Control Assistance of a Mobile Robot Navigating into an Encumbered Environment

Guillaume Blanc, Jérôme Roblet, Youcef Mezouar, Philippe Martinet

Control Assistance of a Mobile Robot Navigating into an Encumbered Environment

Guillaume Blanc, Jérôme Roblet, Youcef Mezouar and Philippe Martinet

LASMEA
Blaise Pascal University UMR 6602 CNRS
63177 AUBIERE Cedex
gblanc@lasmea.univ-bpclermont.fr

Abstract

This paper describes and illustrates the theoretical framework of an original obstacle avoidance module which aims to assist the robot guidance while navigating into a cluttered environment. This module, called CAM (*Control Assistance Module*) corrects the control inputs from a navigation module according to local observations, extracted from range data given by embedded sensors. It aims to warrant the robot security but keeps on taking into account the control orders from the navigation module, in order to satisfy the navigation task for the best. The obstacle avoidance approach is inspired by a reflex avoidance method, which consists on preserving a virtual zone around the robot. In an original way, when an obstacle enters into this virtual zone, the CAM plans dynamicly a judicious path to be followed to keep the zone intact. Avoidance control orders are then computed and merged with navigation orders by the CAM, according to a simple rule which privileges the robot's security. The CAM has been implemented and validated on a cart-like mobile robot, equipped with a sonar sensors belt. It also has been put on the test when the robot was teleoperated into an unknown environment for exploration.

1 Obstacles to navigation

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.

1.1 On the environment representation

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 simply describe obstacles by their configuration into the map.

In this context, the first classical approach to obstacle avoidance is to deal with it as a path planning problem, taking into account geometric constraints [6]. A SLAM (*Simultaneous Localization And Mapping*) based environment mapping may be usefull to deal with dynamic obstacles and to accordingly plan an adequate path for a given task. 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 by assigning a repulsive potential field to obstacles [5, 1]. This potential field maps the configuration space. When the robot comes closer to an obstacle, it is under a force which constraints it to move off.
In the case of sensor based approaches of navigation, which do not necessarily refer to an absolute geometric representation, this force can be computed as the image of the estimated distance from the robot to the obstacle by a function which models the potential field. No geometric model of the obstacle is *a priori* required. The robot avoids it according to a sensor based reflex behavior.

1.2 Sensor based approaches

Generally, a sensor based approach consits in mapping the robot control inputs with sensor observations through a behavior model. Works inspired by neurosciences used to base this model on an artificial neural network [3, 4]. More classical approaches of robotics would rather use sensor based control formalism, proposed in [8].
Although this formalism has been highly used in mobile robotics, the nonholonomy of most of them complicates the use of it. The induced kinematic constraints may not allow the robot to execute instantaneously every motion, which may be lead an avoidance task to end in failure. This issue is elegantly addressed by Zapata *et al* with the DVZ approach (*Deformable Virtual Zone*) [9]. This 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. When an obstacle tries to come into the zone, it deforms its shape, as it was made of a supple membrane. The controls are computed in order to minimize this uncontrolled deformation. The approach proposed in the sequel of this article is inspired by this formalism.

1.3 Navigation framework

This paper focuses on an obstacle avoidance module for a nonholonomic mobile robot which navigates autonomously by referring to a topological representation of its environment where obstacles are not modeled explicitly [2]. This representation called visual memory, is formalized as a graph. Each node of the visual memory is a key image of the environment, taken by an embedded camera, and can be used to control the robot motion by visual servoing. Each edge between two key images notifies that the robot has already been able to navigate into a free space from the pose where the camera took one of these images to the pose where it took the other one. Since the navigation environment is not hardly static in time, each edge existence is perpetually questionned. While navigating from a key image to another, the robot may meet on its way an obstacle which was not here when the robot built its visual memory. Of course, this obstacle may be static or dynamic and may take numerous shapes. In an indoor environment, a house for instance, it may be a ball which was not tidied up. The height and the width

of this obstacle are supposed to be roughly the same or inferior compared to the robot dimensions. It does not occult enough the camera image to wreck the vision based control. More precisely, we assume that this kind of obstacle is small enough to be skirted by the robot without loosing the visual reference the navigation module needs to control the robot. The obstacle avoidance module thus aims to assist the navigation module. It must correct the control inputs computed by the navigation module to prevent it driving the robot to the obstacle.
In a more general point of view, this obstacle avoidance module, called CAM (*Control Assistance Module*), acts as a security filter of the control inputs, when they come from a navigation module which guides the robot according to its own strategy. The navigation module supervises the robot motions. The CAM is a parapet. It acts according to the local context of obstacles it can observe. Consequently, the CAM is classified as a reflex obstacle avoidance. Of course the navigation module may take into account the CAM actions to adapt its strategy.

1.4 Contribution

The approach used to design the CAM is based on the DVZ concept. A virtual observation zone surrounds the robot. The shape of this observation zone varies with the instantaneous kinematic state of the robot. The main principle of the approach is also to protect this zone from intrusions of obstacles, but the CAM uses an original strategy to throw them out.
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 judicious path that should be followed for avoidance. The CAM provides a control vector which should be applied to follow the planned path, and merges it adequately with the one from the navigation module.
As a consequence, in an original way, the avoidance method consists in a reflex action which is performed according to a local path following based control.

The specific formalism associated to the proposed approach is the subject of the following Section 2. Then in Section 3, this formalism is used to design the robot motion control, how the avoidance control vector is computed and then merged with the one coming from the navigation module. Section 4 presents an implementation of the CAM on an indoor mobile robot, equipped with a sonar range sensors. The functioning of the CAM is illustrated with a very simple and meaningfull real experiment, then we discuss the use of the CAM in more complex situations.

2 Strategy and formalism

The CAM inherits its main principle from DVZ based approaches. It consists in acting on the robot's controls to thrust obstacles aside a zone which virtually surrounds the robot. The CAM own main idea is to consider the shape of entering obstacles as a path the robot has to follow and maintain to a given distance, which depends on the shape of the virtual zone.

2.1 Virtual zone and obstacles

The robot is modeled by an euclidian frame $\mathsf{F}_R \triangleq (O_R, \mathbf{X}_R, \mathbf{Y}_R, \mathbf{Z}_R)$, with the origin O_R fixed on the control point. We assume that the robot control vector $\mathbf{U}$ only allows it to move on a plan Π, orthogonal to the vector $\mathbf{Z}_R$ and containing O_R. The polar coordinates of every point on Π are ρ for the module and θ for the argument.
Let f_z be a continuous and derivable function:

$$f_z \ : \ \left\{ \begin{array}{rcl} [-\pi, \pi] & \mapsto & I\!\!R \\ \theta & \longrightarrow & \rho = f_z\left(\theta\right) \end{array} \right. \tag{1}$$

such that

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

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

$$\varrho_z = inf\left\{\rho > 0 \mid \rho = f_z(\theta)\right\} \tag{2}$$

We assume now that an obstacle can be known in F_R as a polar function f_o in Π which describes its shape. Let ϱ_o be the corresponding curve:

$$\varrho_o = inf\left\{\rho > 0 \mid \rho = f_o(\theta)\right\} \tag{3}$$

ϱ_o defines the set $\mathcal{O}$, containing every point of the obstacle. Assuming that the robot did not collide with the obstacle, $O_R \notin \mathcal{O}$.

2.2 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 ϱ_f:

$$\varrho_f = min(\varrho_z, \varrho_o) \tag{4}$$

ϱ_f is a closed curve, always defined since ϱ_z is also a closed curve. $\mathcal{F}$ is the inner set defined by ϱ_f.
ϱ_f can be considered as the result of the deformation of ϱ_z by the uncontrolled intrusion of an obstacle.

2.3 Controlled deformation

In section 2.1, ϱ_z is only a geometric curve which does not depends on time since f_z is fixed. However, it may be judicious to adapt the shape of ϱ_z with the navigation context. Remembering that the robot is guided by a navigation module, ϱ_z may be deformed in a controlled mode by taking into account the input control vector $\mathbf{U_c}$ provided by this module. Then the definition of f_z by equation (1) has to be

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

modified with $\rho = f_z\left(\theta, \mathbf{U}_c\right)$.
In the special case of a nonholonomic mobile robot, we will consider in the sequel that this control vector is $\mathbf{U_c} = \left[u_{1_c}, u_{2_c}\right]^\top$, where u_{1_c} is a longitudinal velocity along $\mathbf{X}_R$ and u_{2_c} a rotation velocity around $\mathbf{Z}_R$. An appropriated deformation should be to control a scale factor on ϱ_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 where, of course, it has to be reactive to obstacles, but where its motions also should not be too much constrainted, so that it can carry on making progress.

2.4 Security zone

Up till now, the robot has been reduced to a simple model, made of a single point O_R which motions have to be controlled to avoid collisions. However, we can not ignore that the robot is a volumic object. That is why we define an other virtual zone, called *security zone*, with a closed curve ϱ_s, according to a similar definition as ϱ_z by equation (2). The associated function f_s must be designed so that ϱ_s closely fits the area defined by the orthogonal projection of the robot volume on Π.
ϱ_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 ϱ_s, but if unfortunately it happens, the obstacle must be detected to adapt the control strategy. This also implies that ϱ_s should be within ϱ_z. Controlled deformations of ϱ_z are thus limited:

$$\forall \theta \in \left[-\pi, \pi\right], \ f_z\left(\theta, \mathbf{U}_c\right) \geq f_s\left(\theta\right) \tag{5}$$

3 Control design

3.1 Control strategy

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

$$M \triangleq min\left\{\varrho_f \ - \ \varrho_z\right\} \tag{6}$$

M is the orthogonal projection of O_R on the piece of curve ϱ_f which is not in ϱ_z. Note that if $\mathcal{F} = \mathcal{A}$, $M = \emptyset$. 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 e_ρ be an error which models the deformation from ϱ_z to ϱ_f by a signed distance:

$$e_\rho = \rho_M - f_z(\theta_M, \mathbf{U}_c) \tag{7}$$

e_ρ is a lateral deviation which may be regulated to zero, so that the obstacle is pushed away from the virtual zone. We also define an angular deviation e_φ to be regulated to zero. It is defined by the orientation of the tangent vector to ϱ_f at M, with respect to the longitudinal axis $\mathbf{X}_R$.
The avoidance control must stabilize both e_ρ and e_φ to zero. This defines a path following problem. We address it using the theoretic framework of chained systems [7].

3.2 Lateral control

In the sequel, we assume that the robot is a cart-like robot, classically modeled like a unicycle which is driven with two kinematic inputs. Let $\mathbf{U}_z = [u_{1_z}, u_{2_z}]^\top$ be the kinematic inputs vector computed by the CAM. The robot is supposed to roll without slipping and O_R is confounded with the axle's middle point.
Let F_M be the Frenet frame defined at M, indexed on ϱ_f by the curvilinear abscissae s. The robot motion are described with respect to F_M by the following system of kinematics equations:

$$\begin{cases} \dot{s} = \dfrac{u_{1_z} \cos e_\varphi}{1 - e_\rho c(s)} \\ \dot{e_\rho} = u_{1_z} \sin e_\varphi \\ \dot{e_\varphi} = u_{2_z} - \dot{s} c(s) \end{cases} \tag{8}$$

where $c(s)$ is the curvature of ϱ_f at M.
The system (8) can be transformed to a chained system of dimension three $[a_1, a_2, a_3]^\top \in I\!R^2 \times \left]-\frac{\pi}{2}, \frac{\pi}{2}\right[$ with a control vector of dimension two $[m_1, m_2]^\top \in I\!R^2$:

$$\begin{cases} \dot{a_1} = m_1 \\ \dot{a_2} = a_3 m_1 \\ \dot{a_3} = m_2 \end{cases} \tag{9}$$

by writing:

$$[a_1, a_2, a_3]^\top = [s, e_\rho, (1 - e_\rho c(s)) \tan e_\varphi]^\top \tag{10}$$

If this state vector is derived 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:

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

a_2 and a_3 are regulated to zero according to a settling distance, fixed by the choice of (K_p, K_d). These two gains control the reactivity of the robot when it avoids an obstacle.
In equation (11), m_1 is considered like a parameter which does not influence the regulation dynamic. m_1 is directly linked to u_{1_z} by:

$$m_1 = \dot{s} = u_{1_z} \cos(e_\varphi) \tag{12}$$

The system 9 leads to uncouple the controls m_1 and m_2. This implies that u_{1_z} and u_{2_z} are also uncoupled in this formalism of path following. u_{2_z} leads the lateral regulation of the robot with respect to the obstacle while u_{1_z} can be used to adapt the longitudinal velocity 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 a coefficient 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:

$$\begin{array}{l} \text{if } M \neq \emptyset \\ \quad \alpha \; : \left\{ \begin{array}{ccll} [-\pi,\pi] & \mapsto & [0,1] & \\ \theta_M & \longrightarrow & \alpha\left(\theta_M\right) = \dfrac{\rho_M - f_s\left(\theta_M\right)}{f_z\left(\theta_M\right) - f_s\left(\theta_M\right)} & \text{if } \rho_M \geq f_s\left(\theta_M\right) \\ \theta_M & \longrightarrow & \alpha\left(\theta_M\right) = 0 & \text{otherwise} \end{array} \right. \\ \text{otherwise} \\ \quad \alpha = 1 \end{array} \tag{13}$$

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

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 and consider that most of dynamic obstacles the robot may meet are persons who do not willingly want to bang into it.

3.4 Global control

As exposed in sections (3.2) and (3.3), the CAM provides a control vector $\mathbf{U}_z$ which aims to correct $\mathbf{U}_c$ from a navigation module. u_{1_z} is directly linked to u_{1_c} but the computation of u_{2_z} does not take u_{2_c} into account. But the CAM must favour the navigation module orders if the local configuration of obstacles allows it. A solution simply consists on confering more importance to $\mathbf{U}_z$ when an obstacle draws near the security zone. Let $\mathbf{U}_g = \left[u_{1_g}, u_{1_g}\right]^\top$ be the result of the correction of $\mathbf{U}_c$ by the CAM. $\mathbf{U}_g$ is computed according to the following system:

$$\left\{ \begin{array}{l} u_{1_g} = u_{1_z} \\ u_{2_g} = \gamma u_{2_c} + (1-\gamma) u_{2_z} \end{array} \right. \tag{15}$$

Simply, γ could be equal to α. But, 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:

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

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

$$\begin{array}{l} \text{if } M \neq \emptyset \\ \quad \gamma \; : \left\{ \begin{array}{ccll} [-\pi,\pi] & \mapsto & [0,1] & \\ \theta_M & \longrightarrow & \gamma\left(\theta_M\right) = \dfrac{\rho_M - f_s\left(\theta_M\right) + \varepsilon}{f_z\left(\theta_M\right) - f_s\left(\theta_M\right) + \varepsilon} & \text{if } \rho_M \geq f_s\left(\theta_M\right) + \varepsilon \\ \theta_M & \longrightarrow & \alpha\left(\theta_M\right) = 0 & \text{otherwise} \end{array} \right. \\ \text{otherwise} \\ \quad \gamma = 1 \end{array} \tag{17}$$

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 but not least, e_φ may be equal to $\pm\frac{\pi}{2}$. From a practical point of view, the first two singularities can be shunned. Curves ϱ_z and ϱ_z should result from interpolation of scanning range sensors data. In fact, they are not usually treated as analytic curves, but as a set of sampled points, which estimated position in F_R is affected by noise on measures. Consequently, M is rarely chosen among many points. However, if a choice has to be made, criteria can direct it. We could notably privilege the closest point to the $\mathbf{X}_R$ axis, which directs the longitudinal robot motion. Because curves are sampled every constant step $\Delta\theta$, we simply define the k^{th} derivative f_z^k of f_z by $f_z^k\left(\theta\right) = \frac{f_z^{k-1}(\theta+h)-f_z^{k-1}(\theta-h)}{2h}$, with $h = n\ \Delta\theta$ and n constant and strictly positive integer. Then $c(s)$ is always defined although it is roughly estimated.
The third singularity is more embarrassing. It corresponds to the frequent case where the robot goes orthogonally to an obstacle. The state (10) is not 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 $\pm\frac{\pi}{2}$. We did not found a better solution to this problem than forcing m_2 to zero when e_φ tends to $\pm\frac{\pi}{2}$. In this way, the longitudinal control law prevents the robot from colliding with the obstacle, for want of something better.

4 Implementation

This section illustrates the proposed approach from a practical point of view. The framework proposed in sections (2) and (3) applied to assist the driving of an indoor mobile robot, equipped with a telemetric sensors belt.
The CAM 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 set at $\mathrm{f_e}$. Telemetric data are collected as eight range values, associated respectively to each sensor which angle of sight is known in the robot frame. None filtering is processed on these data and none uncertainty on measures is taken into account.

4.1 Virtual zones

ϱ_z and ϱ_s are two concentric circles. Security zone diameter is fixed to $\phi_s = 0.6\mathrm{m}$. Thus ϱ_s surrounds every sightless zone between to consecutive sensors. ε is fixed to $0.3\mathrm{m}$. The diameter ϕ_z of ϱ_z linearly depends on u_{1_c}. ϕ_z is *a priori* limited by the maximum range of sensors ($3\mathrm{m}$). But we choose this limit $\phi_z max$ to only $1\mathrm{m}$ in order not to constraint the robot motion because of relatively distant obstacles. Let $u_{1_c max}$ be the maximum longitudinal velocity of the robot:

$$\phi_z = (\phi_s + 2\varepsilon)\left(1 - \frac{u_{1_c}}{u_{1_c max}}\right) + \phi_z max\left(\frac{u_{1_c}}{u_{1_c max}}\right) \tag{18}$$

4.2 Planning of a straight line

Although it can be possible to interpolate the eight getted points at each sample in order to obtain a smooth curve which delimits roughly the set of visible obstacles by the sonars, range data are not both

numerous and accurate enough so that it is not worth the trouble. We prefer to refer the avoidance algorithm to a simple straight line. This line determines ϱ_o. ϱ_o is defined by the point M, when it exists, and a vector $\mathbf{\Delta_o}$.
M is chosen by deducing from the measures vector at instant k , $\sigma_k = [\sigma_{1_k}, \sigma_{2_k}, \ldots, \sigma_{8_k}]$, the closest point to O_R within the circle ϱ_z. $\mathbf{\Delta_o}$ depends on the navigation module orders. Let $\Omega_c = \frac{u_{2c}}{\mathrm{f_e}}$ be the instantaneous steering angle which should have directed the robot if it was only guided by the navigation module. We define $\Omega_c \mathbf{z}_R$, with $\mathbf{z}_R$ the unitary vector which directs the axis $\mathbf{Z}_R$. $\Omega_c \mathbf{z}_R$ is the directing vector of the instantaneous robot motion. Let σ_{i_k} be the range data holden to build 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 CAM does not try to 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 deduced from σ_{1_k} or σ_{8_k}, N may not exist. Then $\mathbf{\Delta_o}$ is forced to be $\Omega_c \mathbf{z}_R$.

Following a straight line, with $\forall s, c(s) = 0$, instead of any curve implies a simplification of the control design exposed in section (3.2). Equation 11 conducts to the following simple control law:

$$u_{2_z} = -u_{1_z} \cos {e_\varphi}^3 K_p e_\rho - \mid u_{1_z} \cos {e_\varphi}^3 \mid K_d \tan e_\varphi \tag{19}$$

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

4.3 Results

We have validated the good working of the CAM with two kind of experiments.
The first one consists on a very simple robotic task. The robot's task only 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 $0.3\mathrm{m}$. In these conditions, the CAM 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 lab while the operator can not directly see the robot. He only can see the image provide by an embedded frontal camera, with $60\mathrm{deg}$ of angle of view. Images are wireless transfered, what implies some unexpected and non constent delays on the visual feedback. In this case the CAM ensures the robot security and assist the operator. However, circular virtual zones rapidely appear unadequated. When an obstacle comes closer to robot from a side, the CAM tends to correct the trajectory what creates unexpected oscillations of the robot trajectory. To remedy that, elliptic zones should be a solution to be explored.

5 Conclusion and Perspectives

This paper has presented the design and an implementation of an obstacle avoidance module, called CAM, which aims to act on the robot control inputs in order to preserve the robot's security when it is driven by a navigation module, which does not take explicitely into account the possible presence of

unexpected obstacles.
The CAM is based on an original obstacle avoidance approach. Although it is based on the known DVZ concept, it proposes a new strategy and formalism. They consist on assigning to the robot a reactive behavior by following a dynamically computed path, which results from the local observation of the robot's environment by range sensors. 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 put it away a virtual zone. This zone is a closed curve which surrounds the robot and which 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 CAM, 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 only can get 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, we have tried many different shapes for virtual zones. Notably, ellipses appears as a very judicious choice, which efficiency will be improved by using sensors, like a laser range scanner, which should provide more numerous and accurate range data. The proposed formalism should be implemented without resorting to restrictive practical simplifications.

References

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

[2] G. Blanc, Y. Mezouar, and P. Martinet. Indoor navigation of a wheeled mobile robot along visual routes. In *IEEE International Conference on Robotics and Automation, ICRA'05*, Barcelona, Spain, 18-22 Avril 2005.

[3] V. Braitenberg. *Vehicles: Experiments in Synthetic Psychology*. The MIT Press, Cambridge, Massachusetts, 1984.

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

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

[6] J.P. Laumond. *Robot motion planning and control*, volume 229, chapter Guidelines in nonholonomic motion planning, pages 1–54. Springer, 1998.

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

[8] C. Samson, B. Espiau, and M. Le Borgne. *Robot Control : The Task Function Approach*. Oxford University Press, 1991.

[9] R. Zapata, P. Lepinay, and P. Thompson. Reactive behaviors of fast mobile robots. *Journal of Robotic Systems*, pages 13–20, January 1994.