



**HAL**  
open science

## **A comparison of two sequencing techniques to perform a vision-based navigation task in a cluttered environment**

Viviane Cadenat, David Folio, Adrien Durand-Petiteville

### ► **To cite this version:**

Viviane Cadenat, David Folio, Adrien Durand-Petiteville. A comparison of two sequencing techniques to perform a vision-based navigation task in a cluttered environment. *Advanced Robotics*, 2012, 26 (5-6), pp.487-514. 10.1163/156855311X617470 . hal-00678670

**HAL Id: hal-00678670**

**<https://hal.science/hal-00678670>**

Submitted on 13 Mar 2012

**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.

*Full paper*

# Comparison of Two Sequencing Techniques to Perform a Vision-Based Navigation Task in a Cluttered Environment

Viviane Cadenat<sup>a,b,\*</sup>, David Folio<sup>c</sup> and Adrien Durand Petiteville<sup>a,b</sup>

<sup>a</sup> CNRS, LAAS, 7, avenue du Colonel Roche, F-31077 Toulouse Cedex 4, France

<sup>b</sup> Université de Toulouse, UPS, INSA, INP, ISAE, UT1, UTM, LAAS, F-31077 Toulouse Cedex 4, France

<sup>c</sup> Laboratoire PRISME, Ecole Nationale Supérieure d'Ingenieurs de Bourges, 88 Boulevard Lahitolle, F-18020 Bourges, France

Received 14 January 2011; accepted 1 March 2011

---

## Abstract

We address the problem of multi-sensor-based navigation in a cluttered environment for a non-holonomic robot. To perform such a successful and safe navigation, three controllers realizing, respectively, nominal vision-based navigation, obstacle bypassing and occlusion avoidance have been designed using the task function approach. Then it suffices to sequence them to realize the complete mission. To guarantee the control continuity when switching between two successive controllers, two sequencing approaches have been used and compared. Simulation results validate our work.

© Koninklijke Brill NV, Leiden and The Robotics Society of Japan, 2012

## Keywords

Task sequencing, visual servoing, obstacle avoidance, occlusion avoidance, task function

## 1. Introduction

Visual servoing techniques aim at controlling robot motion using vision data provided by a camera to reach a desired goal defined in the image [1]. Therefore, these techniques cannot be used if these features are lost during the robotic task. The idea is then to use methods allowing us to preserve the visibility of visual features during the whole execution of the mission. Most of them are dedicated to manipulator arms and propose to treat this kind of problem by using redundancy [2, 3], path planning [4], specific degrees of freedom (d.o.f.) [5, 6], zoom [7] or even by making a tradeoff with the nominal vision-based task [8].

---

\* To whom correspondence should be addressed. E-mail: cadenat@laas.fr

In a mobile robotics context, the issue is slightly different. Indeed, first of all, the successful realization of a vision-based navigation task in a cluttered environment requires us to preserve not only image data visibility, but also robot safety. Therefore, there is an additional objective with respect to the manipulator arm case. Furthermore, the number of available d.o.f. may be reduced and the use of redundancy is then more difficult. In our case, the considered robot is a non-holonomic vehicle equipped with an ultrasonic (US) sensor belt and a camera mounted on a pan platform. Only 3 d.o.f. are then available; redundancy will, thus, be very limited. In this work, we aim at designing a sensor-based control law allowing us to fulfill the two previously mentioned objectives — non-collision and landmark non-occlusion. The proposed control strategy is the sequel to previous works [9, 10]. It consists in a three-step method: (i) the navigation task is split into several subtasks to be sequenced, (ii) then a separate controller allowing us to realize each subtask is designed and finally (iii) the global control law is deduced by sequencing them. The implementation of such a control strategy requires a smooth transition between the different controllers. The literature provides two different approaches able to guarantee the global control law continuity. In the first approach, the controllers are merged using a simple convex combination based on time as in Ref. [11] or on parameters characterizing the risks of occlusions and collisions as in Refs [9, 10]. In the second approach, the controllers are dynamically sequenced (i.e., the control law is designed so that it is naturally smooth when the switch occurs). Few works use this second approach as it is a bit more difficult to handle than the first one. We may nonetheless mention the work by Souères and Cadenat who first proposed a formalism dedicated to dynamical sequencing and applied it to merge two controllers to perform a complex navigation task [12]. The same formalism was then used by Gao to improve the robotic task execution by dynamically sequencing multi-criteria visual-servoing controllers [13]. However, for these two cases, the switch depends on time and not on relevant events, such as an occlusion or a collision. Thus, an important part of this work will be devoted to the choice of switching conditions. Finally, let us also note that dynamical sequencing has been also used for manipulator arms and humanoid robots to switch smoothly between different secondary tasks [14]. However, as previously mentioned, the issue is rather different because the proposed approach strongly benefits from the redundancy with a number of available d.o.f. considerably larger than in our case.

In this paper, we deal with the problem of executing a vision-based navigation task amidst possibly occluding obstacles. The proposed control strategy relies on the continuous switch between three controllers, respectively, realizing nominal vision-based task, obstacle bypassing and occlusion avoidance. The transition is performed using two different methods, respectively, based on convex combination and dynamical sequencing. These two approaches are then analyzed and compared.

This paper is organized as follows. Sections 2 and 3 are dedicated to problem modeling and control issues. The obtained simulation results and a comparative

analysis are presented in Section 4. Section 5 presents the conclusions and future work.

## 2. Problem Modeling

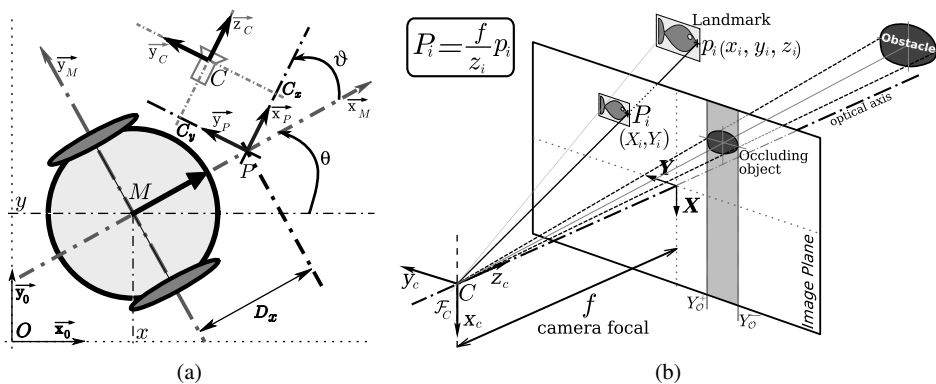
We first describe the robotic system before modeling the three above-mentioned subtasks — the nominal vision-based task, occlusion avoidance and collision avoidance.

### 2.1. Robotic System

We first model our robotic system to characterize the camera kinematic screw  $T^c$ . To this aim, considering Fig. 1a, we define the successive frames:  $\mathcal{F}_M (M, \vec{x}_M, \vec{y}_M, \vec{z}_M)$  linked to the robot,  $\mathcal{F}_P (P, \vec{x}_P, \vec{y}_P, \vec{z}_P)$  attached to the pan platform and  $\mathcal{F}_C (C, \vec{x}_C, \vec{y}_C, \vec{z}_C)$  linked to the camera. Let  $\vartheta$  be the direction of the pan platform with respect to  $\vec{x}_M$ ; where  $P$  is the pan platform center of rotation, and  $D_x$  is the distance between the robot reference point  $M$  and  $P$ . The control input is defined by  $\dot{q} = (v, \omega, \varpi)^T$ , where  $v$  and  $\omega$  are the cart linear and angular velocities, and  $\varpi$  is the pan platform angular velocity with respect to  $\mathcal{F}_M$ . The kinematic screw  $T^c$  is related to the control input by the robot Jacobian  $\mathbf{J}$ :  $T^c = \mathbf{J}\dot{q}$ . As the camera is constrained to move horizontally, it is sufficient to consider a reduced kinematic screw  $T_r^c = (V_{\vec{y}_C}, V_{\vec{z}_C}, \Omega_{\vec{x}_C})^T$  and a reduced Jacobian matrix  $\mathbf{J}_r$  as:

$$T_r^c = \begin{pmatrix} V_{\vec{y}_C} \\ V_{\vec{z}_C} \\ \Omega_{\vec{x}_C} \end{pmatrix} = \begin{pmatrix} -\sin(\vartheta) & D_x \cos(\vartheta) + C_x & C_x \\ \cos(\vartheta) & D_x \sin(\vartheta) - C_y & -C_y \\ 0 & -1 & -1 \end{pmatrix} \begin{pmatrix} v \\ \omega \\ \varpi \end{pmatrix} = \mathbf{J}_r \dot{q}, \quad (1)$$

where  $C_x$  and  $C_y$  are the coordinates of  $C$  along axes  $\vec{x}_P$  and  $\vec{y}_P$  (Fig. 1). Notice that  $\mathbf{J}_r$  is a regular matrix as  $\det(\mathbf{J}_r) = D_x \neq 0$ .



**Figure 1.** System modeling. (a) Cart-like robot with a camera mounted on a pan-platform. (b) The pinhole camera model.

## 2.2. Sensor-Based Model

Recalling that our goal is to realize a vision-based navigation task with respect to a static landmark, we denote by  $\mathbf{s}$  a set of relevant visual data characterizing this landmark and by  $\mathbf{z}$  a vector describing its depth (Fig. 1b). The focal length  $f$  is supposed to be equal to 1 as in Ref. [15]. For a fixed landmark, the variation of the visual signal  $\dot{\mathbf{s}}$  is related to the camera kinematic skew  $\mathcal{T}_r^c$  by means of the interaction matrix  $\mathbf{L}_{(\mathbf{s},\mathbf{z})}$  as [15]:

$$\dot{\mathbf{s}} = \mathbf{L}_{(\mathbf{s},\mathbf{z})} \mathcal{T}_r^c. \quad (2)$$

$\mathbf{L}_{(\mathbf{s},\mathbf{z})}$  allows to relate the visual features motion in the image to the three-dimensional camera motion. It depends mainly on the depth  $\mathbf{z}$  (which is not always available online) and on the considered visual data. Expressions of  $\mathbf{L}_{(\mathbf{s},\mathbf{z})}$  are available for different kinds of features, such as points, straight lines, circles [15], image moments [16], etc.

## 2.3. Vision-Based Task

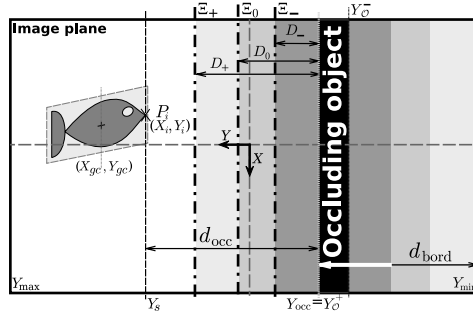
Here, we only consider the nominal vision-based navigation task to be performed. The goal of the considered task is to position the embedded camera with respect to a fixed landmark. To this aim, we have applied the visual servoing technique given in Ref. [15] to mobile robots as in Ref. [11]. The proposed approach relies on the task function formalism [17]. This formalism deals with a special kind of output functions called task functions. The idea is to characterize the desired mission by a suitable task function  $\mathbf{e}(q(t))$  so that its regulation to zero leads to a successful execution of the task. A sufficient condition that guarantees the control problem to be well conditioned is that  $\mathbf{e}$  is admissible. Indeed, this property ensures the existence of a diffeomorphism between the task space and the space of generalized coordinates  $\mathbb{Q}$ , so that the ideal trajectory  $q^*(t)$  corresponding to  $\mathbf{e}(q^*(t)) = 0$  is unique. This condition is fulfilled if  $\frac{\partial \mathbf{e}(q(t))}{\partial q}$  is regular around  $q^*(t)$  [17]. Using the task function formalism, the visual servoing task is classically defined as the regulation of the following error function to zero [1]:

$$\mathbf{e}_{vs} = \mathbf{L}_{(\mathbf{s}^*)}^+ (\mathbf{s} - \mathbf{s}^*), \quad (3)$$

where  $\mathbf{s}^*$  is the visual features desired value and  $\mathbf{L}^+$  denotes the matrix interaction pseudo-inverse.

## 2.4. Occlusion Avoidance Task

Now, we consider that only occlusions can occur — an occluding object can enter the camera field of view to temporarily hide the image features, but is too far to induce a collision risk. Our goal is to define the occlusion avoidance task function so that the image features visibility is preserved along the execution of the mission. We first describe the way we determine the occlusion before defining a suitable task function.



**Figure 2.** Occlusion detection.

**2.4.1. Occlusion Detection**

Here, we suppose that an occluding object  $\mathcal{O}$  lies in the camera field of view. Its projection appears in the image plane as shown on Fig. 1b, and we denote by  $Y_{\mathcal{O}}^-$  and  $Y_{\mathcal{O}}^+$  the ordinates of its left and right borders. The proposed strategy only relies on the detection of these two borders. As the camera is constrained to move in the horizontal plane, there is no loss of generality in stating the reasoning on  $Y_{\mathcal{O}}^-$  and  $Y_{\mathcal{O}}^+$ . Considering Fig. 2,  $Y_{\min}$  and  $Y_{\max}$  represent the ordinates of the two image sides, and we introduce the following distances:

- $d_{\text{occ}}$  characterizes the distance before occlusion (i.e., the shortest distance between  $s$  and  $\mathcal{O}$ ). It can be defined as:  $d_{\text{occ}} = |Y_s - Y_{\text{occ}}|$ , where  $Y_s$  is the ordinate of the closest point  $P_i$  to  $\mathcal{O}$ , while  $Y_{\text{occ}}$  represents the closest border of  $\mathcal{O}$  to the visual features (in the case of Fig. 2,  $Y_{\text{occ}} = Y_{\mathcal{O}}^+$ ). When there is no occluding object in the image,  $Y_{\text{occ}}$  is set to the closest image side ( $Y_{\min}$  or  $Y_{\max}$ ) to keep the visual features in the camera line of sight.
- $d_{\text{bord}}$  corresponds to the distance separating  $\mathcal{O}$  and the opposite image side to the visual features. It can be defined as:  $d_{\text{bord}} = |Y_{\text{occ}} - Y_{\text{bord}}|$ , where  $Y_{\text{bord}}$  corresponds to the image border towards which  $\mathcal{O}$  must move to leave the image without occluding the target.
- $D_+$ ,  $D_0$  and  $D_-$  define three envelopes  $\Xi_+$ ,  $\Xi_0$  and  $\Xi_-$  that surround the occluding object. The envelope  $\Xi_+$  delimits the region inside which the risk of occlusion is detected and  $\Xi_-$  the region where the danger of occlusion is the highest.  $\Xi_0$  defines the zone where it is necessary to start avoiding the occlusion.

**Remark 1.** The above occlusion detection method has been presented in the case of a sole occluding obstacle for the sake of clarity. When several objects are lying in the image, we compute all the corresponding distances  $d_{\text{occ}}$  and we consider only the smallest one. In this way, we first take into account the closest occluding object, as it is the most dangerous one. This approach is naturally limited if too many objects are lying in the image or if their positions in the image do not allow

us to realize the navigation task while avoiding occlusions. In such a case, the robot will be stopped to preserve its safety. Other approaches consisting in tolerating occlusions must then be used (e.g., see Ref. [18]).

2.4.2. *Task Function Design*

The redundant task function formalism consists of defining a redundant task  $\mathbf{e}_1$ , which is a low-dimensional task that does not constrain all the robot d.o.f. The idea is to benefit from this redundancy to perform an additional objective. The latter can be modeled as a cost function  $h$  to be minimized under the constraint that  $\mathbf{e}_1$  is perfectly performed. The resolution of this optimization problem leads us to regulate the following global task [17]:

$$\mathbf{e} = \mathbf{W}^+ \mathbf{e}_1 + \beta (\mathbb{I} - \mathbf{W}^+ \mathbf{W}) \mathbf{g}, \tag{4}$$

where  $\mathbf{W}^+ = \mathbf{W}^T (\mathbf{W} \mathbf{W}^T)^{-1}$ ,  $\mathbf{g} = \frac{\partial h}{\partial \mathbf{q}}$  and  $\beta > 0$ . Under some assumptions, which are verified if  $\mathbf{W} = \frac{\partial \mathbf{e}_1}{\partial \mathbf{q}}$ , the task Jacobian  $\frac{\partial \mathbf{e}}{\partial \mathbf{q}}$  is positive-definite around the ideal trajectory  $q^*(t)$ , ensuring that  $\mathbf{e}$  is admissible [17].

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 priority task. The target tracking will then be considered as the secondary objective and will be modeled as a criterion  $h_s$  to be minimized. We propose the following occlusion avoidance (oa) task function:

$$\mathbf{e}_{oa} = \mathbf{J}_{occ}^+ \mathbf{e}_{occ} + \beta_{oa} (\mathbb{I} - \mathbf{J}_{occ}^+ \mathbf{J}_{occ}) \mathbf{g}_s, \tag{5}$$

where  $\mathbf{e}_{occ}$  represents the redundant task function allowing us to avoid the occlusions,  $\mathbf{J}_{occ} = \frac{\partial \mathbf{e}_{occ}}{\partial \mathbf{q}}$ , and  $\beta_{oa} > 0$  and  $\mathbf{g}_s = \frac{\partial h_s}{\partial \mathbf{q}}$  are as explained above. We propose the following criterion to track the target and keep it in the line of sight of the camera:  $h_s = \frac{1}{2} (\mathbf{s} - \mathbf{s}^*)^T (\mathbf{s} - \mathbf{s}^*)$ . Its gradient is then given by:  $\mathbf{g}_s = ((\mathbf{s} - \mathbf{s}^*)^T \mathbf{L}_{(s)} \mathbf{J}_f)^T$ . Now, let us define the priority task function  $\mathbf{e}_{occ}$  to avoid occlusions. We propose the following redundant task function:

$$\mathbf{e}_{occ} = \left( \tan \left( \frac{\pi}{2} - \frac{\pi}{2} \cdot \frac{d_{occ}}{D_+} \right) \right) \cdot d_{bord}. \tag{6}$$

The first component allows us to avoid target occlusions: it increases when the occluding object gets 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,  $\forall d_{occ} \geq D_+$ ,  $\mathbf{e}_{occ}$  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 (i.e., they can be simultaneously realized) 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  $d_{bord}$  (Fig. 2). The Jacobian  $\mathbf{J}_{occ} = \frac{\partial \mathbf{e}_{occ}}{\partial q}$  expresses as follows:

$$\mathbf{J}_{occ} = \frac{\partial \mathbf{e}_{occ}}{\partial q} = \begin{pmatrix} -\frac{1}{D_+} \frac{\pi}{2} \varepsilon_{occ} (1 + \tan^2(\frac{\pi}{2} - \frac{\pi}{2} \cdot \frac{d_{occ}}{D_+})) (\frac{\partial Y_s}{\partial q} - \frac{\partial Y_{occ}}{\partial q}) \\ \varepsilon_{bord} \frac{\partial Y_{occ}}{\partial q} \end{pmatrix}, \quad (7)$$

where  $\varepsilon_{occ} = \text{sign}(Y_s - Y_{occ})$  and  $\varepsilon_{bord} = \text{sign}(Y_{occ} - Y_{bord})$ .  $\frac{\partial Y_s}{\partial q}$  and  $\frac{\partial Y_{occ}}{\partial q}$  are deduced from the optic flow equations [10]:

$$\begin{cases} \frac{\partial Y_s}{\partial q} = \begin{pmatrix} -\frac{1}{z_s} & \frac{Y_s}{z_s} & 1 + Y_s^2 \end{pmatrix} \mathbf{J}_r \\ \frac{\partial Y_{occ}}{\partial q} = \begin{pmatrix} -\frac{1}{z_{occ}} & \frac{Y_{occ}}{z_{occ}} & 1 + Y_{occ}^2 \end{pmatrix} \mathbf{J}_r \end{cases} \quad (8)$$

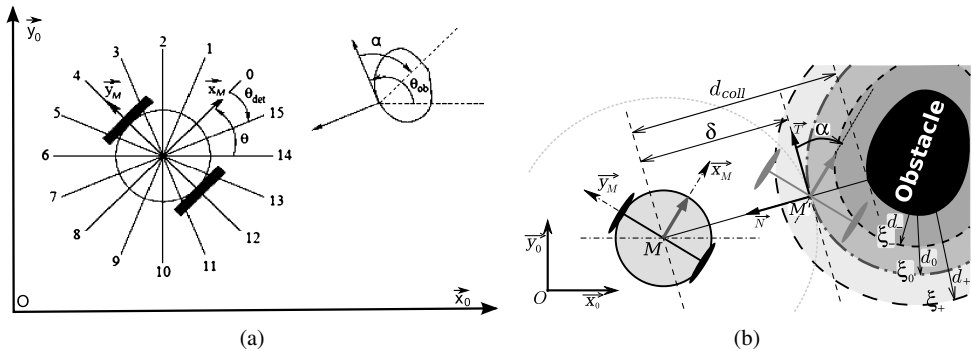
with  $z_s$  and  $z_{occ}$  the depth of the visual target and of the occluding object.

### 2.5. Collision Avoidance Task

Here, we aim at modeling the collision avoidance task. The robot is close to an obstacle; therefore, a collision may occur while an occlusion is possible depending on the obstacle height. In such a case, it will be necessary to guarantee non-collision while preserving the visibility of image features if needed. We first describe how to detect the risk of collision. We then show how to ensure a safe motion for the robot and how to take into account the possible occlusions that may occur in the obstacle neighborhood.

#### 2.5.1. Collision Detection

Collision detection relies on the proximetric data provided by 16 US sensors positioned all around the robot as shown in Fig. 3a. As one can see, any obstacle lying in a 360° zone around the vehicle can be detected. The US data are processed to compute the distance  $d_{coll}$ , and the relative orientation  $\alpha$  between the robot and the obstacle (Fig. 3) as follows. (i) The US sensors are successively fired. (ii) We associate the smallest distance value to the closest obstacle, we determine the number



**Figure 3.** Collision detection and US sensor modeling. (a) US sensor modeling. (b) Collision detection.



$n \in [0, 15]$  of the corresponding sensor and we compute the angle  $\theta_{\text{det}} = \frac{2n\pi}{16}$  between two successive sensors. (iii) We express the angle  $\theta_{\text{ob}}$  between the tangent to the obstacle and the  $x$ -axis of the world frame. We get:  $\theta_{\text{ob}} = \theta + \theta_{\text{det}} + \varepsilon \frac{\pi}{2}$  where  $\varepsilon = \pm 1$  depends on the direction of the avoidance motion. (iv) We deduce the angle  $\alpha = \theta - \theta_{\text{ob}}$ . Naturally, using such sensors, only rough values of  $d_{\text{coll}}$  and  $\alpha$  can be obtained. However, their precision is sufficient to successfully perform obstacle avoidance, as shown in Ref. [19].

To evaluate the risk of collision, three envelopes are defined around each obstacle (Fig. 3b). The first one,  $\xi_+$ , located at a distance  $d_+$ , surrounds the zone inside which the obstacle is detected. 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,  $\xi_0$ , located at a lower distance  $d_0$ , constitutes the virtual path along which the reference point  $M$  will move around to safely avoid the obstacle. The last one,  $\xi_-$ , defines the region inside which the risk of collision is maximal.

### 2.5.2. Task Function Design

Now, let us design the collision avoidance task function. The chosen strategy consists of making the vehicle avoid the obstacle by following the security envelope  $\xi_0$  as shown in Fig. 3b. To this aim, we have chosen to use the path-following formalism introduced in Ref. [20]. This formalism consists in defining a Frenet frame whose origin  $M'$  is the orthogonal projection of the robot reference point  $M$  on the reference path (here  $\xi_0$ ). Now, let  $\delta = d_{\text{coll}} - d_0$  be the signed distance between  $M$  and  $M'$ . With respect to the moving frame, the dynamics of the error terms  $(\delta, \alpha)$  are described by:

$$\begin{cases} \dot{\delta} = v \sin \alpha \\ \dot{\alpha} = \omega - v \chi \cos \alpha \end{cases} \quad \text{with } \chi = \frac{1/R}{1 + (\sigma/R)\delta}, \tag{9}$$

where  $R$  is the curvature radius of the obstacle and  $\sigma = \{-1, 0, +1\}$  depending on the sense of the robot motion around the obstacle. The path following problem is classically defined as the search for a controller  $\omega$  allowing us to steer the pair  $(\delta, \alpha)$  to  $(0, 0)$  under the assumption that  $v$  never vanishes to preserve the system controllability.

Classically, the path following problem requires us to use nonlinear control techniques. Here, we propose to solve it using the task function formalism. To this aim, we have to find a task function whose regulation to zero will make  $\delta$  and  $\alpha$  vanish while ensuring  $v \neq 0$ . We propose the following redundant task function  $\mathbf{e}_{\text{coll}}$ :

$$\mathbf{e}_{\text{coll}} = \begin{pmatrix} l - v_{\text{r}} t \\ \delta + k\alpha \end{pmatrix}, \tag{10}$$

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 us to regulate the linear velocity of the mobile base to a non-zero constant value  $v_{\text{r}}$  ( $v_{\text{r}}$  must be chosen small enough to let the robot sufficiently slow down to avoid collisions when entering the critical zone).

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  $\xi_0$  during the obstacle avoidance. The second component of  $\mathbf{e}_{\text{coll}}$  can be seen as a sliding variable whose regulation to zero makes both  $\delta$  and  $\alpha$  vanish (see Ref. [21] for a detailed proof). The value of  $k$  determines the relative convergence velocity of  $\delta$  and  $\alpha$  as the sliding variable converges. Therefore, the regulation to zero of  $\mathbf{e}_{\text{coll}}$  guarantees that the robot follows the security envelope  $\xi_0$ , ensuring non-collision. As the chosen task function does not constrain the whole d.o.f. 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:  $h_{\text{occ}} = \frac{1}{d_{\text{occ}}}$ . Thus, the collision avoidance (ca) task function  $\mathbf{e}_{\text{ca}}$  expresses as:

$$\mathbf{e}_{\text{ca}} = \mathbf{J}_{\text{coll}}^+ \mathbf{e}_{\text{coll}} + \beta_{\text{ca}} (\mathbb{I} - \mathbf{J}_{\text{coll}}^+ \mathbf{J}_{\text{coll}}) \mathbf{g}_{\text{occ}}, \tag{11}$$

where  $\beta_{\text{ca}} > 0$  and the gradient  $\mathbf{g}_{\text{occ}}$  is given by  $\mathbf{g}_{\text{occ}} = \frac{\partial h_{\text{occ}}}{\partial q} = -\frac{1}{d_{\text{occ}}^2} \frac{\partial d_{\text{occ}}}{\partial q} = -\frac{\varepsilon_{\text{occ}}}{d_{\text{occ}}^2} \left( \frac{\partial Y_s}{\partial q} - \frac{\partial Y_{\text{occ}}}{\partial q} \right)^T$ . Finally, a straightforward calculus shows that the task Jacobian  $\mathbf{J}_{\text{coll}}$  is given by:

$$\mathbf{J}_{\text{coll}} = \frac{\partial \mathbf{e}_{\text{coll}}}{\partial q} = \begin{pmatrix} 1 & 0 & 0 \\ \sin \alpha - k \chi \cos \alpha & k & 0 \end{pmatrix}. \tag{12}$$

### 3. Control Design

At this step, three task functions have been designed: the first,  $\mathbf{e}_{\text{vs}}$  (3), models the nominal vision-based task in the free space, the second,  $\mathbf{e}_{\text{oa}}$  (5), treats the occlusions generated by obstacles located far from the robot, and the third,  $\mathbf{e}_{\text{ca}}$  (11), handles collision problems while avoiding occlusions if needed. Now, it remains to determine three controllers allowing us to regulate them to zero and to select a convenient sequencing technique. In fact, these two problems are closely related because the control law discontinuity at the switching instant comes from the separate computation of the controllers to be sequenced. As mentioned in Section 1, two kinds of methods may be used to suppress it: either the different controllers are independently designed and then the global control law smoothness is ensured thanks to convex combination [9–11] or the smoothness constraint is directly taken into account in the design step so that the obtained control law is naturally continuous [12–14]. We successively consider these two different approaches.

#### 3.1. Convex Combination

In this case, the controllers are merged thanks to convex combinations depending on time [11] or on parameters that significantly characterize the task execution [9, 10]. Therefore, when using such techniques, the global control law smoothness only depends on the convex combination parameters. The controllers to be sequenced can then be separately designed, using any convenient dynamics, provided that it

allows us to guarantee that the considered task functions will be efficiently regulated to zero. In this work, we have applied classical results from the literature to determine the three desired controllers. We briefly describe the design techniques for each subtask before proposing a global control strategy.

3.1.1. *Design of the Controllers to be Sequenced*

- *Visual servoing controller.* To make  $\mathbf{e}_{vs}$  vanish, a classical solution consists of imposing an exponential decrease, i.e.,  $\dot{\mathbf{e}}_{vs} = -\gamma_{vs}\mathbf{e}_{vs}$ , where  $\gamma_{vs}$  is a positive scalar or a positive definite matrix [1]. The visual servoing controller can then be written as:

$$\dot{\mathbf{q}}_{vs} = (\mathbf{L}_{(s^*)}^+ \mathbf{L}_{(s)} \mathbf{J}_r)^{-1} (-\gamma_{vs}) \mathbf{e}_{vs}. \tag{13}$$

Note that, as  $\mathbf{J}_r$  is a regular matrix, the stability of this control scheme is mainly related to the positivity of  $\mathbf{L}_{(s)} \mathbf{L}_{(s^*)}^+$  and that the admissibility of the task requires that this product remains always invertible [1].

- *Occlusion avoidance controller.* Now, we address the occlusion avoidance controller determination. Our goal is then to design a controller allowing us to regulate task function  $\mathbf{e}_{oa}$  to zero by imposing an exponential decay. As  $\mathbf{J}_{occ}$  and  $\beta_{oa}$  are chosen to fulfill the assumptions of the redundant task formalism [17], the task Jacobian  $\mathbf{J}_{oa} = \frac{\partial \mathbf{e}_{oa}}{\partial \mathbf{q}}$  is positive definite around  $\mathbf{q}^*(t)$  and  $\mathbf{e}_{oa}$  is admissible. This result allows us to simplify the control synthesis, as it can be shown that a controller making  $\mathbf{e}_{oa}$  vanish is given by [15]:

$$\dot{\mathbf{q}}_{oa} = -\gamma_{oa} \mathbf{e}_{oa}, \tag{14}$$

where  $\gamma_{oa}$  is a positive scalar or a positive definite matrix.

- *Collision avoidance controller.* Here, we consider the problem of designing a control law able to drive safely the robot in the obstacle vicinity. Using the same previous reasoning to regulate  $\mathbf{e}_{ca}$  to zero, we obtain:

$$\dot{\mathbf{q}}_{ca} = -\gamma_{ca} \mathbf{e}_{ca}, \tag{15}$$

where  $\gamma_{ca}$  is a positive scalar or a positive definite matrix.

**Remark 2.** As controllers (14) and (15) are computed by approximating  $\mathbf{J}_{oa}$  and  $\mathbf{J}_{ca}$ , the exponential decrease is lost. Indeed, it can be shown that the closed-loop evolutions of  $\mathbf{e}_{oa}$  and  $\mathbf{e}_{ca}$  are expressed by:

$$\dot{\mathbf{e}}_{oa} = -\mathbf{J}_{oa} \hat{\mathbf{J}}_{oa}^{-1} \gamma_{oa} \mathbf{e}_{oa} \quad \text{and} \quad \dot{\mathbf{e}}_{ca} = -\mathbf{J}_{ca} \hat{\mathbf{J}}_{ca}^{-1} \gamma_{ca} \mathbf{e}_{ca}, \tag{16}$$

where  $\hat{\mathbf{J}}_{oa}$  and  $\hat{\mathbf{J}}_{ca}$  correspond to the approximations of  $\mathbf{J}_{oa}$  and  $\mathbf{J}_{ca}$ . When  $\hat{\mathbf{J}}_{oa} = \hat{\mathbf{J}}_{ca} = Id$ , stability is preserved only if  $\mathbf{J}_{oa}$  and  $\mathbf{J}_{ca}$  are positive definite. This last condition is verified if the assumptions of the redundant task function formalism are fulfilled [17], which is the case in our work. A detailed proof can be found in Ref. [15].

### 3.1.2. Global Control Law Design

At this step, the controllers allowing us to perform the three desired subtasks have been designed. Now, it remains to select the right one depending on the environment (i.e., depending on the risk of collisions and/or occlusions). To this aim, we use a convex combination and we propose the following global control law:

$$\dot{q} = (1 - \mu_{\text{coll}})[(1 - \mu_{\text{occ}})\dot{q}_{\text{vs}} + \mu_{\text{occ}}\dot{q}_{\text{oa}}] + \mu_{\text{coll}}\dot{q}_{\text{ca}}, \quad (17)$$

where  $\dot{q}_{\text{vs}}$ ,  $\dot{q}_{\text{oa}}$  and  $\dot{q}_{\text{ca}}$  are, respectively, given by (13), (14) and (15).  $\mu_{\text{occ}}$  and  $\mu_{\text{coll}} \in [0, 1]$  allow us to switch continuously from one control to the other depending on the risk of occlusion and of collision. As one can see, the proposed strategy guarantees that the collision avoidance is performed as a priority, guaranteeing the robot safety. Several cases may occur:

- If there is no occluding object  $\mathcal{O}$  in the neighborhood of the visual features  $\mathbf{s}$  ( $d_{\text{occ}} > D_0$ ) and no obstacle in the vicinity of the vehicle ( $d_{\text{coll}} > d_0$ ),  $\mu_{\text{occ}}$  and  $\mu_{\text{coll}}$  are fixed to 0 and the sole visual control  $\dot{q}_{\text{vs}}$  is used.
- If the visual features  $\mathbf{s}$  enter the zone delimited by  $\Xi_0$  ( $d_{\text{occ}} < D_0$ ), the danger of occlusion becomes higher and  $\mu_{\text{occ}}$  progressively increases to reach 1 when they cross  $\Xi_-$ . Therefore, while the visual features remain between  $\Xi_0$  and  $\Xi_-$ , the robot is controlled by a linear combination of  $\dot{q}_{\text{vs}}$ ,  $\dot{q}_{\text{oa}}$  and possibly  $\dot{q}_{\text{ca}}$  (see Remark 3). If the action of the global controller is sufficient to avoid the occlusions, the visual features may naturally leave the critical zone defined by  $\Xi_0$  and  $\mu_{\text{occ}}$  goes back to 0 without having reached 1. On the contrary, once  $\Xi_-$  is crossed,  $\mu_{\text{occ}}$  is maintained to 1 until  $\mathcal{O}$  leaves the image ( $d_{\text{bord}} = 0$ ) or at least goes out of the critical zone ( $d_{\text{occ}} > D_0$ ) (i.e., the risk of occlusion is considered to be sufficiently reduced). Therefore, the leaving condition is obtained when either  $d_{\text{bord}} = 0$  or  $d_{\text{occ}} = D_0$ . When one of these two events occurs,  $\mu_{\text{occ}}$  is progressively reduced from 1 to 0 and vanishes once the visual features cross  $\Xi_+$ . Following this reasoning,  $\mu_{\text{occ}}$  depends on the distance between the occluding object and the visual features in the image (i.e.,  $d_{\text{occ}}$ ).
- If the mobile base enters the zone surrounded by  $\xi_0$  ( $d_{\text{coll}} < d_0$ ), the danger of collision rises and  $\mu_{\text{coll}}$  is continuously increased from 0 to reach 1 when  $d_{\text{coll}} < d_-$ . If  $\xi_-$  is never crossed, the robot naturally leaves the critical zone defined by  $\xi_0$  and  $\mu_{\text{coll}}$  is brought back to 0, once  $\xi_+$  is crossed. If  $\mu_{\text{coll}}$  reaches 1, the collision risk 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  $\dot{q}_{\text{ca}}$  is applied to the vehicle once  $\mu_{\text{coll}}$  is set to 1. In this way, the robot is controlled using the sole controller  $\dot{q}_{\text{ca}}$ , allowing us to guarantee non-collision while performing the occlusion avoidance at best. The robot is then brought back on the security envelope  $\xi_0$  and follows it until the obstacle is overcome. This event occurs when the camera and the mobile base have the same direction  $\theta = \vartheta$ . A flag LEAVE is then positioned to 1 and  $\mu_{\text{coll}}$  is rapidly decreased to vanish on  $\xi_+$ .

**Table 1.**  
Switching strategy

	$\mu_{occ} = 0$	$\mu_{occ} \in ]0, 1[$	$\mu_{occ} = 1$
$\mu_{coll} = 0$	$\dot{q} = \dot{q}_{vs}$	$\dot{q}_{vs} \leftrightarrow \dot{q}_{oa}$	$\dot{q} = \dot{q}_{oa}$
$\mu_{coll} \in ]0, 1[$	$\dot{q}_{vs} \leftrightarrow \dot{q}_{ca}$	$\dot{q} = f(\dot{q}_{vs}, \dot{q}_{oa}, \dot{q}_{ca})$	$\dot{q}_{oa} \leftrightarrow \dot{q}_{ca}$
$\mu_{coll} = 1$	$\dot{q} = \dot{q}_{ca}$	$\dot{q} = \dot{q}_{ca}$	$\dot{q} = \dot{q}_{ca}$

**Remark 3.** The different envelopes are chosen close enough in order to reduce the duration of the transition phase and ensure that the robot will be rapidly controlled by the most relevant controller depending on the environment.

Naturally,  $\mu_{coll}$  and  $\mu_{occ}$  can be defined by any expression satisfying the previous constraints. A possible choice is given as [10]:

$$\left\{ \begin{array}{ll} \mu_{occ} = 0 & \text{if } d_{occ} > D_0 \text{ and } OCCLU = 0 \\ \mu_{occ} = \frac{d_{occ} - D_0}{D_- - D_0} & \text{if } d_{occ} \in [D_-, D_0] \text{ and } OCCLU = 0 \\ \mu_{occ} = \frac{d_{occ} - D_+}{D_0 - D_+} & \text{if } d_{occ} \in [D_0, D_+] \text{ and } d_{occ} = D_0 \\ \mu_{occ} = \frac{d_{occ} - D_+}{D_1 - D_+} & \text{if } d_{occ} \in [D_1, D_+] \text{ and } (d_{bord} = 0 \text{ or } d_{occ} \geq D_0) \\ \mu_{occ} = 1 & \text{otherwise,} \end{array} \right. \quad (18)$$

where OCCLU is the flag indicating that  $\mu_{occ}$  has reached its maximal value 1 and  $D_1$  is the value of the distance  $d_{occ}$  for which the leaving condition has been fulfilled. As for  $\mu_{coll}$ , we propose the following definition [10]:

$$\left\{ \begin{array}{ll} \mu_{coll} = 0 & \text{if } (d_{coll} > d_0) \text{ and } AVOID = 0 \text{ and } LEAVE = 0 \\ \mu_{coll} = \frac{d_{coll} - d_0}{d_- - d_0} & \text{if } d_{coll} \in [d_-, d_0] \text{ and } AVOID = 0 \\ \mu_{coll} = \frac{d_{coll} - d_+}{d_1 - d_+} & \text{if } d_{coll} \in [d_1, d_+] \text{ and } LEAVE = 1 \\ \mu_{coll} = 1 & \text{otherwise,} \end{array} \right. \quad (19)$$

where  $d_1$  is defined by the distance  $d_{coll}$  when leave = 1. The switching strategy is summarized in Table 1.

### 3.2. Dynamical Sequencing

As previously explained, the control law discontinuity at the switching instant arises when the controllers to be sequenced are separately computed. In the above solution, the smoothness is *a posteriori* imposed thanks to a convex combination. Here, the idea is to propose a method allowing us to naturally obtain a continuous control law at the switching time. It is then necessary to guarantee that the values of the

two successive controllers are identical at this time. Souères and Cadenat have proposed a formalism allowing us to fulfill this particular property [12]. We first briefly describe this formalism before applying it in our specific case.

### 3.2.1. Sequencing Formalism

This formalism relies on the admissibility property. As previously mentioned, this property allows us to define a local diffeomorphism between the task space and the generalized coordinates space  $\mathbb{Q}$ . Souères and Cadenat have shown that it can also be used to define a structure of differential manifold on  $\mathbb{Q}$  [12]. The evolution of the next task induced by the execution of the current task can then be uniquely expressed. From this result, it is possible to deduce a controller able to take into account the constraints on the evolutions of the two tasks to be sequenced at the switching time. The control law smoothness can then be guaranteed. Hereafter, we only present the general reasoning. Theoretical aspects can be found in Ref. [12]. We consider two tasks to be sequenced. We assume that they can be modeled by two admissible task functions  $\mathbf{e}_1(q(t))$  and  $\mathbf{e}_2(q(t))$ , which implies that their Jacobian matrices  $\mathbf{J}_1 = \mathbf{J}_1(q(t)) = \frac{\partial \mathbf{e}_1}{\partial q}$  and  $\mathbf{J}_2 = \mathbf{J}_2(q(t)) = \frac{\partial \mathbf{e}_2}{\partial q}$  are invertible. Derivating  $\mathbf{e}_1(q(t))$  and  $\mathbf{e}_2(q(t))$  with respect to time leads to the following natural result:

$$\dot{\mathbf{e}}_1 = \dot{\mathbf{e}}_1(q(t)) = \mathbf{J}_1 \dot{q} \quad \text{and} \quad \dot{\mathbf{e}}_2 = \dot{\mathbf{e}}_2(q(t)) = \mathbf{J}_2 \dot{q}. \tag{20}$$

Now, we assume that the goal consists of executing first  $\mathbf{e}_1$  and then  $\mathbf{e}_2$ . To this aim, we design a kinematic controller  $\dot{q}$  so that  $\mathbf{e}_1$  is regulated to zero. Using (20) and recalling that  $\mathbf{J}_1$  is invertible, the desired controller is given by:

$$\dot{q} = \mathbf{J}_1^{-1} \dot{\mathbf{e}}_1^*, \tag{21}$$

where  $\dot{\mathbf{e}}_1^* = \dot{\mathbf{e}}_1^*(q(t))$  is the chosen dynamics to make  $\mathbf{e}_1$  vanish. Following the same reasoning, we can also design a controller to perform the second task. We obtain:

$$\dot{q} = \mathbf{J}_2^{-1} \dot{\mathbf{e}}_2^*, \tag{22}$$

where  $\dot{\mathbf{e}}_2^* = \dot{\mathbf{e}}_2^*(q(t))$  is the dynamics imposed to regulate  $\mathbf{e}_2$  to zero.

Thus, executing first  $\mathbf{e}_1$  and then  $\mathbf{e}_2$  requires us to switch continuously between controllers (21) and (22). As the dynamics of  $\mathbf{e}_1$  has been already fixed to realize the first task the best way, the only parameter that can be tuned to ensure the global control law smoothness is the dynamics of  $\mathbf{e}_2$ , namely  $\dot{\mathbf{e}}_2^*$ . The key idea consists then in choosing  $\dot{\mathbf{e}}_2^*$  so that the continuity is preserved at the switching instant  $t = t_s$ . To this aim, it is necessary to evaluate the effect of the execution of task  $\mathbf{e}_1$  on the dynamics of  $\mathbf{e}_2$  at  $t = t_s$ . Using (20) and taking into account that  $\dot{q}$  is given by relation (21) when the first task is performed, we obtain:

$$\dot{\mathbf{e}}_2(q(t)) = \mathbf{J}_2 \dot{q} = \mathbf{J}_2 \mathbf{J}_1^{-1} \dot{\mathbf{e}}_1^*(q(t)). \tag{23}$$

Thus, the global control law continuity will be insured if  $\dot{\mathbf{e}}_2^*$  is chosen so that relation (23) is verified at  $t = t_s$ . Finally, the global control law is given by:

$$\dot{q}(t) = \begin{cases} \mathbf{J}_1^{-1} \dot{\mathbf{e}}_1^*(q(t)) & \forall t \leq t_s \\ \mathbf{J}_2^{-1} \dot{\mathbf{e}}_2^*(q(t)) & \forall t \geq t_s, \end{cases} \tag{24}$$

where  $\dot{\mathbf{e}}_2^*(q(t))$  is computed so that its value at the switching instant is given by relation (23) expressed at  $t = t_s$ . Therefore, this relation provides the condition that must be verified by the second task dynamics for the controller to be continuous. More details about the theoretical foundations of this result can be found in Ref. [12].

### 3.2.2. Chosen Dynamics

Now, it remains to impose a suitable dynamics for  $\mathbf{e}_2$ . Following the previous reasoning, the dynamics  $\dot{\mathbf{e}}_2^*$  must be chosen so that:

- (i) The task function  $\mathbf{e}_2$  is regulated to zero, which allows us to successfully perform the mission.
- (ii) The value of  $\dot{\mathbf{e}}_2^*(q(t_s))$  can be imposed to verify (23) at  $t = t_s$ , guaranteeing the control law smoothness at the switching time.

Therefore, a simple classical exponential decay does not fulfill the second requirement as it does not allow us to constrain the value of  $\dot{\mathbf{e}}_2^*$  at  $t = t_s$ . Another solution, suggested by Souères and Cadenat in Ref. [12], is to use a second-order linear dynamics:

$$\ddot{\mathbf{e}}_2 + k_1 \dot{\mathbf{e}}_2 + k_2 \mathbf{e}_2 = 0, \quad (25)$$

where the two parameters  $k_1 > 0$  and  $k_2 > 0$  are used to control both the error decreasing speed and the duration of the transient time response. Although this dynamics satisfies the two previous objectives, it may be difficult to choose the values of these parameters to obtain a convenient behavior. This is the reason why Mansard and Chaumette propose to use a nonhomogeneous first-order dynamics [14]:

$$\dot{\mathbf{e}}_2^*(q(t)) = -\lambda \mathbf{e}_2(q(t)) + \rho(t), \quad (26)$$

where  $\rho(t) = \kappa e^{-\tau(t-t_s)}$  with  $\kappa$  given by:

$$\kappa = \dot{\mathbf{e}}_2(q(t_s)) + \lambda \mathbf{e}_2(q(t_s)) = \mathbf{J}_2(q(t_s)) \mathbf{J}_1^{-1}(q(t_s)) \dot{\mathbf{e}}_1^*(q(t_s)) + \lambda \mathbf{e}_2(q(t_s)), \quad (27)$$

$\lambda > 0$  and  $\tau > 0$  represent two gains which allow us to set independently the error decreasing speed and the transient time duration (in order to ensure a short transient time response with respect to the decreasing time of the task error, it suffices to choose  $\tau > \lambda$  [14]). Note that the above dynamics is equivalent to the one designed by Souères and Cadenat [12] if  $k_1 = \lambda + \tau$  and  $k_2 = \lambda \tau$ . Therefore, the two approaches are equivalent, except that the latter offers the advantage of decoupling the error decreasing speed from the transient time duration.

Now, it remains to verify that the proposed dynamics (26) allows us to satisfy the two previously mentioned requirements. To this aim, let us analyze its behavior at the switching time  $t_s$ . Two cases may occur:

- At  $t = t_s$ : it is straightforward to show that the necessary initial condition on  $\dot{\mathbf{e}}_2^*$  is obtained at this instant, because relation (26) can be rewritten as  $\dot{\mathbf{e}}_2^*(q(t_s)) = \mathbf{J}_2(q(t_s)) \mathbf{J}_1^{-1}(q(t_s)) \dot{\mathbf{e}}_1^*(q(t_s))$ .

- At  $t \gg t_s$ :  $\rho(t) \rightarrow 0 \Rightarrow \dot{\mathbf{e}}_2(q(t), t) = -\lambda \mathbf{e}_2(q(t), t)$ . The exponential decay that is classically imposed in the visual servoing literature can then be retrieved after the switch. The task function is regulated to zero and the mission can be successfully performed.

Hence, the proposed dynamics is suitable for our purpose.

### 3.2.3. Application to Our Case

Here, our objective is to show how to apply the above results to our specific case and deduce a convenient control law. Contrary to the convex combination case where the dangers of collision and occlusion were evaluated through continuous parameters of  $d_{\text{coll}}$  and  $d_{\text{occ}}$ , the strategy proposed below relies only on binary flags that are enabled or disabled depending on the values of these two distances. Therefore, we still consider the same envelopes and distances as described in Figs 2 and 3b. Before detailing the chosen control strategy, we first introduced two flags COLL and OCC that are, respectively, enabled when there is a risk of collision ( $d_{\text{coll}} \leq d_-$ ) and of occlusion ( $d_{\text{occ}} \leq D_-$ ). Note that these two events must be anticipated for three main reasons: (i) it is necessary to guarantee robot safety and visual features visibility for the task to be successfully performed, (ii) as the proposed controllers aim at avoiding collisions and occlusions, they will be efficient if they are used before the corresponding problems occur, and (iii) dynamical sequencing requires us to compute the necessary initial conditions at the switching instant. Now, let us describe the control strategy. Several cases may occur:

- If there is no risk of occlusion nor collision (i.e.,  $d_{\text{occ}} > D_-$  and  $d_{\text{coll}} > d_-$ ) the flags OCC and COLL are set to 0, and a controller making  $\mathbf{e}_{\text{vs}}$  vanish must be applied to the vehicle. It is then necessary to modify the control law so that the vision-based task is performed. At this instant (which corresponds to the switching time  $t_s$ ), we compute the initial values  $\mathbf{e}_{\text{vs}}(q(t_s))$  and  $\dot{\mathbf{e}}_{\text{vs}}(q(t_s))$  that are required to compute a convenient dynamics (see (26)). The first one will be directly evaluated from the visual data provided by the camera thanks to (3), while the second one will be determined using (23). Now, following the above dynamical sequencing formalism, a controller able to make  $\mathbf{e}_{\text{vs}}$  decrease while guaranteeing the continuity is given by:

$$\dot{q} = \mathbf{J}_{\text{vs}}^{-1}(q(t)) \dot{\mathbf{e}}_{\text{vs}}^*(q(t)), \quad (28)$$

where  $\dot{\mathbf{e}}_{\text{vs}}^*$  is given by (26). This controller will be applied while OCC and COLL remain disabled.

- If  $d_{\text{occ}} \leq D_-$ , a risk of occlusion is detected and OCC is enabled. In this case, it is necessary to change the control and switch to another controller regulating  $\mathbf{e}_{\text{oa}}$  to zero. Following the same reasoning, in order to ensure the control law smoothness at the switching time  $t_s$ , we determine the requested initial condi-



tions  $\mathbf{e}_{\text{oa}}(q(t_s))$  and  $\dot{\mathbf{e}}_{\text{oa}}(q(t_s))$  using, respectively, (5) and (23). As previously, a suitable controller is expressed as:

$$\dot{q} = \mathbf{J}_{\text{oa}}^{-1}(q(t))\dot{\mathbf{e}}_{\text{oa}}^*(q(t)), \quad (29)$$

where  $\dot{\mathbf{e}}_{\text{oa}}^*$  is given by (26) and  $\mathbf{J}_{\text{oa}} = \frac{\partial \mathbf{e}_{\text{oa}}}{\partial q}$ . As  $\mathbf{J}_{\text{oa}}$  cannot be easily computed, we have chosen to approximate it by the identity matrix as is classically done when using the redundant task function formalism [15]. The controller will then finally express as:  $\dot{q} = \dot{\mathbf{e}}_{\text{oa}}^*(q(t))$ . This choice does not prevent the control law from being continuous, provided that the same approximation is used to compute the initial conditions and the controller. Let us also notice that OCC remains enabled while there exists an occlusion risk. Thus, the previous controller will be applied to the robot until the occluding object  $\mathcal{O}$  leaves the image ( $d_{\text{bord}} = 0$ ) or at least goes out of the critical zone ( $d_{\text{occ}} = D_0$ ). At this time, OCC is disabled, and a flag END\_OCCLU is set to 1 in order to avoid undesirable switches between the above controller and the next suitable one (see Remark 4).

- If the vehicle enters the zone surrounded by  $\Xi_-$  ( $d_{\text{coll}} \leq d_-$ ), a risk of collision occurs and the flag COLL is set to 1. It is then necessary to switch from the current controller to another one able to make  $\mathbf{e}_{\text{ca}}$  vanish. At this time, we compute the required initial conditions  $\mathbf{e}_{\text{ca}}(q(t_s))$  and  $\dot{\mathbf{e}}_{\text{ca}}(q(t_s))$  using, respectively, (11) and (23). We deduce the following controller:

$$\dot{q} = \mathbf{J}_{\text{ca}}^{-1}(q(t))\dot{\mathbf{e}}_{\text{ca}}^*(q(t)), \quad (30)$$

where  $\dot{\mathbf{e}}_{\text{ca}}^*$  is given by (26) and  $\mathbf{J}_{\text{ca}} = \frac{\partial \mathbf{e}_{\text{ca}}}{\partial q}$  is the Jacobian matrix of task function  $\mathbf{e}_{\text{ca}}$ . As previously, this matrix cannot be easily computed and, following the same reasoning as for the occlusion avoidance controller, we approximate  $\mathbf{J}_{\text{ca}}$  by the identity matrix [15]. Therefore, in this case,  $\dot{q} = \dot{\mathbf{e}}_{\text{ca}}^*(q(t))$ . In order to guarantee robot safety, the flag COLL is maintained to 1 until the obstacle does not induce any collision danger. This event occurs when the directions of the pan platform and of the mobile base become identical (i.e., when  $\theta = \vartheta$ ). A flag END\_COLL is set to 1 (see Remark 4), while COLL is disabled. Therefore, the above controller drives the robot while COLL = 1 and END\_COLL = 0, ensuring its safety. Note that  $\mathbf{e}_{\text{ca}}$  is considered to be the most priority task — its execution cannot be interrupted, except when END\_COLL is enabled. Therefore, while COLL = 1, only the collision avoidance controller can be activated.

**Remark 4.** Thanks to flags END\_OCCLU and END\_COLL, we prevent the control law from unnecessarily switching from one controller to the other. For instance, this problem may arise at the end of a collision (respectively, occlusion) avoidance phase. Indeed, in such cases, the robot (respectively, the image features) may enter again in the dangerous zone, whereas the collision (respectively, occlusion) cannot occur anymore. In this way, we avoid some chattering problems and we ensure that only the most suitable controller is applied to the vehicle. The problems of

local minima are then strongly reduced, contrary to approaches relying on convex combination, which remain much more sensitive to them.

Finally, the control law is defined by:

$$\dot{q}(t) = \begin{cases} \mathbf{J}_{vs}^{-1}(q(t))\dot{\mathbf{e}}_{vs}^*(q(t)) & \text{if OCC} = 0 \text{ and COLL} = 0 \\ \mathbf{J}_{oa}^{-1}(q(t))\dot{\mathbf{e}}_{oa}^*(q(t)) \simeq \dot{\mathbf{e}}_{oa}^*(q(t)) & \text{if OCC} = 1 \text{ and COLL} = 0 \\ & \text{and END\_OCCLU} = 0 \\ \mathbf{J}_{ca}^{-1}(q(t))\dot{\mathbf{e}}_{ca}^*(q(t)) \simeq \dot{\mathbf{e}}_{ca}^*(q(t)) & \text{if COLL} = 1 \\ & \text{and END\_COLL} = 0, \end{cases} \quad (31)$$

where  $\dot{\mathbf{e}}_{vs}^*$ ,  $\dot{\mathbf{e}}_{oa}^*$  and  $\dot{\mathbf{e}}_{ca}^*$  are defined so that to satisfy dynamics (26), that is:

$$\begin{cases} \dot{\mathbf{e}}_{vs}^* = -\lambda_{vs}\mathbf{e}_{vs}(q(t)) + \kappa_{vs}e^{-\tau_{vs}(t-t_s)}, \\ \text{with } \kappa_{vs} = \dot{\mathbf{e}}_{vs}(q(t_s)) + \lambda_{vs}\mathbf{e}_{vs}(q(t_s)) \\ \dot{\mathbf{e}}_{oa}^* = -\lambda_{oa}\mathbf{e}_{oa}(q(t)) + \kappa_{oa}e^{-\tau_{oa}(t-t_s)}, \\ \text{with } \kappa_{oa} = \dot{\mathbf{e}}_{oa}(q(t_s)) + \lambda_{oa}\mathbf{e}_{oa}(q(t_s)) \\ \dot{\mathbf{e}}_{ca}^* = -\lambda_{ca}\mathbf{e}_{ca}(q(t)) + \kappa_{ca}e^{-\tau_{ca}(t-t_s)}, \\ \text{with } \kappa_{ca} = \dot{\mathbf{e}}_{ca}(q(t_s)) + \lambda_{ca}\mathbf{e}_{ca}(q(t_s)). \end{cases} \quad (32)$$

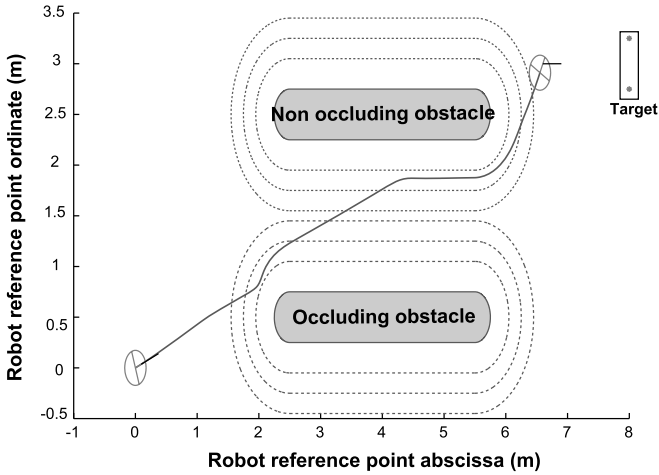
The different gains  $\lambda_{xx} > 0$  and  $\tau_{xx} > 0$  are chosen to fix the error decreasing speed and the transient time duration of each subtask. Note that, contrary to (24) and to most of the related works presented in the literature, the switch between the different controllers is performed on the base of significant events and not on the base of time. This means that the switching time  $t_s$  is not *a priori* known, but determined depending on the execution of the mission. Hence,  $t_s$  is computed at each detection of an relevant event.

## 4. Results and Discussion

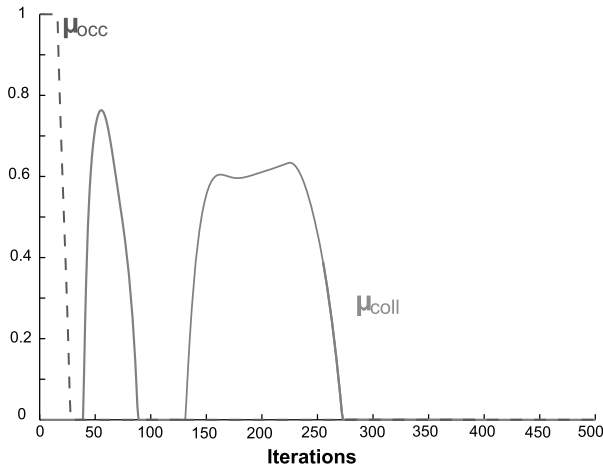
### 4.1. Simulation Results

The above-mentioned sequencing approaches have been validated using MATLAB software. Numerous tests have been performed. We first detail two results allowing us to discuss the efficiency of each approach. For both tests, the scene has been cluttered with two obstacles that may occlude the camera or represent a danger for the mobile base. The security envelopes  $D_-$ ,  $D_0$  and  $D_+$  have been, respectively, fixed to 50, 100 and 150 pixels, and  $d_-$ ,  $d_0$ ,  $d_+$  to 0.3, 0.5 and 0.7 m. The control law sampling period has been set to 100 ms, which corresponds to the one used on our real robots. We kept the same values for all the presented simulation results. Finally, let us note that the same conditions (robot initial configuration, obstacles location, gains, etc.) have been chosen for each test in order to ease the study and the comparison of the proposed techniques.

We consider the first simulation case. Here, our goal is to validate both approaches and to propose a first analysis of their respective behavior. The obtained



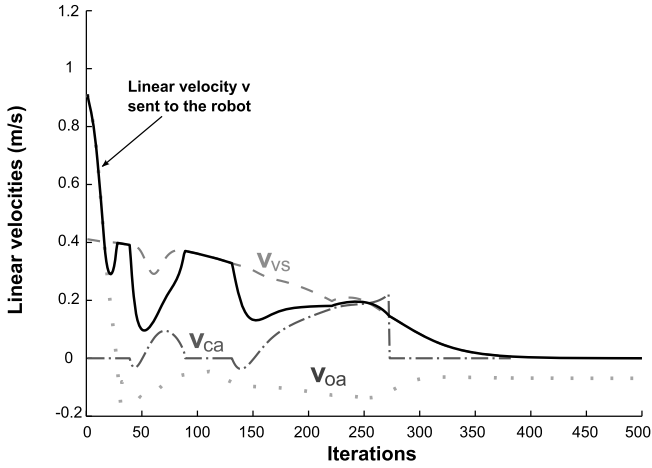
(a)



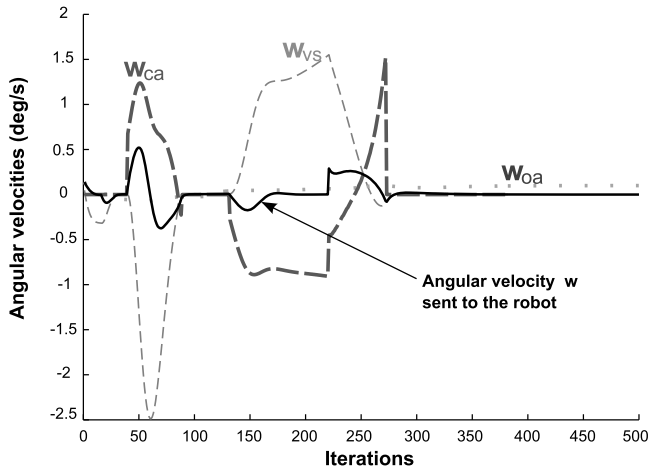
(b)

**Figure 4.** First test: convex combination strategy. (a) Robot trajectory. (b) Collision and occlusion risks. (c) Linear velocity (m/s). (d) Angular velocities (deg/s).

results for the convex combination approach are shown in Fig. 4. As one can see, the mission is successfully performed despite the obstacles. At the beginning, the robot is positioned out of the danger zone, but its camera is orientated so that an occlusion risk occurs. As  $\mu_{occ} = 1$  and  $\mu_{coll} = 0$ , only  $\dot{q}_{oa}$  is sent to the robot, which goes towards the target while avoiding the occlusion. Thus,  $\mu_{occ}$  decreases to vanish after about 30 iterations and  $\dot{q}_{vs}$  progressively replaces  $\dot{q}_{oa}$  (Fig. 4c and d). However, the mobile base enters the first obstacle vicinity and  $\mu_{coll}$  increases. During this phase, a combination of  $\dot{q}_{vs}$  and  $\dot{q}_{ca}$  is used as shown in Fig. 4c and d. As this combination is sufficient to make the robot leave the dangerous zone,  $\mu_{coll}$  progressively decreases to 0 without reaching 1. Once again, only  $\dot{q}_{vs}$  is sent to the vehicle. The second ob-



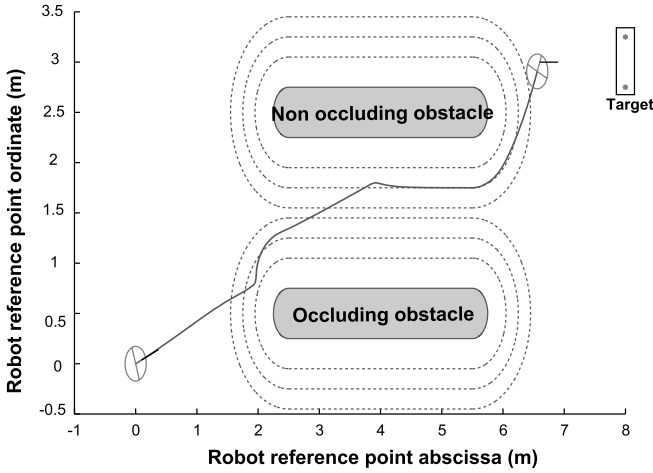
(c)



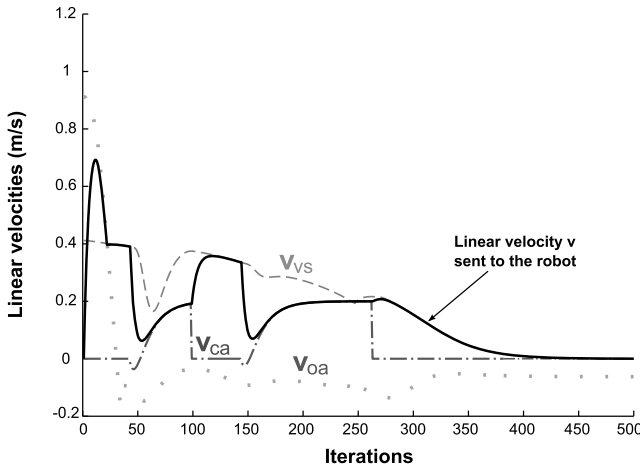
(d)

**Figure 4.** (Continued.)

stacle is then encountered. As the camera is correctly orientated towards the target, only a collision risk occurs and  $\mu_{coll}$  rises again. A combination of the controllers is applied anew and, as previously, suffices to make the robot leave the danger region. Note that, during both avoidance phases, the robot does not exactly follow the security envelope  $\xi_0$  as requested because  $\dot{q}_{ca}$  is never applied alone. When there is no more collision nor occlusion risk, the robot converges towards the target using the sole visual servoing controller. The control inputs are presented on Fig. 4c and d, and appear to be smooth during the whole mission, validating the proposed sequencing technique. We have plotted all the computed velocities to highlight the way the different switches are performed. As one can see, while  $\mu_{occ}$  and  $\mu_{coll}$  remain equal to 0 (or 1), a sole controller is applied to the robot. However, when they



(a)



(b)

**Figure 5.** First test: dynamical sequencing: linear and angular velocities. (a) The robot trajectory. (b) Linear velocity (m/s). (c) Angular velocities (deg/s).

vary, the global control law takes into account several opposite controllers, which may lead to some undesired behaviors if the most suitable velocity is not selected sufficiently rapidly.

Now, we consider the obtained results for dynamical sequencing presented in Fig. 5. The task is once again performed perfectly. As the reasoning is quite similar to the one described above, we will only focus on the differences between the two approaches. First, let us compare Figs 5a and 4a. At first glance, both trajectories look like each other; however, we can observe that, during the avoidance phases, the robot is more rapidly brought back towards the security envelope  $\xi_0$  for dynamical sequencing. The vehicle even follows it when avoiding the second encountered

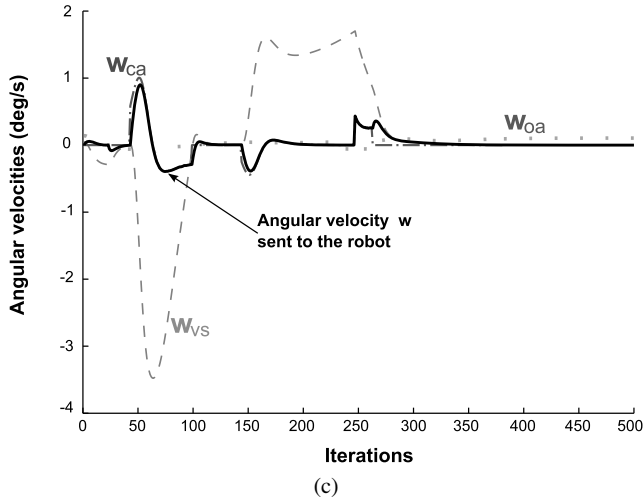
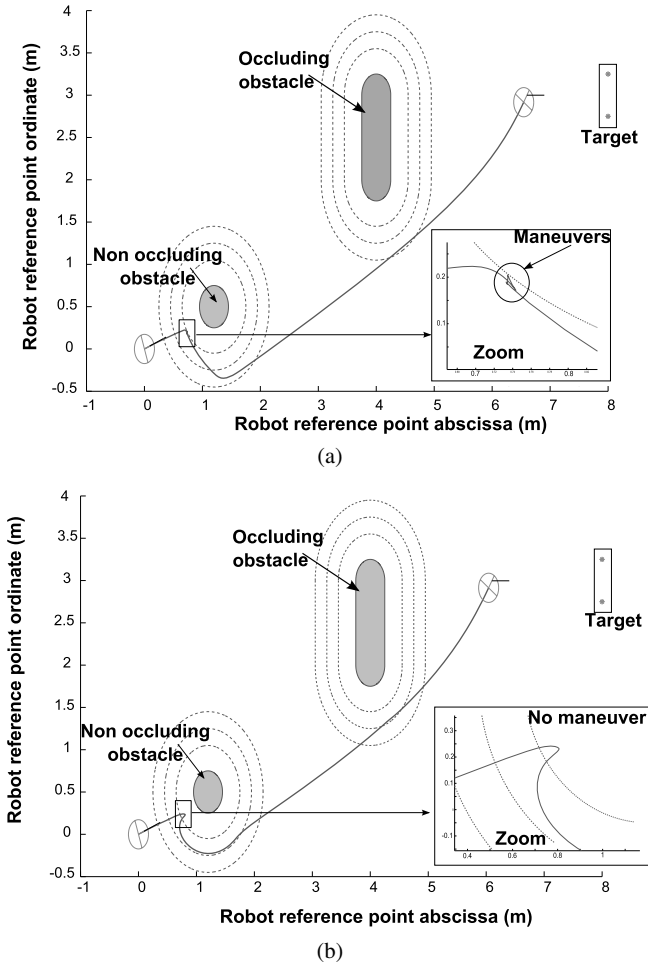


Figure 5. (Continued.)

obstacle (the result is slightly different for the first obstacle, because the leaving condition occurs before the robot has been settled on  $\xi_0$ ). This result is rather logical. Indeed, as only the most relevant controller (and not a combination of opposite ones) is applied to the robot, the avoidance motion quality is significantly improved. The same reasoning holds when treating the occlusions. This particular characteristic is the main advantage of dynamical sequencing over convex combination. Let us now analyze the control input evolutions shown in Fig. 5b and c. As one can see, they remain smooth, which validates the proposed sequencing approach. Furthermore, as mentioned before, we can observe that only the most suitable controller (and not several ones) is rapidly applied to the vehicle, limiting undesired behaviors. Note that the transient time duration of a switch is fixed by the sole parameter  $\tau_{xx}$ , making the implementation easier.

Now, we consider the second simulation test. Here, we clearly demonstrate the limitations of convex combination approaches with respect to dynamical sequencing. The results are presented in Fig. 6. Figure 6a shows that, for convex combination, the robot has to perform maneuvers to avoid both collisions and occlusions, which occur in the first obstacle vicinity. Indeed, during this phase, as  $\mu_{\text{coll}}$  never reaches 1 (Fig. 6c), three opposite controllers simultaneously act on the vehicle, making it go towards the obstacle and then backward. In fact, the robot is close to a local minimum. Fortunately, the maneuvers suffice to make the robot leave this zone and the task can be finally performed. On the contrary, this problem does not occur when using dynamical sequencing as shown by Fig. 6b. Indeed, as the most relevant controller is always applied, there is no risk of local minimum and maneuvers are no longer required. Once again, we observe that the robot accurately follows the security envelope, which demonstrates that the quality of the avoidance motion is improved. The mission can then be performed more efficiently thanks to this last approach.

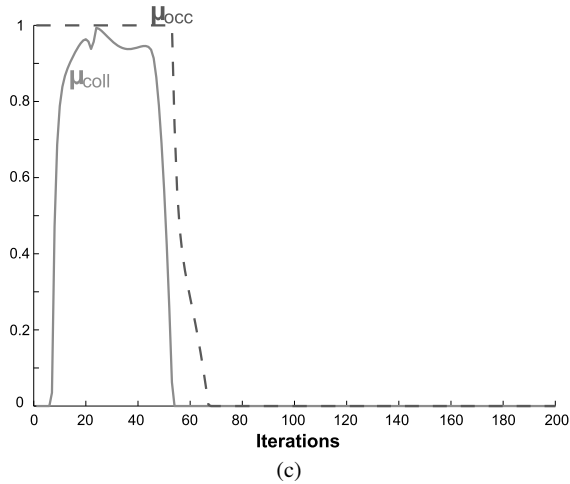


**Figure 6.** Second test: results for dynamical sequencing and convex combination. (a) Convex combination. (b) Dynamical sequencing. (c) Risks of collision and occlusion (convex combination).

To demonstrate the efficiency of the proposed sequencing techniques, we present one additional result where the environment is highly cluttered. As one can see from Fig. 7, the task is correctly performed with both approaches, despite several occluding and non-occluding obstacles. We do not detail the reasoning as it is quite similar to the previous cases.

4.2. Discussion

Now, we discuss the advantages and drawbacks of two sequencing techniques from both theoretical and practical points of view. The convex combination approach relies on a simple reasoning and allows us to independently design the three basic controllers using any convenient dynamics. Applications can then be rapidly carried out. This characteristic is its main advantage. However, it suffers from three

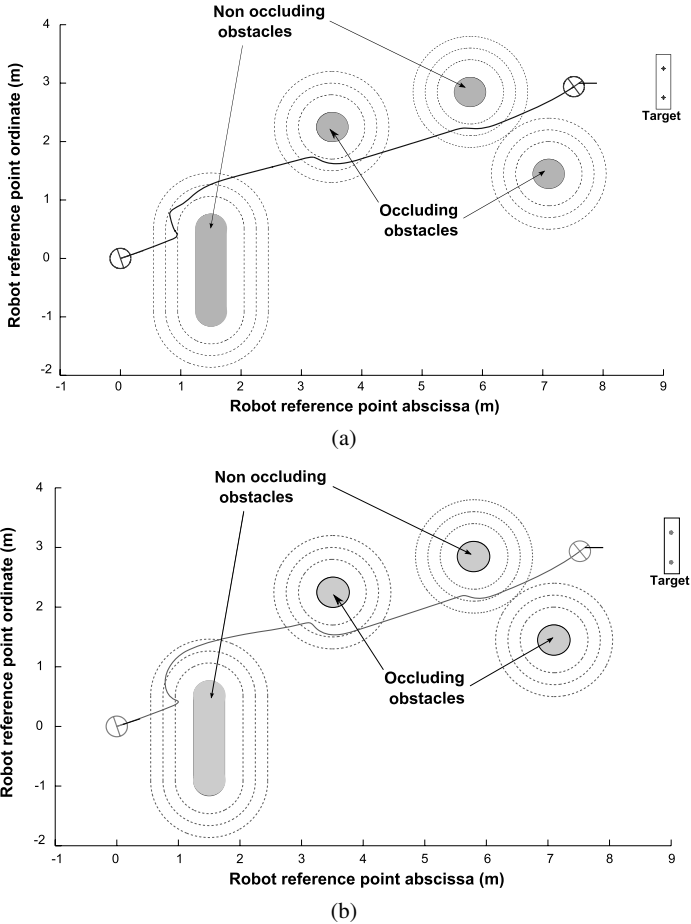


**Figure 6.** (Continued.)

main drawbacks. (i) It is usually difficult to guarantee the task feasibility because of local minima that may occur when different opposite controllers act on the robot at the same time (see Remark 3). (ii) The sequencing quality depends on numerous parameters (three control gains, three flags, two switching functions, six distances) that are strongly coupled: the three flags and the two switching functions depend on the six distances that have been defined to handle occlusions ( $D_-$ ,  $D_0$ ,  $D_+$ ) and collisions ( $d_-$ ,  $d_0$ ,  $d_+$ ). (iii) All of them influence both the switching time and the way the control law will vary when switching. Thus, it is very difficult to *a priori* set these parameters and evaluate how the control will behave when modifying one (or several) of them. Finally, simulations have shown that their values are closely related to the task to be realized and to the environment (obstacles and goal locations, robot initial configuration, etc.). Some technical expertise is then required to implement the strategy.

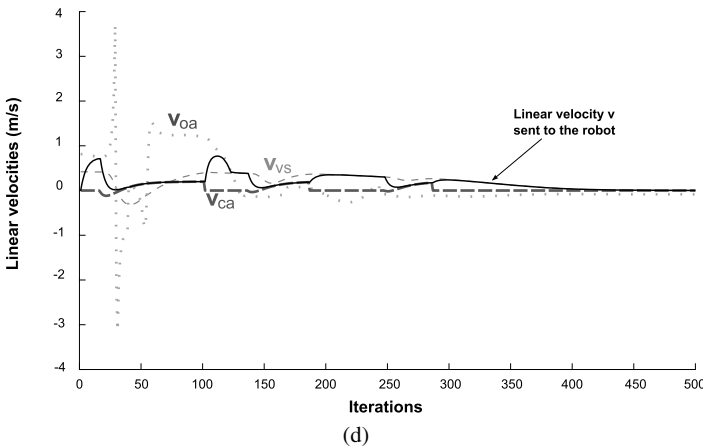
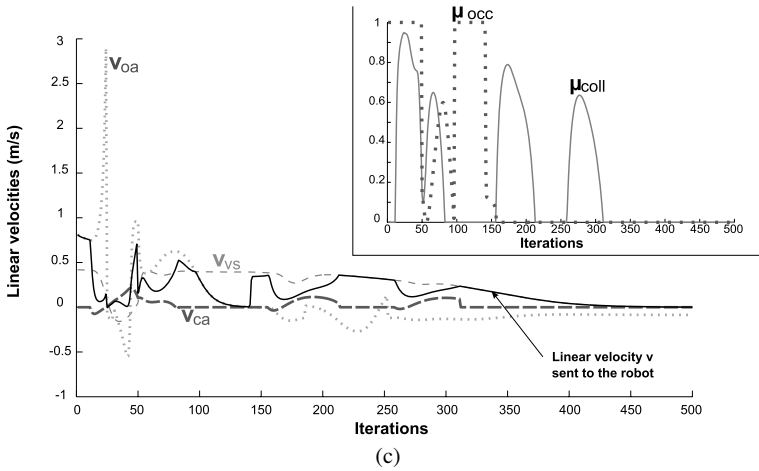
Dynamical sequencing relies on a more complex mathematical theory than convex combination. However, this approach presents several strong advantages with respect to the previous one. (i) It allows us to naturally guarantee the control law smoothness at the switching time. The continuity is only ensured at the first-order, which requires a careful choice of the gains  $\lambda_{xx}$  and  $\tau_{xx}$  to avoid some saturation problems at the acceleration level. (ii) The implementation is significantly simplified for the following reasons. First of all, the proposed control strategy relies on the definition of simple flags allowing us to detect the events that influence the task execution. If they are properly defined, only the most suitable controller (and not a combination of opposite control laws as previously) is applied to the robot. The problem of local minima is then noticeably reduced. Furthermore, few parameters (two gains and four flags depending on three distances) are involved in the control strategy. These parameters are mathematically decoupled; indeed, the four flags define the time  $t_s$  when the control has to switch and the two gains the way the





**Figure 7.** Additional test: robot trajectories, linear velocities, collision and occlusion risks. (a) Convex combination. (b) Dynamical sequencing. (c) Convex combination. (d) Dynamical sequencing.

control law will vary at  $t_s$ . Thus, contrary to the first sequencing technique, the distances between the envelopes only have an effect on  $t_s$  but does not influence the switching dynamics. Nonetheless, if these parameters remain independent from a mathematical point of view, it is necessary to choose them consistently to obtain nice behaviors. To do so, it suffices to fix the switching dynamics depending on the robot abilities (velocities and acceleration maximal bounds, etc.) and then to define some adequate distances allowing us to respect it. Therefore, a non-specialist user is able to *a priori* select suitable parameters, whereas a solid technical expertise is required in the first case. These aspects are the key advantages of dynamical sequencing techniques with respect to convex combination approaches. A summary is proposed in Table 2.



**Figure 7.** (Continued.)

## 5. Conclusions

In this paper, we have presented a multi-sensor-based control strategy allowing us to visually drive a robot towards a landmark while avoiding visual feature occlusions and obstacle collisions. The proposed method relies on the switch between three different controllers, depending on the environment. In order to guarantee the control law smoothness, two sequencing techniques have been proposed. The first one consists of merging the controllers thanks to a convex combination, while the second one allows us to naturally guarantee the continuity by suitably choosing the initial conditions at the switching time. These two approaches have been tested using MATLAB software. The obtained results have validated the key idea of this work: designing several separate controllers and sequencing them at the right instant to efficiently achieve the mission. The advantages and drawbacks of both techniques have also been highlighted and the proposed comparative analy-

**Table 2.**

Comparative analysis summary

Sequencing technique	Advantages	Drawbacks
Convex combination approach	based on a simple reasoning acceleration constraints can be easily treated	high risk of local minima requires to choose numerous coupled parameters that depend on the task, on the robot and on the environment requires a solid technical expertise to fix the different parameters
Dynamical sequencing approach	natural control law smoothness local minima problems are nearly suppressed requires few independent parameters that can be easily <i>a priori</i> chosen, on the basis of the robot abilities once implemented, the technique can be easily used by a non-specialist user	relies on a more complex mathematical theory sudden variations in the acceleration may occur

sis has clearly shown the interest of using dynamical sequencing instead of convex combination to perform complex navigation tasks.

For future work, we first plan to experiment these approaches on our mobile robot, as the obtained simulation results are quite satisfactory. From a more theoretical point of view, different research axes may be considered. First, it would be interesting to automatize the choice of parameters  $\lambda_{xx}$ ,  $\tau_{xx}$  and  $\mu_{xx}$  to avoid undesirable accelerations values while keeping suitable decreasing speed and transient time duration. We would also like to test other kinds of possible ‘switching dynamics’ and compare them to those proposed in this work. Finally, using these techniques in other contexts to perform other kinds of missions or for different types of robots appear to us as a great and interesting challenge.

## References

1. F. Chaumette and S. Hutchinson, Visual servo control, part I: basic approaches, *IEEE Robotics Automat. Mag.* **13**, 82–90 (2006).
2. E. Marchand and G. Hager, Dynamic sensor planning in visual servoing, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, Leuven, pp. 1988–1993 (1998).
3. N. Mansard and F. Chaumette, A new redundancy formalism for avoidance in visual servoing, in: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Edmonton, AB, pp. 1694–1700 (2005).
4. Y. Mezouar and F. Chaumette, Avoiding self-occlusions and preserving visibility by path planning in the image, *Robotics Autonomous Syst.* **41**, 77–87 (2002).
5. P. Corke and S. Hutchinson, A new partitioned approach to image-based visual servo control, *IEEE Trans. Robotics Automat.* **17**, 507–515 (2001).

6. V. Kyrki, D. Kragic and H. Christensen, New shortest-path approaches to visual servoing, in: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Sendai, pp. 349–355 (2004).
7. S. Benhimane and E. Malis, Vision-based control with respect to planar and non-planar objects using a zooming camera, in: *Proc. IEEE Int. Conf. on Autonomous Robots*, Coimbra, pp. 991–996 (2003).
8. A. Remazeilles, N. Mansard and F. Chaumette, Qualitative visual servoing: application to the visibility constraint, in: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Beijing, pp. 4297–4303 (2006).
9. V. Cadenat, P. Souères and M. Courdresses, Using system redundancy to perform a sensor-based navigation task amidst obstacles, *Int. J. Robotics Automat.* **16**, 61–73 (2001).
10. D. Folio and V. Cadenat, A controller to avoid both occlusions and obstacles during a vision-based navigation task in a cluttered environment, in: *Proc. Eur. Control Conf.*, Seville, pp. 3898–3903 (2005).
11. R. Pissard-Gibollet and P. Rives, Applying visual servoing techniques to control a mobile hand-eye system, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, Nagoya, pp. 166–171 (1995).
12. P. Souères and V. Cadenat, Dynamical sequence of multi-sensor based tasks for mobile robots navigation, in: *Proc. IFAC Symp. on Robot Control*, Wroclaw, pp. 423–428 (2003).
13. B. Gao, Contribution à la synthèse de commandes référencées vision 2D multi-critères, *PhD Dissertation*, University of Toulouse (2006).
14. N. Mansard and F. Chaumette, Task sequencing for sensor-based control, *IEEE Trans. Robotics* **23**, 60–72 (2007).
15. B. Espiau, F. Chaumette and P. Rives, A new approach to visual servoing in robotics, *IEEE Trans. Robotics Automat.* **8**, 313–326 (1992).
16. F. Chaumette, Image moments: a general and useful set of features for visual servoing, *IEEE Trans. Robotics Automat.* **20**, 713–723 (2004).
17. C. Samson, M. L. Borgne and B. Espiau, *Robot Control: The Task Function Approach*. Oxford Science, Oxford (1991).
18. N. Garcia-Aracil, J. A. Poveda, J. S. Navarro, C. P. Vidal and R. S. Pazmino, Visual control of robots with changes of visibility in image features, *IEEE Trans. Robotics Automat.* **4**, 27–33 (2006).
19. V. Cadenat, P. Souères and T. Hamel, A reactive path-following controller to guarantee obstacle avoidance during the transient phase, *Int. J. Robotics Automat.* **21**, 256–265 (2006).
20. C. Samson, Path following and time-varying feedback stabilization of a wheeled mobile robot, in: *Proc. Int. Conf. on Control Automation, Robotics and Vision*, Singapore, pp. RO-13.1.1–RO-13.1.5 (1992).
21. 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*, Leuven, pp. 1269–1274 (1998).

## About the Authors



**Viviane Cadenat** obtained her PhD, in 1999, from Paul Sabatier University, Toulouse, France. She was hired as an Associate Professor at Paul Sabatier University, in 2000. She is currently teaching robotics and automatic control theory at both the Bachelor and the Master levels. She is also Member of the RAP (Robotics, Action and Perception) group from the Laboratory for Analysis and Architecture of Systems (LAAS) and is still interested in the navigation of a mobile robot amidst obstacles using vision-based control.



**David Folio** received his PhD degree in Control Systems, in 2007, in the Robotic Action and Perception (RAP) group of the Laboratory for Analysis and Architecture of Systems (LAAS), Toulouse, France. Between 2007 and 2008, he was a Postdoctorate Fellow in the Lagadic team of the IRISA Laboratory, Rennes, France. He has worked on the design of sensor-based control strategies able to tolerate data loss. Since 2008, he has been an Assistant Professor at the Institut PRISME, Ecole Nationale Supérieure d'Ingénieurs de Bourges (ENSI Bourges), Bourges, France. He is now working on the design and modeling of navigation control strategies for micro and nanorobotics systems.



**Adrien Durand Petiteville** is PhD student in the Robotic Action and Perception (RAP) Group of the Laboratory for Analysis and Architecture of Systems (LAAS), Toulouse, France. He is working on sensor-based navigation in cluttered environments.