



HAL
open science

An anticipative reactive control strategy to deal with unforeseen obstacles during a multi-sensor-based navigation task

Adrien Durand Petiteville, Viviane Cadenat

► **To cite this version:**

Adrien Durand Petiteville, Viviane Cadenat. An anticipative reactive control strategy to deal with unforeseen obstacles during a multi-sensor-based navigation task. European Control Conference (ECC), Jun 2014, Strasbourg, France. hal-01959464

HAL Id: hal-01959464

<https://hal.science/hal-01959464v1>

Submitted on 18 Dec 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

An anticipative reactive control strategy to deal with unforeseen obstacles during a multi-sensor-based navigation task

Adrien Durand Petiteville¹ and Viviane Cadenat²

Abstract— We address the problem of the rangefinder-based avoidance of unforeseen static obstacles during a visual navigation task. We extend previous strategies which are efficient in most cases but remain still hampered by some drawbacks (e.g., risks of collisions or of local minima in some particular cases, etc.). The key idea is to complete the control strategy by adding a controller providing the robot some anticipative skills to guarantee non collision and by defining more general transition conditions to deal with local minima. Simulation results show the proposed strategy efficiency.

I. INTRODUCTION

Autonomous navigation is a well-known problem which has been the source of countless contributions. It roughly consists for the robot in reaching a pre-defined goal through a cluttered environment [1]. In the robotics literature, the navigation strategies are traditionally splitted into two categories: the map-based ones and the mapless (or reactive) ones [1], [2], [3]. The first approach covers methods which require an a priori knowledge of the environment. In other words, a map of the scene must be provided to the robot before the mission begins. It then consists in extracting and following a path connecting the initial position to the goal to perform the task. However, doing so, this kind of methods assume that the environment does not change while the robot is moving [4], which does not allow to take into account the unexpected events. Indeed, while navigating, the robot has to deal with static (e.g. walls), semi-static (e.g. chairs) or dynamic (e.g. humans) obstacles. If the first ones can be easily included in the map, it is not the case of the two last ones which may lead to unforeseen risks of collisions. Thus these techniques appear to introduce some rigidity in the navigation system, even if improvements have been made by developing methods allowing to re-plan a new path [5], [6] or to locally modify the robot trajectory [7], [8]. The second category of approach covers reactive techniques which allow to deal much more efficiently with unforeseen events. In this case, the robot will move through the scene depending on the goal to be reached and on the sensory data perceived during the navigation [3]. Some well-known techniques belong to this category: the extended potential field [9], the global DWA [10] or the VFH* technique [11]. An interesting overview of such approaches can be found in [2], [12].

¹A. Durand Petiteville is with Electrical Engineering and Computer Science School, Queensland University of Technology, Brisbane, Queensland, Australia adrien.durandpetiteville@qut.edu.au

²V. Cadenat is with CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France, Univ de Toulouse, UPS, LAAS, F-31400, Toulouse, France cadenat@laas.fr

Our works focus on the sensor-based navigation in a cluttered environment using the task function approach [13]. While navigating towards the target using a visual servoing controller, the robot may have to deal with unforeseen static and semi-static obstacles. To overcome this issue, we have chosen to design a path following controller allowing to make the robot move on a security envelope. The key idea was to switch between these two controllers designed using the task function approach. Although the proposed strategy has led to very interesting results [14], it still suffered from several drawbacks. First of all, the bypassing motion is started only when the obstacle becomes really dangerous, that is when the robot enters a predefined zone around it. This strategy may increase the risk of collision in some cases (e.g., the robot velocity is too fast, the zone is too small, etc.). All these parameters can be hardly set a priori and their values will depend on the context. The same problem holds for the transition conditions allowing to switch between the vision-based task and the obstacle avoidance. Indeed, they strongly depend on multiple parameters which may greatly differ depending on the environment, the task, the robot, etc. It is then difficult to properly tune them a priori and to obtain a set of parameters suitable for any vision-based task. Thus, the previously proposed strategies suffer from a lack of generality which induces problems of local minima, unexpected collisions, etc. Our goal is to improve and generalize them. First, we have developed a new controller which allows to ease the bypassing motion and which does not rely on any particular parameter. Second, we have also improved the supervision algorithm by determining novel transition conditions based on the prediction of the effects of the controllers on the robot behavior. Finally, we have also proposed a new method allowing to sequence the controllers more easily. The obtained control strategy is then more generic.

The article is organized as follows. We first describe the robotic system and our previous navigation strategy. We then detail our contributions. Finally, we present simulation results validating our approach.

II. PRELIMINARIES

A. System modeling

The considered system is a differential robot equipped with a camera mounted on a pan-platform and a laser rangefinder able to detect the obstacles. Fig 1(a) shows the robot model, which requires to define the successive frames: F_O attached to the world, F_M linked to the robot, F_P attached to the platform, and F_C linked to the camera. Let θ

be the direction of the robot wrt. \vec{x}_O , ϑ the direction of the pan-platform wrt. \vec{x}_M , P the pan-platform centre of rotation, C_x and C_y the coordinates of C in F_P , and D_x the distance between M and P . Defining vector $q = (l, \theta, \vartheta)^T$ where l is the robot curvilinear abscissa, the control input is given by $\dot{q} = (v, \omega, \varpi)^T$, where v and ω are the cart linear and angular velocities, and ϖ is the pan-platform angular velocity wrt. F_M . Thus, we obtain the following kinematic model:

$$\dot{\chi} = \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \\ \dot{\vartheta}(t) \end{bmatrix} = \begin{bmatrix} v(t) \cos(\theta(t)) \\ v(t) \sin(\theta(t)) \\ \omega(t) \\ \varpi(t) \end{bmatrix} \quad (1)$$

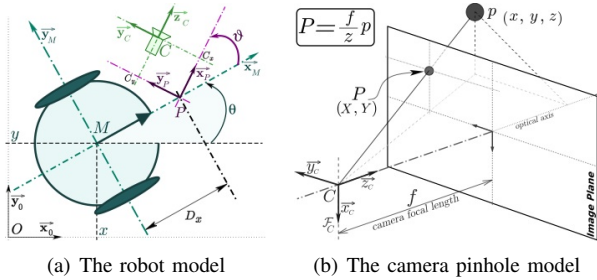


Fig. 1: The robotic system

We now describe the pinhole model used to represent the camera (Fig. 1(b)). A point P , with coordinates (X, Y) in the image plane, is the projection of a point p , with coordinates (x, y, z) in F_O . Moreover we define z as the depth in F_C of the projected point p and f as the camera focal length.

A landmark, which can be characterized by a number k of points P_i (for $i \in [1, \dots, k]$) in the world frame, is represented by a $2k$ -dimensional vector s made of the coordinates (X_i, Y_i) in the image plane. For such a robot and visual features, the relation between \dot{s} and \dot{q} is given by:

$$\dot{s} = L_{(s,z)} J_r \dot{q} \quad (2)$$

where J_r is the robot jacobian [15]

$$J_r = \begin{pmatrix} -\sin(\vartheta(t)) & D_x \cos(\vartheta(t)) + C_x & C_x \\ \cos(\vartheta(t)) & D_x \sin(\vartheta(t)) - C_y & -C_y \\ 0 & -1 & -1 \end{pmatrix} \quad (3)$$

and $L_{(s,z)} = [L_{(P_1)}^T, \dots, L_{(P_k)}^T]^T$ the interaction matrix. $L_{(P_i)}$ is classically given by [16]:

$$L_{(P_i)} = \begin{pmatrix} L_x(s_i, z_i) \\ L_y(s_i, z_i) \end{pmatrix} = \begin{pmatrix} 0 & \frac{X_i}{z_i} & \frac{X_i Y_i}{f} \\ -\frac{f}{z_i} & \frac{Y_i}{z_i} & f + \frac{Y_i^2}{f} \end{pmatrix} \quad (4)$$

B. The control strategy

We now recall the main principles of our earlier control strategy. We have chosen a reactive strategy made of several controllers. They respectively allow to perform the vision-based task and the obstacle avoidance. It will then suffice to switch between them when necessary to perform the task. We first present the two controllers and the sequencing method before highlighting the limitations of the approach.

1) *The vision-based controller*: The vision-based navigation task consists in positioning the camera with respect to the above static landmark. To do so, we have used an image-based visual servoing (IBVS) controller whose goal is to make the current visual signals s converge to their reference values s^* obtained at the desired pose for the camera. To perform this task, we apply the visual servoing technique given in [16] to mobile robots as in [17]. The approach relies on the task function formalism [13] and consists in expressing the visual servoing task by the following task function to be regulated to zero:

$$e_{vs} = s - s^* \quad (5)$$

By imposing an exponential decrease on e_{vs} to make it vanish, we obtain the following a controller [15] :

$$\dot{q}_{vs} = -(L_{(s,z)} J_r)^+ \lambda_{vs} (s - s^*) \quad (6)$$

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

2) *Obstacle avoidance*: We now recall our obstacle avoidance technique [14]. The strategy consists in making the vehicle avoid the obstacle¹ by following the security envelope ξ_0 (Fig. 2). To design the desired controller, the following

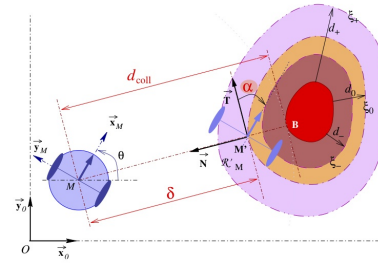


Fig. 2: Collision detection

task function has been defined [14]:

$$e_{mb} = (l - v_r t \quad \delta + \kappa_{oa} \alpha)^T \quad (7)$$

where l is the curvilinear abscissa of point M , while δ and α respectively represent the distance and the relative orientation between the robot and the obstacle (Fig 2). κ_{oa} is a positive gain to be fixed and v_r the desired linear velocity. The first component allows to regulate the linear velocity to v_r . The second component can be seen as a sliding variable whose regulation to zero makes both δ and α vanish. Thanks to task (7), only the mobile base is controlled. To keep the target in the camera field of view during the avoidance phase, we define a specific task function for the pan-platform:

$$e_{pp} = Y_0 \quad (8)$$

where Y_0 is the abscissa of the visual pattern gravity center. We now obtain $e_{oa} = (e_{mb}^T e_{pp})^T$ for the obstacle avoidance. Following the same reasoning as for visual servoing, a controller making e_{oa} vanish is given by:

$$\dot{q}_{oa} = J_{oa}^{-1} (-\lambda_{oa} e_{oa} - A) \quad (9)$$

¹We will suppose in the sequel that the obstacles are non-occluding. However, note that we have already proposed efficient solutions to this problem in previous works [15], [18].

where λ_{oa} is a positive scalar, $A = (-v_r \ 0 \ 0)^T$ and:

$$J_{oa} = \begin{pmatrix} 1 & 0 & 0 \\ \sin(\alpha) - \kappa_{oa}\chi_{oa} \cos(\alpha) & \kappa_{oa} & 0 \\ L_{Y_0} J_r & & \end{pmatrix} \quad (10)$$

$\chi_{oa} = \frac{1}{1 + \frac{\sigma}{R}\delta}$, with R the curvature radius of the obstacle and $\sigma = \{-1, 0, 1\}$ depending on the sense of the robot motion around the obstacle. $L_{Y_0} = (-\frac{f}{z_0}, \frac{Y_0}{z_0}, f + \frac{Y_0^2}{f})$, and z_0 is the depth of the visual pattern gravity center. It should be noticed that the obstacle avoidance task is admissible as the task function jacobian J_{oa} is invertible (see [14]).

3) *Sequencing the controller*: The strategy requires to switch between \dot{q}_{vs} and \dot{q}_{oa} whenever necessary. It is mandatory to preserve the global control law continuity sent to the robot at the switching instant. In previous works, we have used two main techniques to fulfill this objective [14]. The first one relies on convex combination of \dot{q}_{vs} and \dot{q}_{oa} . Although it is simple, it is sensitive to local minima because two antagonist controllers can act on the robot at the same time. The second one, known as dynamical sequencing [19], has been developed to solve this problem. It insures the values of the successive controllers are identical at the switching time, guaranteeing the control law continuity [14].

C. Parameters estimation for visual servoing

As indicated in the introduction, the novel transition conditions rely on the prediction of the effects of the controllers on the robot behavior. To do so, concerning the IBVS, it is necessary to predict the visual features (X_i, Y_i) and their depth z_i . We have already developed such tools to answer the problem of the visual features loss during the navigation task [15], [18]. We plan to use them to fulfill another purpose. We briefly summarize the estimation algorithms below.

1) *The current visual features reconstruction*: In [15], we have developed a method allowing to reconstruct the visual features when the image is totally unavailable. The idea was to integrate the dynamical system (2) during a time control interval $[t_{k-1}, t_k]$, with $t_k = t_{k-1} + T_s$ and T_s the sampling period. We have then obtained an analytical solution which provides the values of $X_i(k), Y_i(k)$ and $z_i(k)$ using $X_i(k-1), Y_i(k-1), z_i(k-1)$ and $\dot{q}(k-1)$. This method requires the initial values of $X_i(t), Y_i(t)$ and $z_i(t)$. The two first ones can be easily obtained from the image processing, while it is not the case of the last one which must be estimated.

2) *The visual feature depth estimation*: To provide the necessary depths for II-C.1, we have developed in [18] an estimation process based on a predictor/corrector pair using a number m of previous images. This latter characteristic allows to improve the signal/noise ratio, making the method robust to image processing and odometry errors, thereby increasing the depth estimation accuracy.

D. Evaluation of the control strategy

Our control strategy is purely reactive. It relies on switching between two controllers (visual servoing and obstacle avoidance). This kind of approach presents the advantage

to deal easily with unforeseen events, but they suffer from the existence of local minima. When designing this strategy, we have paid a particular attention to this problem by: (i) guaranteeing the robot is always given a nonzero linear velocity during obstacle avoidance; and (ii) avoiding that two opposite controllers act at the same time on the vehicle. Thanks to our strategy, in most cases, the robot was able to reach the goal while avoiding obstacles. However, a deep analysis of the results has shown that some problems still may arise [14].

First, the avoidance motion is started only when the obstacle becomes really dangerous, that is when the robot enters the zone surrounded by ξ_0 . Thus, if d_0 is chosen too small or if the vehicle cannot turn sufficiently fast, a collision might still occur. Furthermore, it is quite difficult to tune parameters a priori. As a consequence, this region is often overestimated, preventing the vehicle from moving through narrow passages between obstacles. The second problem is related to the choice of the transition conditions. These conditions cannot be easily set because their values depend on multiple parameters. The last problem concerns the control law continuity at the switching time. We have chosen to use the dynamical sequencing technique to avoid the local minima problem. However, this technique requires to design all the controllers to be sequenced using the task function approach, which may appear limited. Thus, as shown above, although the current strategy offers interesting results, it is still hampered by some problems which are mainly due to a lack of generality. We then propose new solutions allowing to make our control strategy more generic and more efficient.

III. EXTENDED CONTROL STRATEGY

Now, we focus on our contribution. First, we have added a new anticipative controller which modifies the robot trajectory by considering the obstacle before entering the area surrounded by ξ_0 . Thus, the robot will begin turning sooner and will reach the security envelope in such a way that the avoidance motion is significantly eased and does not depend anymore on particular parameters. Then, we propose general transition conditions based on the prediction of the evolution of the visual features. Our last contribution will concern the extension of the sequencing methods to more general cases.

A. The anticipative controller

This controller is intended to provide the robot some anticipative skills to ease the avoidance motion. The idea is to make the robot turn before the obstacle becomes really dangerous to reach the security envelope with a proper orientation and a proper linear velocity. To do so, the controller is defined so that: (i) the robot direction is aligned with respect to the obstacle tangent \vec{T} (Fig. 2) and (ii) the robot linear velocity is equal to v_r . We propose the task function:

$$e_{anti} = [l - v_r t \quad \alpha]^T \quad (11)$$

where α is the same angle as in figure 2. It is important to note that no relative distance to the obstacle is imposed.

The aim of the anticipative controller is to modify the robot trajectory before it enters the area surrounded by ξ_0 . However, this task function just allows to control the mobile base, while it is also necessary to keep the target in the camera line of sight. We propose to use task function e_{pp} which is able to center the landmark in the image. Finally, we will consider $e_a = (e_{anti}^T \ e_{pp})^T$. Making it vanish using an exponential decrease, we get:

$$\dot{e}_a = J_a \dot{q}_a + A = -\lambda_a e_a \quad (12)$$

where λ_a is a positive scalar or a positive definite matrix and $\dot{q}_a = [v_a \ \omega_a \ \varpi_a]^T$ denotes the anticipation controller. J_a is given as shown below:

$$J_a = \begin{bmatrix} 1 & 0 & 0 \\ \chi_{oa} \cos(\alpha) & 1 & 0 \\ & & L_{Y_0} J_r \end{bmatrix} \quad (13)$$

We can now deduce the anticipative controller:

$$\dot{q}_a = J_a^{-1} (-\lambda_a e_a - A) \quad (14)$$

B. The general transition conditions

During the navigation the robot is driven using three controllers. The role of the supervision process is to select the most suitable controller according to the context. This decision will be taken on the base of the sensory data. The proposed supervision process is illustrated on figure 3. It is made of three states $S0$, $S1$ and $S2$ which respectively correspond to the tasks "IBVS", "Anticipate" and "Avoid obstacles". C_{ij} represents the transition condition from state S_i to S_j and depends on geometric parameters and/or prediction of the robot position. We present hereafter the different transition conditions C_{ij} and the parameters involved in the decision.

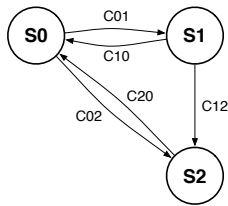


Fig. 3: The control process - S0: IBVS - S1: anticipation - S2: obstacle avoidance

$C01$ is the condition to start the anticipation motion while the robot is performing the IBVS task. The idea is to decide whether keeping on applying \dot{q}_{vs} is dangerous or not. We propose to predict the successive robot poses if it remains driven by \dot{q}_{vs} . To do so, we first place the laser range finder data in the robot frame. Then, using the visual features estimation equations and equation (6), we predict the next n_a values of the IBVS controller. These successive values are denoted here by $\hat{q}_{vs}(t_k + i_a T_s)$, with T_s the sampling time and $i_a \in [1, \dots, n_a]$. n_a corresponds to the number of predicted steps and allows to define the prediction horizon.

To predict the robot poses, we have solved (1) between t_k and $t_{k+1} = t_k + T_s$. Assuming that the velocities are kept constant during the sampling period, we get:

$$\begin{cases} x(k+1) = x(k) + \frac{v}{\omega} \left(\sin(\omega T_s + \theta(k)) - \sin(\theta(k)) \right) \\ y(k+1) = y(k) - \frac{v}{\omega} \left(\cos(\omega T_s + \theta(k)) - \cos(\theta(k)) \right) \\ \theta(k+1) = \omega T_s + \theta(k) \\ \vartheta(k+1) = \varpi T_s + \vartheta(k) \end{cases} \quad (15)$$

where v, ω, ϖ denote the values of the robot velocities at instant t_k . Now, from this result and using the n_a values of $\hat{q}_{vs}(t_k + i_a T_s)$, we can predict the successive robot poses which should be reached under the action of this controller. We can then easily deduce the distances between the robot predicted positions and the measured parts of the obstacle. We denote by $\hat{d}_{coll}(t_k + i_a T_s)$ these distances.

$$\text{If } \hat{d}_{coll}(t_k + i_a T_s) < d_0 \text{ for any } i_a \in [1, \dots, n_a]$$

$$\text{then } C01 = 1, \text{ otherwise } C01 = 0$$

In this way, some anticipation skills are provided to the vehicle. Moreover, it allows to consider only the most disturbing obstacles, that is those which are on the possible path of the robot.

As we have seen the prediction is done by using n_a times the visual features evolution and robot state evolution equations. This process is not really time consuming with respect to others processes such as image processing. For this reason, the value for n_a could only be selected based on the sampling time and the average robot speed in order to accurately anticipate an obstacle.

$C10$ is the condition to stop the anticipation movement and to start executing the IBVS task again. This condition must be set to 1 when the obstacle concerned by the anticipation is not dangerous anymore.

$$\text{If } \hat{d}_{coll}(t_k + i_a T_s) > d_0 \text{ for all } i_a \in [1, \dots, n_a]$$

$$\text{then } C10 = 1, \text{ otherwise } C10 = 0$$

$C12$ and $C02$ are the conditions to start the obstacle avoidance. They become true when the robot enters the area surrounded by ξ_0 . The obstacle is then considered as dangerous and must be strictly avoided.

$$\text{If } d_{coll}(t_k) < d_0 \text{ then } C12 = C02 = 1$$

$$\text{otherwise } C12 = C02 = 0$$

$C20$ is the condition to stop the avoidance motion and to start executing the IBVS task again. The approach is similar to the one used for $C10$. It consists in predicting the next successive n_{out} poses reached by the robot under the action of the IBVS controller.

$$\text{If } \hat{d}_{coll}(t_k + i_a T_s) > d_- \text{ for any } i_{out} \in [1, \dots, n_{out}]$$

$$\text{and } d_{coll}(t_k) > d_0$$

$$\text{then } C20 = 1, \text{ otherwise } C20 = 0$$

C. The global control law and its continuity

The global control law sent to the robot is defined by switching between the three controllers \dot{q}_{vs} , \dot{q}_{oa} and \dot{q}_a . We propose to extend the dynamical sequencing technique [19], [20]. As this technique allows to sequence task functions (and not directly controllers) [14], it is mandatory to use the task function formalism [13] to model the whole problem. Our goal is to propose a solution to be able to sequence any controllers. Let us denote by $\dot{q}_i(t)$ the current controller and by $\dot{q}_{i+1}(t)$ the next one. To guarantee the global control law continuity at the switching time t_s , it is necessary to ensure that:

$$\dot{q}(t) = \begin{cases} \dot{q}_i(t) & \forall t \leq t_s \\ \dot{q}_{i+1}(t) & \forall t \geq t_s \end{cases} \quad (16)$$

To this aim, we propose to design the global control law sent to the robot $\dot{q}(t)$ as follows:

$$\dot{q}(t) = \dot{q}_{i+1}(t) + (\dot{q}_i(t_s) - \dot{q}_{i+1}(t_s))e^{-\gamma(t-t_s)} \quad (17)$$

where γ is a positive scalar allowing to define the transient time. As we can see, $\dot{q}_{i+1}(t_s) = \dot{q}_i(t_s)$, which satisfies constraint (16). The computation of $\dot{q}(t)$ only requires the current and next controllers which can then be independently designed. It also guarantees that only one controller acts on the robot at a time, which was not the case of convex combination. Our third objective is then fulfilled.

IV. SIMULATION RESULTS

We now describe simulation tests demonstrating the control strategy efficiency. We propose hereafter two sets of results which have been obtained using MatlabTM software. The first ones (Fig. 4) are intended to show the interest of the proposed approach; the last one (Fig. 4(e)) allows to evaluate its efficiency in a more tricky environment. In every case, the goal is to perform a navigation task allowing to position the camera so that it faces the target.

First, we have set the control strategy parameters as follows:

$$\begin{aligned} \lambda_{vs} &= 0.6 & \lambda_{oa} &= 0.7 & \lambda_a &= 0.5 \\ \kappa_{oa} &= 0.1 & \gamma &= 500 \text{ ms} & T_s &= 0.1 \text{ s} \\ v_r &= 0.5 \text{ m/s} & n_a &= 25 & n_{out} &= 10 \end{aligned}$$

We have also introduced a ± 2 pixel noise on the image features which results on a 20 cm gaussian noise on the estimated depth value [18]. Finally, we have set the security distances (d_-, d_0) to $(0.3m, 0.5m)$ for the first simulation (see figure 4(a)) and to $(0.1m, 0.3m)$ for all the other ones. In all cases, the robot trajectory plotted in black is performed thanks to \dot{q}_{vs} , the one represented in green is realized using \dot{q}_a and the one shown in red is executed using \dot{q}_{oa} .

Now, we focus on the two first simulations represented on figures 4(a) and 4(b). They have been realized without the anticipation controller. As one can see in figure 4(a), the task is correctly performed thanks to a proper switch between \dot{q}_{vs} and \dot{q}_{oa} when the robot crosses envelope ξ_0 . On the contrary, in figure 4(b), the security distances have been reduced and controller \dot{q}_{oa} is not sufficient to guarantee non collision

(the robot velocity appears to be too high when entering the dangerous zone). The third simulation has been performed using the same set of parameters as in the second test. However, the anticipation controller \dot{q}_a has been added into the control process. As shown in figure 4(c), despite a small value of (d_-, d_0) , the task is successfully performed. Indeed, the prediction of the robot behavior controlled using \dot{q}_{vs} shows that it will make the robot enter the zone surrounded by ξ_0 ($\hat{d}_{coll} < d_0$). Thus, condition C01 is enabled, making the control law switch to \dot{q}_a . Thanks to this controller, the vehicle starts turning sooner than in figure 4(b) and reaches ξ_0 with an orientation which makes the avoidance motion easier. When ξ_0 is crossed, \dot{q}_{oa} is applied to the robot until the prediction shows that the obstacle is no more dangerous and that \dot{q}_{vs} can be applied anew. Thanks to our anticipative controller, non collision can be guaranteed and the navigation is less sensitive to the tuning of parameters such as d_0 or the control gains. It is illustrated in figure 4(d), where the obstacle avoidance parameters have been changed to $\lambda_{oa} = 1.5$ and $\kappa_{oa} = 0.3$. This set of parameters will lead to a collision in the first case (fig. 4(a)). Thanks to the anticipation controller the robot safety is guaranteed. In addition to these results, we have also evaluated the effect of the prediction horizons n_a and n_{out} on the control strategy. In figure 4(e), these horizons have been chosen so that the anticipation step is performed longer than in the third simulation. In this case, the use of the avoidance controller \dot{q}_{oa} appears to be unnecessary as the prediction shows that the obstacle is no longer dangerous. It is then possible to completely suppress the avoidance step, although it is not our goal here. Finally, we have also plotted the evolution of the robot velocities during the navigation task realized in figure 4(c). The corresponding results are presented on figure 4(f). As one can see, thanks to our sequencing technique, the corresponding speeds appear to be continuous at each switching instant. The global control law smoothness is preserved during the whole navigation task.

To definitely validate our approach, we have also considered a more tricky environment which is cluttered with three obstacles (two walls, one cylinder). This environment has been designed so that the robot has to go through a narrow passage to complete the mission. The obtained robot trajectory is shown on figure 5. As we can see, the task is successfully performed. The three controllers are sequenced in such a way that the robot does not collide with the walls and that it is able to navigate between them. It is particularly interesting to remark that, to cross the narrow passage, the robot is driven by visual servoing as the prediction shows that applying this controller allows to leave the dangerous zone. Once this phase is over, the anticipative controller is launched because the predicted distance \hat{d}_{coll} to the cylindrical obstacle drops under d_0 . Finally, when it has been verified that the last obstacle does not induce any risk, the IBVS controller is sent to the vehicle again and the goal is reached.

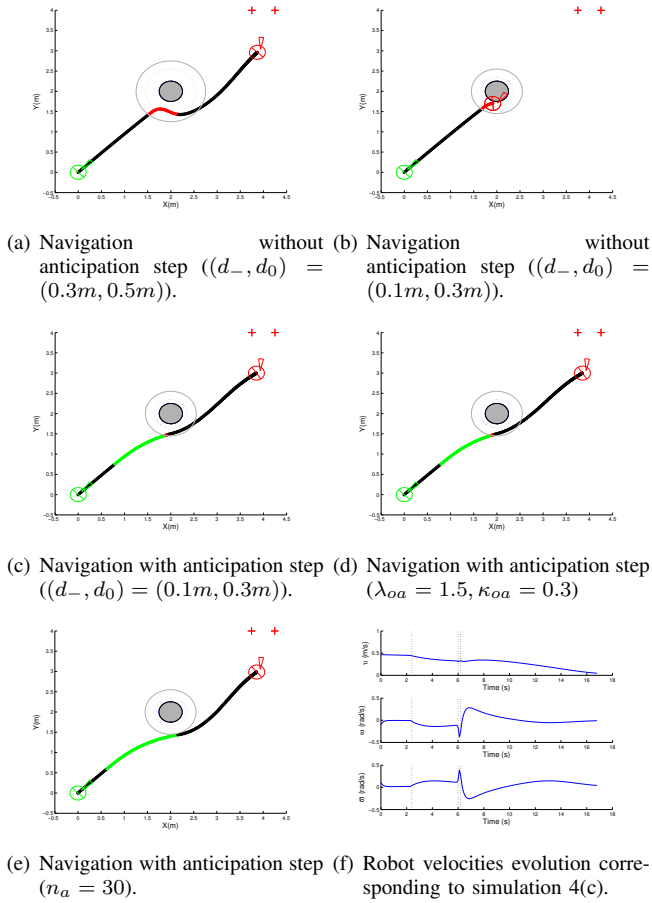


Fig. 4: Navigation simulations results - Black: visual servoing - Green: anticipation - Red: obstacle avoidance.

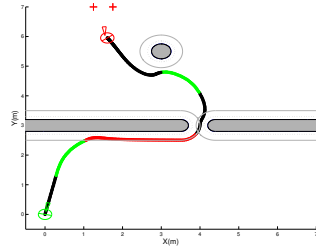


Fig. 5: A navigation task through narrow passages.

V. CONCLUSION

We have addressed the problem of the avoidance of unforeseen obstacles during a vision-based navigation task. We have extended previous strategies, making them more efficient and general. Three main contributions can be highlighted: first, the design of an anticipative controller able to make the robot reach the security envelope with a proper speed and a proper orientation. The sensitivity of the control strategy with respect to key parameters can then be significantly reduced. As second contribution, we have exhibited new general transitions conditions which allow to suppress local minima due to bad switching decisions. The third

contribution concerns the global control law smoothness.

As for future works, we first plan to integrate this strategy in our latest works [21] dedicated to visual navigation to improve the overall execution of the task. Furthermore, mobile obstacles bypassing appears to be a fundamental issue. We then plan to extend this strategy to take into account such obstacles.

REFERENCES

- [1] H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, and S. Thrun, *Principles of Robot Motion*. MIT Press, Boston, 2005.
- [2] R. Siegwart and I. Nourbakhsh, *Introduction to autonomous mobile robots*, ser. A bradford book, Intelligent robotics and autonomous agents series. The MIT Press, 2004.
- [3] F. Bonin-Font, F. Ortiz, and G. Oliver, "Visual navigation for mobile robots : a survey," *Journal of intelligent and robotic systems*, vol. 53, no. 3, p. 263, 2008.
- [4] C. Stachniss and W. Burgard, "An integrated approach to goal-directed obstacle avoidance under dynamic constraints for dynamic environment," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'02)*, Lausanne, Switzerland, October 2002.
- [5] B. Jur-Van-Den and M. Overmars, "Roadmap-based motion planning in dynamic environments," *IEEE Trans. on Robotics*, vol. 21, no. 5, pp. 885–897, October 2005.
- [6] S. Koenig and M. Likhachev, "Fast replanning for navigation in unknown terrain," *Robotics, IEEE Transactions on*, vol. 21, no. 3, pp. 354 – 363, june 2005.
- [7] Khatib, Jaouni, Chatila, and Laumond, "Dynamic path modification for car-like nonholonomic mobile robots," in *IEEE Int. Conf. on Robotics and Automation*, April 1997, pp. 490–496.
- [8] F. Lamiroux, D. Bonnafous, and O. Lefebvre, "Reactive path deformation for nonholonomic mobile robots," *Robotics, IEEE Transactions on*, vol. 20, no. 6, pp. 967 – 977, dec. 2004.
- [9] M. Khatib and R. Chatila, "An extended potential field approach for mobile robot sensor-based motions," in *Intelligent Autonomous Systems (IAS)*, I. Press, Ed., Karlsruhe, Germany, 1995, pp. 490–496.
- [10] O. Brock and O. Khatib, "High-speed navigation using the global dynamic window approach," in *IEEE International Conference on Robotics and Automation*, Detroit, USA, May 1999.
- [11] I. Ulrich and J. Borenstein, "Vfh*: Local obstacle avoidance with look-ahead verification," in *IEEE International Conference on Robotics and Automation*, San Francisco, May 2000.
- [12] J. Minguez, F. Lamiroux, and J. Laumond, *Springer Handbook of Robotics*. Springer, 2008, ch. Motion planning and obstacle avoidance, pp. 827–852.
- [13] Samson, Borgne, and Espiau, "Robot control : The task function approach," *Oxford science publications*, 1991.
- [14] V. Cadenat, D. Folio, and A. Durand Petiteville, "A comparison of two sequencing techniques to perform a vision-based navigation task in a cluttered environment," *Advanced Robotics*, 2012.
- [15] D. Folio and V. Cadenat, *Computer Vision - Treating Image Loss by using the Vision/Motion Link: A Generic Framework*. IN-TECH, 2008, ch. 4.
- [16] Espiau, Chaumette, and Rives, "A new approach to visual servoing in robotics," *IEEE Trans. Robot. Automat.*, vol. 8, pp. 313–326, 1992.
- [17] Pissard-Gibollet and Rives, "Applying visual servoing techniques to control a mobile handeye system," in *IEEE Int., Conf. on Robotics and Automation*, Nagoya, Japan, 1995.
- [18] A. Durand Petiteville, S. Duroola, V. Cadenat, and M. Courdesses, "Management of visual signal loss during image based visual servoing," in *European Control Conference (ECC'13)*, Zurich, July 2013.
- [19] P. Souères and V. Cadenat, "Dynamical sequence of multi-sensor based tasks for mobile robots navigation," in *SYROCO*, Wroclaw, Poland, September 2003.
- [20] N. Mansard and F. Chaumette, "Task sequencing for sensor-based control," *IEEE Trans. on Robotics*, vol. 23, no. 1, pp. 60–72, February 2007.
- [21] A. Durand Petiteville, S. Hutchinson, V. Cadenat, and M. Courdesses, "2d visual servoing for a long range navigation in a cluttered environment," in *50th IEEE Conference on Decision and Control and European Control Conference*, Orlando, USA, December 2011.