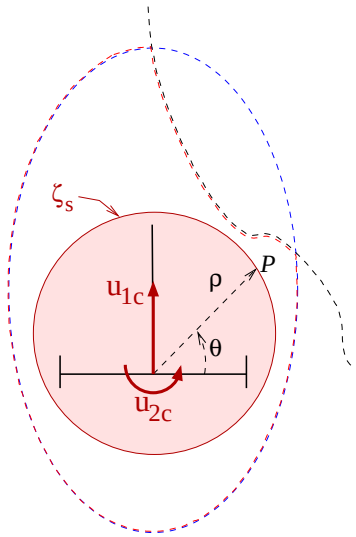

Figure 3: The security zone ζ_s.

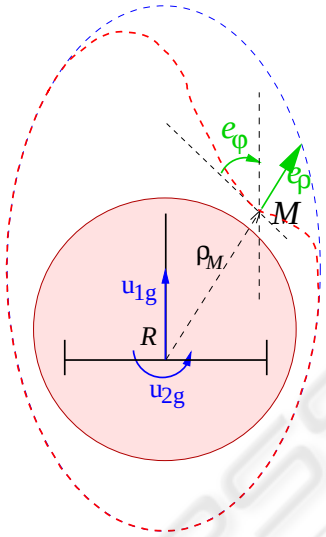

Figure 4: Lateral control: Definition of errors to be regulated for obstacle avoidance.

e_ρ is a lateral deviation which may be regulated to zero to push away the obstacles from the virtual zone. Let us also define the angular deviation e_φ with respect to the longitudinal axis $\mathbf{X}_R$ as the orientation of the tangent vector to ζ_f at M . The avoidance control objective is to stabilize both e_ρ and e_φ to zero. This defines a path following problem. In the sequel, this issue is addressed using the theoretic framework of chained systems (Samson, 1995).

3.2 Lateral Control

The state space model (1) can be converted into a chained system of dimension three:

$$[a_1, a_2, a_3] = [s, e_\rho, (1 - e_\rho c(s)) \tan e_\varphi] \quad (9)$$

with a two dimensional control vector $[m_1, m_2]^\top \in I\!R^2$. The derivative of such a chained system with respect to time is:

$$\begin{cases} \dot{a_1} = m_1 \\ \dot{a_2} = a_3 m_1 \\ \dot{a_3} = m_2 \end{cases} \quad (10)$$

If this state vector is derivated with respect to a_1, then it appears that resulting kinematic system is linear. A simple control which asymptotically stabilizes a_2 and a_3 thus consists on a proportional state feedback:

$$m_2 = -m_1 K_p a_2 - \mid m_1 \mid K_d a_3 \ , \quad (K_p, K_d) \in I\!R^{2+*} \quad (11)$$

$(K_p,\, K_d)$ determines the performances of the controller and defines a settling distance for the regulation of a_2 and a_3 to zero. These two gains control the reactivity of the robot when avoiding an obstacle.

In equation (11), m_1 appears as a parameter which does not influence the regulation dynamic. m_1 is directly linked to u_1 by:

$$m_1 = \dot{s} = u_1 \cos(e_\varphi) \quad (12)$$

The system (10) leads to uncouple the controls m_1 and m_2. This implies that u_1 and u_2 are also uncoupled. The rotational velocity resulting from the lateral regulation of the robot with respect to the obstacle will be denoted u_{2_z} in the sequel. Independently from u_{2_z}, a longitudinal velocity u_{1_z} can be used to adapt the longitudinal velocity u_{1_c} to the navigation context.

3.3 Longitudinal Control

The value of the longitudinal velocity does not *a priori* influence the lateral regulation performances. Then, u_{1_z} could take any value u_{1_c} provided by the navigation module without any dammage. From a practical point of view, it is advantageous to reduce it when the environment seems to become cluttered. Firstly, this ensures that the robot can stop if an obstacle is detected close to the frontier of the security zone. Secondly, this allows to get more accurate and stable data from sensors when obstacles come closer to the robot. The strategy for longitudinal control consists on applying an adaptative gain α on u_{1_c} in order to stop the robot if an obstacle is within de security zone, and to get $u_{1_z} = u_{1_c}$ if no obstacle is observed within ϱ_z:

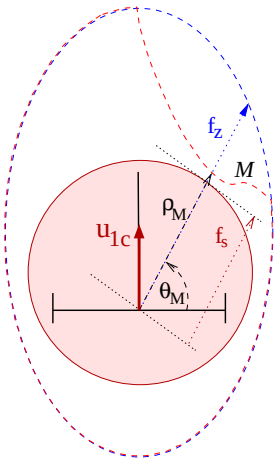

Figure 5: Longitudinal control.

$$\alpha \ : [-\pi, \pi] \mapsto [0, 1]$$

$$\begin{cases} \text{if } M \neq \emptyset & \\ \quad \alpha = \dfrac{\rho_M - f_s\left(\theta_M\right)}{f_z\left(\theta_M\right) - f_s\left(\theta_M\right)} & \text{if } \rho_M \geq f_s\left(\theta_M\right) \\ \quad \alpha = 0 & \text{otherwise} \\ \text{otherwise} & \\ \quad \alpha = 1 & \end{cases} \tag{13}$$

and:

$$u_{1_z} = \alpha \, u_{1_c} \tag{14}$$

Of course, if the robot stops while a dynamic obstacle intends deliberately to run into, the collision will occur. At present, we do not focuses on this problem.

3.4 Global Control

As exposed in sections (3.2) and (3.3), the OAM provides a control vector $\mathbf{U}_z = [u_{1_z}, u_{2_z}]$ which aims to correct $\mathbf{U}_c$ from a navigation module. u_{1_z} is directly linked to u_{1_c}, however the computation of u_{2_z} does not take u_{2_c} into account. The OAM must favor the navigation module outputs if the local configuration of obstacles allows it. One can confere more importance to $\mathbf{U}_z$ when an obstacle draws near the security zone. Let $\mathbf{U}_g = \left[u_{1_g}, u_{1_g}\right]$ be the result of the correction of $\mathbf{U}_c$ by the OAM. $\mathbf{U}_g$ is computed according to the following equations:

$$\begin{cases} u_{1_g} = u_{1_z} & \\ u_{2_g} = \gamma u_{2_c} + (1 - \gamma) u_{2_z} & \end{cases} \tag{15}$$

If γ is taken equal to α, near the frontier of the security zone, the importance given to u_{2_z} would be maximum while u_{1_z} tends to zero. In order to get a more reactive behavior, the definition of f_z is modified as follow

$$\forall \theta \in [-\pi, \pi[\,, \ f_z\left(\theta, \mathbf{U}_c\right) \geq f_s\left(\theta\right) + \varepsilon \tag{16}$$

where ε is a constant scalar which allows to keep a zone around ζ_s when u_{1_z} is non-zero and the robot is exclusively steered by u_{2_z}:

$$\gamma \ : [-\pi, \pi] \mapsto [0, 1]$$

$$\begin{cases} \text{if } M \neq \emptyset & \\ \quad \gamma = \dfrac{\rho_M - f_s\left(\theta_M\right) + \varepsilon}{f_z\left(\theta_M\right) - f_s\left(\theta_M\right) + \varepsilon} & \text{if } \rho_M \geq f_s\left(\theta_M\right) + \varepsilon \\ \quad \gamma = 0 & \text{otherwise} \\ \text{otherwise} & \\ \quad \gamma = 1 & \end{cases} \tag{17}$$

3.5 Singularities

The proposed control design conduces to some singularities which should not be ignored. They result from the lateral control strategy. First, the point M is not necessarily unique. Secondly, the curvature $c(s)$ is not defined if f_z is not two times derivable at θ_M. Last, e_φ may be equal to $\pm\frac{\pi}{2}$. From a practical point of view, the first two singularities can be shunned. Indeed, curves ζ_o and ζ_f result from interpolation of scanning range sensors data. They are not usually treated as analytic curves but as a set of sampled points corrupted by measurement noises. Consequently, M is not chosen among many points. However, if a choice has to be made, criteria can direct it. The closest point to the $\mathbf{X}_R$ axis, which directs the longitudinal robot motion, can be privileged. Furthermore, since curves are sampled with a constant step $\Delta\theta$, the k^{th} derivative f_z^k of f_z are computed as $f_z^k\left(\theta\right) = \frac{f_z^{k-1}(\theta+h) - f_z^{k-1}(\theta-h)}{2h}$, where $h = n\,\Delta\theta$ and n is a strictly positive integer. The curvature $c(s)$ is thus always defined.

The third singularity corresponds to the case where the robot goes orthogonally to an obstacle. The state (9) is no more defined and, as a consequence, not either the control law (11). It is all the more problematical that m_2 should exponentially increase since e_φ tends to $\pm\frac{\pi}{2}$. To avoid such a configuration, one can impose $m_2 = 0$ when e_φ tends to $\pm\frac{\pi}{2}$. In this way, the longitudinal control law prevents the robot from colliding with the obstacle.

4 IMPLEMENTATION

This section illustrates the proposed approach from a practical point of view. The framework proposed in sections (2) and (3) was applied to assist the driving of an indoor mobile robot equipped with a telemetric sensors belt.
The OAM has been implemented on an external standard PC which wireless controls a Pioneer robot. This robot is equipped with a eight sonars belt. Telemetric data are acquired at $\mathrm{f_e} = 10\mathrm{Hz}$. The control inputs are also cadenced at $\mathrm{f_e}$. Telemetric data are collected as eight range values, associated respectively to each sensor. None filtering is processed on these data and none uncertainty on measures is taken into account.

4.1 Virtual Zones

The two virtual zones ϱ_z and ϱ_s are chosen as two concentric circles. The diameter of the security zone is set to $\phi_s = 0.6\mathrm{m}$. In this way, ϱ_s surrounds every sightless zone between two consecutive sensors. The parameter ε is fixed to $0.3\mathrm{m}$. The diameter ϕ_z of ϱ_z linearly depends on u_{1_c}. It is *a priori* limited by the maximum range of sensors (*i.e* $3\mathrm{m}$). However, the maximum value of ϕ_z is set to $\phi_{zmax} = 1\mathrm{m}$ in order to not constrain the robot motion when the obstacles are far away from ϱ_z. Let $u_{1_{cmax}}$ be the maximum longitudinal velocity of the robot then ϕ_z is given by:

$$\phi_z = (\phi_s + 2\varepsilon)\left(1 - \frac{u_{1_c}}{u_{1_cmax}}\right) + \phi_z max\left(\frac{u_{1_c}}{u_{1_cmax}}\right)$$

4.2 Planning of a Straight Line

Although it is possible to interpolate the data at each sample to obtain a smooth curve which roughly delimits the set of obstacles observate by the sonars, we refer the avoidance algorithm to a simple straight line. This line determines ϱ_o. ϱ_o is defined by the point M and a vector $\mathbf{\Delta_o}$. M is obtained from the observations at time k : $\sigma_k = [\sigma_{1_k}, \sigma_{2_k}, \ldots, \sigma_{8_k}]$ as the closest point to O_R within the circle ϱ_z. $\mathbf{\Delta_o}$ depends on the navigation module outputs.

Let $\Omega_c = \frac{u_{2_c}}{\mathrm{f_e}}$ be the instantaneous steering angle which should have directed the robot if it was only guided by the navigation module. The vector $\Omega_c \mathbf{z}_R$, where $\mathbf{z}_R$ is the unitary vector which directs the axis $\mathbf{Z}_R$, represents the directing vector of the instantaneous robot motion. Let σ_{i_k} be the range data holden to estimate the point M. Another range data is needed to determine a second point N to build ϱ_o. The choice falls on σ_{i+1_k} if $\theta_M \geq \Omega_c$ or σ_{i-1_k} otherwise. Then let θ_{MN} be the argument of the vector $\overrightarrow{MN}$. If $\theta_{MN} > \Omega_c$, then $\mathbf{\Delta_o} = \overrightarrow{MN}$. Otherwise, $\mathbf{\Delta_o} = \Omega_c \mathbf{z}_R$. In this way, we take care that the OAM does not servo the robot on a line from which the navigation module would have a tendancy to get the robot out of the way. If M is obtained from σ_{1_k} or σ_{8_k}, N may not exist. Then $\mathbf{\Delta_o}$ is forced to be $\Omega_c \mathbf{z}_R$.

In the case of a straight line following, we have $\forall s, c(s) = 0$ which implies a simplification of the control design exposed in section (3.2). Equation (11) leads to the following simple control law:

$$u_{2_z} = -u_{1_z} \cos e_\varphi{}^3 K_p e_\rho - \mid u_{1_z} \cos e_\varphi{}^3 \mid K_d \tan e_\varphi$$

This law ruled the robot motion during the experiments which results are presented in the next section.

4.3 Results

We have validated the OAM with two kind of experiments.
The first one consists on a very simple robotic task. It consists on going forward, with $\mathbf{U}_c = [v, 0]^\top$, where v is a constant. We have tried for v a range of longitudinal velocities from $0.1\,\mathrm{m.s^{-1}}$ to $0.5\,\mathrm{m.s^{-1}}$, which is a respectable value for an indoor robot. Initially, the robot is placed in order to be directed to a wall, with an incidence angle of approximately $-\frac{\pi}{6}\mathrm{rad}$. The diameter of the circular security zone is fixed to $0.6\mathrm{m}$ while ε is set to $0.3\mathrm{m}$. In these conditions, the OAM works as it is forecast in such a simple case.
The second task is more interesting. The robot is teleoperated in offices and corridors of our laboratory and the operator can not directly see the robot. He only can see the image provided by an embedded frontal camera which has an angle of view of $60\mathrm{deg}$. Images are wireless transfered, what implies some unexpected and non constant delays on the visual feedback. In this case the OAM ensures the robot security and assists the operator. However, circular virtual zones rapidely appear unadequated. When an obstacle comes closer to the robot from a side, the OAM tends to correct the trajectory and unexpected oscillations of the robot trajectory are induced. Elliptic zones should be a solution to this issue.

5 CONCLUSION

This paper has presented the design and an implementation of an obstacle avoidance module (OAM) which aims to act on the robot control inputs in order to preserve the security of the robot when it is driven by a navigation module which does not take explicitely into account the possible presence of unexpected obstacles.
The OAM is based on an original obstacle avoidance

approach. It consists on assigning to the robot a reactive behavior by following a dynamically computed path which results from the local observation of the robot environment. None *a priori* model of obstacles is needed since their shape can be estimated from range data when the robot get closer to them. This estimated shape serves as reference of a path following based control law which aims to pushing away obstacles from a virtual zone. This zone is a closed curve surrounding the robot and its shape can be modified according to control inputs provided by the navigation module.

The approach has been illustrated in a simple case. However, we have put it on the test in much more complex contexts. Helped by the OAM, a user can teleoperate the robot from a different room to the one where the robot is navigating. The user can safely drive the robot when he has only a limited view of the cluttered navigation environment from an embedded camera looking forward.

Although a simple case of implementation, based on circular virtual zones, has been presented, many different shapes for virtual zones have been tested. Notably, ellipses appears as a very judicious choice. The efficiency of the OAM should be improved by using sensors as a laser range scanner since it can provide more numerous and accurate range data. The proposed formalism should then be implemented without resorting to restrictive practical simplifications.

REFERENCES

Barraquand, J., Langlois, B., and Latombe, J. (1992). Numerical potential field techniques for robot path planning. *IEEE Transactions on System, Man and Cybernetics*, 22(2):224–241.

Blanc, G., Mezouar, Y., and Martinet, P. (2005). Indoor navigation of a wheeled mobile robot along visual routes. In *IEEE International Conference on Robotics and Automation, ICRA'05*, Barcelona, Spain.

Braitenberg, V. (1984). *Vehicles: Experiments in Synthetic Psychology*. The MIT Press, Cambridge, Massachusetts.

Filliat, D., Kodjabachian, J., and Meyer, J.-A. (1999). Evolution of neural controllers for locomotion and obstacle avoidance in a 6-legged robot. *Connection Science*, 11:223–240.

Khatib, O. (1986). Real time obstacle avoidance for manipulators and mobile robots. *Int. Journal of Robotics Research*, 5(1):90–98.

Laumond, J. (1998). *Robot motion planning and control*, volume 229, chapter Guidelines in nonholonomic motion planning, pages 1–54. Springer.

Samson, C. (1995). Control of chained systems. application to path following and time-varying stabilization of mobile robots. *IEEE Transactions on Automatic Control*, 40(1):64–77.

Samson, C., Espiau, B., and Borgne, M. L. (1991). *Robot Control : The Task Function Approach*. Oxford University Press.

Zapata, R., Lepinay, P., and Thompson, P. (1994). Reactive behaviors of fast mobile robots. *Journal of Robotic Systems*, pages 13–20.