A method to safely perform a visually guided navigation task amidst occluding obstacles

David Folio† and Viviane Cadenat
LAAS/CNRS, 7, Avenue du Colonel Roche, 31077 Toulouse Cedex 4, France.
dfolio@laas.fr, cadenat@laas.fr

Abstract

This paper presents a sensor-based controller allowing to drive a mobile robot towards a target while avoiding visual features occlusions and obstacle collisions. We consider the model of a cart-like robot equipped with an ultrasonic sensors belt and a camera mounted on a pan-platform. The proposed method relies on the task function formalism and combines visual servoing with path following control whenever an occlusion or a collision may occur. Simulation results are given at the end of the paper.

I. INTRODUCTION

Visual servoing techniques are often divided into two main classes: position-based and image-based [1][2]. In the first kind of methods, the robotic task is described in terms of a camera situation to be reached and the control law will have to make the camera converge from its initial pose to the desired one. On the contrary, in the second one, the task is directly defined in the image and the key idea is to control the displacement of the camera using only visual features. The task function formalism [3] also provides a general framework for designing sensor-based control laws. This formalism was initially developed for manipulators [4]. It has been more recently extended to control mobile robots [5] by adding some degrees of freedom to the robotic system to let the camera move independently from the nonholonomic mobile base. In this way, the camera motion becomes holonomic and it is possible for the system to perform various vision-based tasks using the task function formalism. The visual servoing techniques mentioned above require that the image features remain always in the field of view of the camera, and that they are never occluded during the whole execution of the task. Most of the works which address this kind of problems are dedicated to manipulator arms [6][7][8][9].

Here, we address the problem of avoiding occlusions and collisions for a mobile robot equipped with a camera mounted on a pan-platform when executing a given vision-based task in a cluttered environment. The proposed method is in the sequel of previous works [10][11][12] where different obstacle avoidance techniques were merged to vision-based control laws to perform a visually guided navigation task in a cluttered environment. Although non collision was always guaranteed by these methods, the problem of the target visibility was not addressed. The proposed strategy consists in designing three controllers, the first one performing the desired vision-based task in the free space, the second one guaranteeing occlusion avoidance whenever a risk of occlusion occurs and the last one insuring non collision in the vicinity of the obstacles. Then, we switch from one controller to the other depending on the risk of occlusion and of collision.

The paper is organized as follows: System modelling and problem statement are given in section II. The different controllers and the control strategy are presented in section III. Finally, simulation results are described in section IV.

II. MODELLING AND PROBLEM STATEMENT

We consider the following model of a cart-like robot with a CCD camera mounted on a pan platform (see figure 1).

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\theta}_{pl} \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 & 0 \\ \sin\theta & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} v \\ \omega \\ \varpi \end{pmatrix} \tag{1}$$

(x,y) are the coordinates of the robot reference point M with respect to the world frame $\mathcal{F}_O$. θ and θ_{pl} are respectively the direction of the vehicle, and the direction of the pan-platform, with respect to the x-axis. P is the pan-platform center of rotation and D_x the distance between M and P. We consider the successive frames: $\mathcal{F}_M$ (M, x_M, y_M, z_M) linked to the robot, $\mathcal{F}_P$ (P, x_P, y_P, z_P) attached to the pan-platform, and $\mathcal{F}_C$ (C, x_C, y_C, z_C) linked to the camera. The transformation between $\mathcal{F}_P$ and $\mathcal{F}_C$ is deduced from a hand-eye calibration method. It consists of an horizontal translation of vector $(a,b,0)^T$ and a rotation of angle $\frac{\pi}{2}$ about the y_P-axis. The control input is defined by the vector $\dot{q} = (v, \omega, \varpi)^T$, where v and ω are the linear and the angular velocities of the cart, and ϖ is the pan-platform angular velocity with respect to $\mathcal{F}_M$. Let $T^c = (V^c_{\mathcal{F}_C/\mathcal{F}_O}, \Omega^c_{\mathcal{F}_C/\mathcal{F}_O})^T$ be the kinematic screw

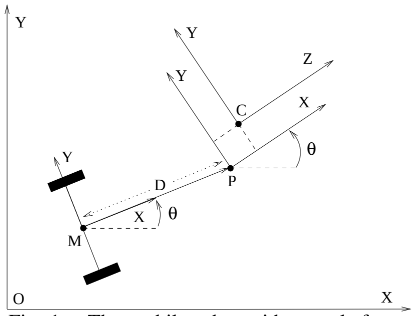

Fig. 1. The mobile robot with pan-platform

† David Folio's work is supported by the European Social Fund.

representing the translational and rotational velocity of $\mathcal{F}_C$ with respect to $\mathcal{F}_O$, expressed in $\mathcal{F}_C$. The kinematic screw is related to the joint velocity vector by the robot jacobian J: $T^c = J\,\dot{q}$. As the camera is constrained to move horizontally, three rows of zeros appear in the jacobian matrix. It is then sufficient to consider a reduced kinematic screw T^c_{red}, and a reduced jacobian matrix J_{red}:

$$T^c_{\text{red}} = \begin{pmatrix} V_{y_c} \\ V_{z_c} \\ \Omega_{x_c} \end{pmatrix} = \begin{pmatrix} -\sin(\theta_{pl}-\theta) & D_x\cos(\theta_{pl}-\theta)+a & a \\ \cos(\theta_{pl}-\theta) & D_x\sin(\theta_{pl}-\theta)-b & -b \\ 0 & -1 & -1 \end{pmatrix} \begin{pmatrix} v \\ \omega \\ \varpi \end{pmatrix} \tag{2}$$

In addition to the CCD camera, the robot is equipped with an ultrasonic (US) sensors belt which allows to characterize locally the closest obstacle. The US data, together with the visual signals, will be considered at the same level for designing the sensor-based controller when dealing with both obstacle and occlusion avoidance.

The problem: We consider the problem of determining a sensor-based closed-loop controller for driving the robot until the camera is positioned in front of a target while avoiding occlusions and obstacles when necessary.

III. Control design

A. The visual servoing control

In this part, we present the nominal vision-based controller in the case that occlusions do not occur. We consider the visual servoing technique introduced in [4]. This approach relies on the task function formalism, which consists in expressing the desired task as a task function e to be regulated to zero [3]. A sufficient condition which guarantees the control problem to be well conditioned is that e is $\rho-$admissible. Indeed, this property ensures the existence of diffeomorphism between the task space and the state space, so that the ideal trajectory $q_r(t)$ corresponding to $e = 0$ is unique. This condition is fulfilled if $\frac{\partial e}{\partial q}$ is regular around q_r [3].

In our application, the target is made of 4 points, defining an 8 dimensional vector of visual signals s in the camera plane. At each configuration of the robot, the variation of the signals $\dot{s}$ is related to the kinematic screw T^c_{red} by means of the interaction matrix L_{red} [4]:

$$\dot{s} = L_{\text{red}}\, T^c_{\text{red}} \tag{3}$$

For a point p of coordinates $(x,y,z)^T$ in $\mathcal{F}_C$ projected into a point $P(X,Y)$ in the image plane (see figure 2), L_{red} is directly deduced from the optic flow equations [4] and given by the following matrix L_{red}:

$$L_{\text{red}} = \begin{bmatrix} 0 & \frac{X}{z} & XY \\ -\frac{1}{Z} & \frac{Y}{Z} & 1+Y^2 \end{bmatrix} \tag{4}$$

Following the task function formalism, the positioning task is defined as the regulation of the following task function $e_{\text{vs}}(q(t))$ to zero:

$$e_{\text{vs}}(q(t)) = C(s(q(t)) - s^*) \tag{5}$$

where s^* is the desired value of the visual signal and $q = [l, \theta, \theta_{pl}]^T$, l representing the curvilinear abscissa of the robot. As the target is fixed, s depends only on $q(t)$ and s^* takes a constant value. C is a full-rank 3×8 combination matrix which allows to take into account more visual features than available degrees of freedom. A simple way to choose C is to consider the pseudo-inverse of the interaction matrix, that is $C = (L^T_{\text{red}} L_{\text{red}})^{-1} L^T_{\text{red}}$ as in [4]. In this way, the task jacobian $\frac{\partial e_{\text{vs}}}{\partial q} = C L_{\text{red}} J_{\text{red}} = J_{\text{red}}$ is always invertible, insuring the ρ-admissibility property. The control law design relies on this property. Indeed, classically, a kinematic controller can be determined by imposing an exponential convergence of e_{vs} to zero as shown below:

$$\dot{e}_{\text{vs}} = J_{\text{red}}\dot{q} = -\lambda_{\text{vs}} e_{\text{vs}} \Longleftrightarrow \dot{q} = \dot{q}_{\text{vs}} = -\lambda_{\text{vs}} J^{-1}_{\text{red}} e_{\text{vs}} \tag{6}$$

where λ_{vs} is a positive scalar or a positive definite matrix.

B. The occlusion avoidance control

Now, we suppose that an occluding object O is present in the camera line of sight. Its projection appears in the image plane as shown on figure 2 and we denote by Y^-_{obs} and Y^+_{obs} the ordinates of its left and right borders. X_{im} and Y_{im} correspond to the axes of the frame attached to the image plane. The proposed strategy only relies on the detection of the two borders of O. As the camera is constrained to move in the horizontal plane, there is no loss of generality in stating the reasoning on Y^-_{obs} and Y^+_{obs}.

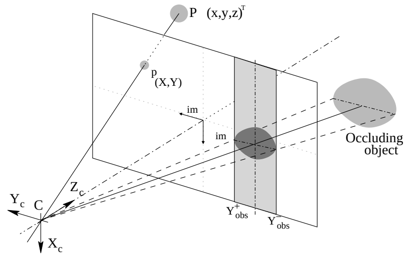

Fig. 2. Projection of both target and occluding object in the image plane

Our goal is to define a task function allowing to preserve the visibility of the visual features in the image. To this aim, we have chosen to use the redundant task function formalism [3]. Let e_1 be a redundant task, that is a low-dimensioned task which does not constraint all degrees of freedom of the robot. Therefore, e_1 is not ρ-admissible and an infinity of ideal trajectories q_r corresponds to the regulation of e_1 to zero. The basic idea of the formalism is to benefit from this redundancy to perform an additional objective. This latter can be modelled as a cost function h to be minimized under the contraint that e_1 is perfectly performed. In that case, the resolution of this optimization leads one to define e as follows :

$$e = W^+ e_1 + \beta(I - W^+ W)g$$

where $W^+ = W^T(WW^T)^{-1}$ is the pseudo-inverse of W, $g = \frac{\partial h}{\partial q}$ and β is a positive scalar (see [3] for details). Under some assumptions (which are verified if $W = \frac{\partial e_1}{\partial q}$), the task jacobian $\frac{\partial e}{\partial q}$ is positive-definite around q_r, insuring that e is ρ-admissible [3].

Our objective is to apply these theoretical results to avoid occlusions while keeping the target in the image. We have chosen to define the occlusion avoidance as the prioritary task. The target tracking will then be considered as the secondary objective and will be modelled as a criterion h_s to be minimized. We propose the following task function e_{oa} (oa is the acronym of "occlusion avoidance"):

$$e_{\text{oa}}(q(t)) = W_{\text{occ}}^+ e_{\text{occ}} + \beta_{\text{oa}}(I - W_{\text{occ}}^+ W_{\text{occ}}) g_s \tag{7}$$

e_{occ} is the redundant task function allowing to avoid the occlusions, $W_{\text{occ}} = \frac{\partial e_{\text{occ}}}{\partial q}$, β_{oa} is a positive scalar and $g_s = \frac{\partial h_s}{\partial q}$ as explained above. We propose the following criterion to track the target and keep it in the camera line of sight:

$$h_s = \frac{1}{2}(s - s^*)^T(s - s^*) \Longrightarrow g_s = ((s - s^*)^T L_{\text{red}} J_{\text{red}})^T \tag{8}$$

Now, considering figure 3, we denote by (X_{s_j}, Y_{s_j}) the coordinates of each point P_j of the target in the image frame, $Y_{\min}$ and $Y_{\max}$ representing the ordinates of the two image sides. We introduce the following distances:

- d_{occ} characterizes the distance before occlusion, that is the shortest distance between the visual features s and the occluding object O. It can be defined as:

$$d_{\text{occ}} = \min\left(\min_j \left|Y_j - Y_{\text{obs}}^+\right|, \min_j \left|Y_j - Y_{\text{obs}}^-\right|\right) = |Y_{\text{s}} - Y_{\text{occ}}| \tag{9}$$

Y_{s} is the ordinate of the closest point P_j to object O, while Y_{occ} represents the closest border of O to the visual features (in the case of figure 3, $Y_{\text{occ}} = Y_{\text{obs}}^+$).

- d_{bord} corresponds to the distance separating the occluding object O and the opposite image side to the visual features.

$$d_{\text{bord}} = \min\left(\left|Y_{\text{obs}}^+ - Y_{\max}\right|, \left|Y_{\text{obs}}^- - Y_{\min}\right|\right) = |Y_{\text{occ}} - Y_{\text{bord}}| \tag{10}$$

where Y_{bord} corresponds to the image border towards which the occluding object must move to leave the image without occluding the target (see figure 3).

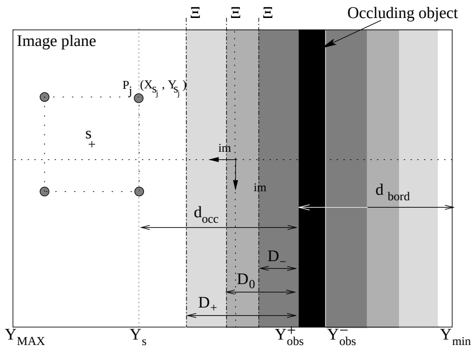

Fig. 3. Definition of the relevant distances for occlusion avoidance

- D_+ defines an envelope Ξ_+ delimiting the region inside which the risk of occlusion is detected.

- D_0 and D_- define two additional envelopes Ξ_0 and Ξ_-. They respectively surround the critical zone inside which it is necessary to start avoiding occlusion and the region where the danger of occlusion is the highest. They will be used in the sequel to determine the global controller.

From these definitions, we propose the following redundant task function e_{occ}:

$$e_{\text{occ}} = \begin{pmatrix} \tan\left(\frac{\pi}{2} - \frac{\pi}{2} \cdot \frac{d_{\text{occ}}}{D_+}\right) \\ d_{\text{bord}} \end{pmatrix} \tag{11}$$

The first component allows to avoid target occlusions: indeed, it increases when the occluding object is getting closer to the visual features and becomes infinite when d_{occ} tends to zero. On the contrary, it decreases when the occluding object is moving far from the visual features and vanishes when d_{occ} equals D_+. Note that, $\forall d_{\text{occ}} \geq D_+$, e_{occ} is maintained to zero. The second component makes the occluding object go out of the image, which is realized when d_{bord} vanishes. Let us remark that these two tasks must be compatible (that is, they can be realized simultaneously) in order to guarantee the control problem to be well stated. This condition is fulfilled by construction thanks to the choice of d_{occ} and d_{bord} (see figure 3). Now, let us determine $W_{\text{occ}} = \frac{\partial e_{\text{occ}}}{\partial q}$. We get :

$$W_{\text{occ}} = \begin{pmatrix} -\frac{1}{D_+}\frac{\pi}{2}\varepsilon_{\text{occ}}\left(1 + \tan^2\left(\frac{\pi}{2} - \frac{\pi}{2} \cdot \frac{d_{\text{occ}}}{D_+}\right)\right)\left(\frac{\partial Y_{\text{s}}}{\partial q} - \frac{\partial Y_{\text{occ}}}{\partial q}\right) \\ \varepsilon_{\text{bord}}\frac{\partial Y_{\text{occ}}}{\partial q} \end{pmatrix} \text{ with } \left\{ \begin{array}{ccc} \frac{\partial Y_{\text{s}}}{\partial q} & = & \begin{pmatrix} -\frac{1}{z_s} & \frac{Y_{\text{s}}}{z_s} & 1 + Y_{\text{s}}^2 \end{pmatrix} J_{\text{red}} \\ \frac{\partial Y_{\text{occ}}}{\partial q} & = & \begin{pmatrix} -\frac{1}{z_{\text{occ}}} & \frac{Y_{\text{occ}}}{z_{\text{occ}}} & 1 + Y_{\text{occ}}^2 \end{pmatrix} J_{\text{red}} \end{array} \right.$$

where $\varepsilon_{\text{occ}} = \text{sign}(Y_{\text{s}} - Y_{\text{occ}})$, $\varepsilon_{\text{bord}} = \text{sign}(Y_{\text{occ}} - Y_{\text{bord}})$, while z_s and z_{occ} are the depth of the target and of the occluding object.

At this step, the task function e_{oa} guaranteeing the occlusion avoidance while keeping the target in the image is completely determined (see relation (7)). Now, it remains to design a controller allowing to regulate it to zero. As W_{oa} and β_{oa} are chosen to fulfill the assumptions of the redundant task formalism [3], the task jacobian $J_{\text{oa}} = \frac{\partial e_{\text{oa}}}{\partial q}$ is positive definite around the ideal trajectory and e_{oa} is ρ-admissible. As well, this result allows to simplify the control synthesis. Indeed, it can be shown that a controller making e_{oa} vanish is given by [4]:

$$\dot{q} = \dot{q}_{\text{oa}} = -\lambda_{\text{oa}} e_{\text{oa}} \text{ where } \lambda_{\text{oa}} \text{is a positive scalar or a positive definite matrix.} \tag{12}$$

C. The obstacle avoidance control

The avoidance strategy is based on the ultrasonic data. From these data, we compute a couple (d_{av}, α) for any obstacle located at a distance inferior to d_+ (see figure 4). d_{av} is the signed distance between M and the closest point Q on the obstacle, and α is the angle between the tangent to the obstacle at Q and the robot direction. Note that there exists two angles α corresponding to the two possible directions for the avoidance motion. As the obstacle can also be an occluding object, we propose to maintain the target visibility by defining α so that the robot moves around the obstacle in the direction given by the pan-platform.

Consider figure 4. Around each obstacle, three envelopes are defined. The first one ξ_+ located at a distance $d_+ > 0$ surrounds the zone inside which the obstacle is detected by the robot. For the problem to be well stated, the distance between two obstacles is assumed to be greater than $2d_+$ to prevent the robot from considering several obstacles simultaneously. The second one ξ_0, located at a lower distance $d_0 > 0$, constitutes the virtual path along which the reference point M will move around the obstacle. The last one ξ_- defines the region inside which the risk of collision is maximal (this envelope will be used in the sequel to define the global controller). Using the path following formalism introduced in [13], we define a mobile frame on ξ_0 whose origin Q' is the orthogonal projection of M. During obstacle avoidance, the robot linear velocity is supposed to be kept constant. Let $\delta = d_{\text{av}} - d_0$ be the signed distance between M and Q'. With respect to the moving frame, the dynamics of the error terms (δ, α) is described as follows:

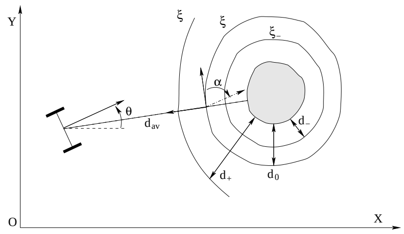

Fig. 4. Obstacle avoidance

$$\begin{cases} \dot{\delta} = v \sin\alpha \\ \dot{\alpha} = \omega - v\chi cos\alpha \end{cases} \quad \text{with } \chi = \frac{\frac{\sigma}{R}}{1 + \frac{\sigma}{R}\delta} \tag{13}$$

where R is the curvature radius of the obstacle, $\sigma = \{-1, 0, +1\}$ depending on the sense of the robot motion around the obstacle. The path following problem is classically defined as the search for a controller ω allowing to steer the pair (δ, α) to $(0,0)$ under the assumption that v never vanishes to preserve the system controllability. Here, our goal is to solve this problem using the task function formalism. To this aim, we have to find a task function whose regulation to zero will make δ and α vanish while insuring $v \neq 0$. We propose the following redundant task function e_{av}:

$$e_{\text{av}} = \begin{pmatrix} l - v_r t \\ \delta + k\alpha \end{pmatrix} \tag{14}$$

where l is the curvilinear abscissa of point M and k a positive gain to be fixed. The first component of this task function allows to regulate the linear velocity of the mobile base to a nonzero constant value[1] v_r. In this way, the linear velocity never vanishes, guaranteeing that the control problem is well stated and that the robot will not remain stuck on the security envelope ξ_0 during the obstacle avoidance. The second component of e_{av} can be seen as a sliding variable whose regulation to zero makes both δ and α vanish[2] (see [14] for a detailed proof). Therefore, the regulation to zero of e_{av} guarantees that the robot follows the security envelope ξ_0, insuring non collision. As the chosen task function does not constraint the whole degrees of freedom of the robot, we use the redundant task function formalism to perform a secondary task (here, avoiding target loss and occlusions at best) while the obstacle avoidance is realized. We propose to model this secondary objective using the cost function $h_{\text{occ}} = \frac{1}{d_{\text{occ}}}$. This criterion can be seen as a potential function allowing to avoid both occlusions and target loss as d_{occ} is defined with respect to the image borders when no occluding object lies in the image plane. The global task function e_{ob} expresses as follows:

$$e_{\text{ob}} = W_{\text{av}}^{+} e_{\text{av}} + \beta_{\text{ob}}(I - W_{\text{av}}^{+} W_{\text{av}}) g_{\text{occ}} \tag{15}$$

where β_{ob} is a positive scalar. Finally, a straightforward calculus shows that :

$$g_{\text{occ}} = \frac{\partial h_{\text{occ}}}{\partial q} = -\frac{1}{d_{\text{occ}}^2}\frac{\partial d_{\text{occ}}}{\partial q} = -\frac{\varepsilon_{\text{occ}}}{d_{\text{occ}}^2}\left(\frac{\partial Y_{\text{s}}}{\partial q} - \frac{\partial Y_{\text{occ}}}{\partial q}\right)^T \text{ and } W_{\text{av}} = \frac{\partial e_{\text{av}}}{\partial q} = \begin{pmatrix} 1 & 0 & 0 \\ \sin\alpha - k\chi\cos\alpha & k & 0 \end{pmatrix}$$

[1]- v_r must be chosen small enough to let the robot sufficiently slow down to avoid collision when entering the critical zone.

[2]- The value of k determines the relative convergence velocity of δ and α as the sliding variable converges.

Following the redundant task function formalism, a controller making e_{ob} vanish is given by:

$$\dot{q} = \dot{q}_{\text{ob}} = -\lambda_{\text{ob}} e_{\text{ob}} \text{ where } \lambda_{\text{ob}} \text{is a positive scalar or a positive definite matrix.} \tag{16}$$

D. The global controller

There exist globally two approaches for sequencing tasks. In the first one, the switch between two successive tasks is dynamically performed [15][16], while in the second one, it relies on convex combinations between either the successive task functions or the successive controllers [5][12]. The latter technique appears to be simpler, allowing to carry out tasks more easily although it is harder to guarantee its feasability. Here, we have chosen to use the second class of approaches and the global control law is computed by linearly combining the three previously defined controllers.

$$\dot{q} = (1-\mu_{\text{oa}})(1-\mu_{\text{ob}})\dot{q}_{\text{vs}} + (1-\mu_{\text{ob}})\mu_{\text{oa}}\dot{q}_{\text{oa}} + \mu_{\text{ob}}\dot{q}_{\text{ob}} \tag{17}$$

where $\dot{q}_{\text{vs}}$, $\dot{q}_{\text{oa}}$ and $\dot{q}_{\text{ob}}$ are respectively given by equations (6), (12) and (16). μ_{oa} and $\mu_{\text{ob}} \in [0,1]$ allow to switch continuously from one control to the other depending on the risk of occlusion and of collision. Several cases may occur:

- If the occluding object O lies outside the region defined by Ξ_0 or outside the image ($d_{\text{occ}} > D_0$) and if there is no obstacle in the vicinity of the robot ($d_{\text{av}} > d_0$), μ_{oa} and μ_{ob} are fixed to 0 and the sole visual control (6) is used.

- If the visual features enter the zone delimited by Ξ_0, the danger of occlusion becomes higher and μ_{oa} progressively increases to reach 1 when they cross Ξ_-. Therefore, while the visual features remain between Ξ_0 and Ξ_-, the robot is controlled by a linear combination of $\dot{q}_{\text{vs}}$, $\dot{q}_{\text{oa}}$ and possibly $\dot{q}_{\text{av}}$ (see remark 1). If the action of the global controller is sufficient to avoid occlusions, the visual features may leave naturally the critical zone defined by Ξ_0 and μ_{oa} goes back to 0 without having reached 1. On the contrary, if Ξ_- is crossed, a flag OCCLU is set to 1, while μ_{oa} is fixed and maintained to 1 until the object O leaves the image ($d_{\text{bord}} = 0$) or at least goes out the critical zone ($d_{\text{occ}} = D_0$). When one of these two events occurs, μ_{oa} is progressively reduced from 1 to 0 and vanishes once the visual features cross Ξ_+. Following this reasoning, μ_{oa} depends on d_{occ} as shown on equation (18).

Remark 1: The presence of an occluding object in the image does not necessarily mean that a collision may occur. Indeed, an obstacle may be detected by the camera before it becomes dangerous for the mobile base. This is the reason why we consider two different controllers $\dot{q}_{\text{oa}}$ and $\dot{q}_{\text{av}}$ depending on the occlusion occurs far from the obstacle or close to it. Moreover, the different envelops are chosen close enough in order to reduce the duration of the transition phase and insure that the robot will be rapidly controlled by the most relevant controller depending on the environment.

- If the mobile base enters the zone surrounded by ξ_0 ($d_{\text{av}} < d_0$), the danger of collision rises and μ_{ob} is continuously increased from 0 to reach 1 when $d_{\text{av}} < d_-$. If ξ_- is never crossed, the robot naturally leaves the critical zone defined by ξ_0 and μ_{ob} is brought back to 0, once ξ_+ is crossed. If μ_{ob} reaches 1, the danger of collision is maximum and a flag AVOID is enabled. As the robot safety is considered to be the most important objective, the global controller (17) has been designed so that only $\dot{q}_{\text{av}}$ is applied to the vehicle once μ_{ob} has reached its maximal value. In this way, the robot is controlled using the sole controller $\dot{q}_{\text{av}}$, allowing to guarantee non collision while performing the occlusion avoidance at best. The robot is then brought back on the security envelope ξ_0 and follows it until the condition to leave is fulfilled. This event occurs when the camera and the mobile base have the same direction ($\theta = \theta_{pl}$). A flag LEAVE is then positioned to 1 and μ_{ob} is rapidly decreased to vanish on ξ_+. Therefore, μ_{ob} depends on the distance d_{av} as shown on equation (18):

$$\begin{cases} \mu_{\text{oa}} = 0 & \text{if } d_{\text{occ}} > D_0 \text{ and OCCLU} = 0 \\ \mu_{\text{oa}} = \frac{d_{\text{occ}} - D_0}{D_- - D_0} & \text{if } d_{\text{occ}} \in [D_-, D_0] \text{ and OCCLU} = 0 \\ \mu_{\text{oa}} = \frac{d_{\text{occ}} - D_+}{D_{\text{leave}} - D_+} & \text{if } d_{\text{occ}} \in [D_{\text{leave}}, D_+] \\ & \text{and } (d_{\text{bord}} = 0 \text{ or } d_{\text{occ}} \geq D_0) \\ \mu_{\text{oa}} = 1 & \text{otherwise} \end{cases} \quad \begin{cases} \mu_{\text{ob}} = 0 & \text{if } d_{\text{av}} > d_0 \text{ and AVOID} = 0 \\ & \text{and LEAVE} = 0 \\ \mu_{\text{ob}} = \frac{d_{\text{av}} - d_0}{d_- - d_0} & \text{if } d_{\text{av}} \in [d_-, d_0] \text{ and AVOID} = 0 \\ \mu_{\text{ob}} = \frac{d_{\text{av}} - d_+}{d_s - d_+} & \text{if } d_{\text{av}} \in [d_s, d_+] \text{ and LEAVE} = 1 \\ \mu_{\text{ob}} = 1 & \text{otherwise} \end{cases} \tag{18}$$

where D_{leave} is the distance d_{occ} for which the leaving condition for occlusion avoidance has been fulfilled and d_s is defined by the distance d_{av} when LEAVE $= 1$.

IV. Simulation results

Our method has been implemented to simulate a mission whose objective is to position the camera in front of a given target. To validate our approach, the environment has been cluttered with two cylindric obstacles which may occlude the camera or represent a danger for the mobile base. For this test, D_-, D_0 and D_+ have been respectively fixed to 40, 60 and 75 pixels, and d_+, d_0, d_- to 0.7m, 0.55m, and 0.45m. The sampling period is equal to 150ms and is the same as on the real robot. The robot initial configuration and the the obstacles position has been chosen to induce occlusions and collisions.

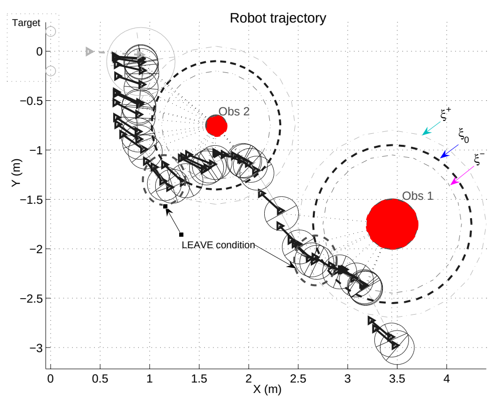

Fig. 5. Robot trajectory

The obtained results are presented on figures 5 and 6. As shown on figure 5, the task is correctly performed: target occlusions and obstacle collisions never occur during the whole mission. At the beginning of the task, there is no risk of occlusion or collision, the robot is only controlled by $\dot{q}_{\mathrm{vs}}$ and starts converging towards the target. When the vehicle enters the vicinity of the first obstacle, μ_{ob} is progressively increased to its maximum value (see figure 6), and the robot first follows the security envelope ξ_0 before being brought back on envelope ξ_+ once the leaving condition has been fulfilled. During this phase, μ_{oa} remains equal to 0 as the two obstacles do not induce any risk of occlusion. When ξ_+ is crossed, μ_{ob} vanishes and the robot executes once again the nominal vision-based task. However, the second obstacle induces a risk of both collision and occlusion. μ_{oa} and μ_{ob} are then continuously increased and the robot starts avoiding both occlusion and obstacle. However, this first motion does not suffice to prevent the vehicle from crossing ξ_-. At this time, the collision danger is the highest and the sole controller $\dot{q}_{\mathrm{av}}$ is used to guarantee a safe motion for the robot. Therefore, the vehicle is controlled to follow the security envelope ξ_0 while occlusions are avoided at best. When the leaving condition is obtained, μ_{ob} is rapidly decreased while μ_{oa} has already vanished because the avoidance motion has made the obstacle leave the image plane. Therefore, once again, the robot starts converging towards the target. However, we have chosen the target and the obstacle positions to insure that this motion brings the vehicle back towards the obstacle. As a consequence, instead of vanishing, μ_{ob} rises again, making the robot avoid the obstacle. Thanks to the avoidance motion, the vehicle leaves progressively the obstacle vicinity and μ_{ob} vanishes. The sole visual servoing controller is then applied to the vehicle and the camera finally reaches its desired position. The task is then successfully realized.

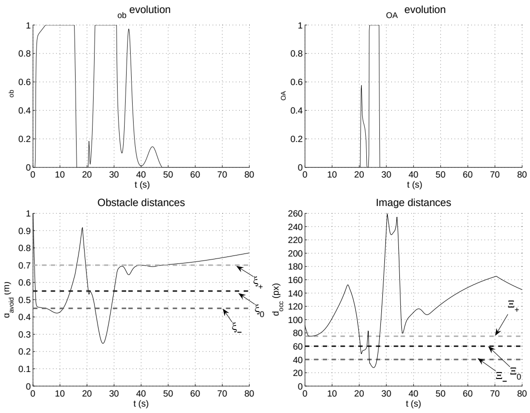

Fig. 6. Evolution of μ_{oa}, μ_{ob}, d_{occ} and d_{av}

V. CONCLUSION

The proposed sensor-based controller allows the mobile robot to perform safely a vision-based task in a cluttered environment while avoiding visual features occlusions. The method relies on the continuous switch between different controllers, depending on the environment. The obtained results are quite satisfying and these control laws will be experimented on our mobile robots. This work has been also shown to be restricted to missions where occlusions can be effectively avoided, which is not the case of all robotic tasks. Therefore, further extensions will have to accept that occlusions may occur rather than in avoiding them.

REFERENCES

[1] P. Corke, *Visual control of robots : High performance visual servoing*. Research Studies Press LTD, 1996.

[2] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, October 1996.

[3] C. Samson, B. Espiau, and M. L. Borgne, *Robot control: the task function approach*. Oxford: Oxford University Press, 1991.

[4] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Transaction on Robotics and Automation*, vol. 5, June 1992.

[5] R. Pissard-Gibollet and P. Rives, "Applying visual servoing techniques to control a mobile hand-eye system," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA' 95)*, Nagoya, Japan, May 1995, pp. 166–171.

[6] Y. Mezouar and F. Chaumette, "Avoiding self-occlusions and preserving visibility by path planning in the image," *Robotics and Autonomous Systems*, vol. 41, no. 2, pp. 77–87, November 2002.

[7] E. Marchand and G. Hager, "Dynamic sensor planning in visual servoing," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'98)*, vol. 3, Leuven, Belgium, May 1998, pp. 1988–1993.

[8] A. I. Comport, E. Marchand, and F. Chaumette, "Robust model-based tracking for robot vision," in *Proc. IEEE/RSJ International Conference on intelligent Robots and Systems (IROS'04)*, Sendai, Japan, Oct. 2004.

[9] P. Wunsch and G. Hirzinger, "Real-time visual tracking of 3d objects with dynamic handling of occlusion," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'97)*, Albuberque, Mexico, April 1997, pp. 2868–2873.

[10] V. Cadenat, P. Souères, and M. Courdesses, "An hybrid control for avoiding obstacles during a vision-based tracking task," in *Proc. European Control Conference (ECC'99)*, Karlsruhe, Germany, Sept. 1999.

[11] V. Cadenat, P. Souères, and M.Courdesses, "Two multi-sensor-based control strategies for driving a robot amidst obstacles," in *Proc. Conference on Decision and Control (CDC'00)*, Sydney, Australia, Dec. 2000.

[12] V. Cadenat, P. Souères, and M. Courdesses, "Using system redundancy to perform a sensor-based navigation task amidst obstacles," *International Journal of Robotics and Automation*, vol. 16, Issue 2, pp. 61–73, 2001.

[13] C. Samson, "Path following and time varying feedback stabilization of a wheeled mobile robot," in *Proc.International Conference on Control, Automation, Robotics and Vision ICARV'92)*, Singapore, Sept. 1993, pp. 13.1.1–13.1.5.

[14] P. Souères, T. Hamel, and V. Cadenat, "A path following controller for wheeled robots which allows to avoid obstacles during the transition phase," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'98)*, Leuven, Belgium, May 1998.

[15] P. Souères and V. Cadenat, "Dynamical sequence of multi-sensor based tasks for mobile robots navigation," in *Proc. 7th Symposium on Robot Control (SYROCO'03)*, Wroclaw, Poland, Sept. 2003, pp. 423–428.

[16] N. Mansard and F. Chaumette, "Tasks sequencing for visual servoing," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'04*, vol. 1, Sendai, Japan, September 2004, pp. 992–997.