Using the task function approach to avoid robot joint limits and kinematic singularities in visual servoing

Éric Marchand, François Chaumette, Alessandro Rizzo

IRISA - INRIA Rennes - Université de Rennes I
Campus universitaire de Beaulieu, 35042 Rennes Cedex, France
Email {marchand, chaumett}@irisa.fr

Abstract

We propose in this paper solutions to avoid robot joint limits and kinematic singularities in visual servoing. We use a control scheme based on the task function approach. It combines the regulation of the selected vision-based task with the minimization of a secondary cost function, which reflects the manipulability of the robot in the vicinity of internal or external singularities. Several methods are proposed to avoid joint limits and a comparison between them is presented. We have demonstrated on various experiments the validity of our approach.

1 Introduction

In robotics, and in visual servoing in particular, it is important to avoid the manipulator joint limits and the kinematic singularities. Joint limits are physical limits to the extension of the operational space of the robot, and singularities are peculiar configurations where the manipulator locally looses a degree of freedom. Thus, for such situations, some motions may be impossible to realize. Dealing with visual servoing [6][7][10][5], which is a closed loop reacting to image data, planning the camera trajectory is not possible. If the control law computes a motion that exceeds a joint limit, or if the robot encounters a kinematic singularity, visual servoing generally fails. Control laws taking into account the region of space located in the vicinity of these joint limits and singularities have thus to be considered.

In order to avoid joint limits and singularities, Chang and Dubey [2] have proposed a method based on a weighted least norm solution for a redundant robot. This method does not try to maximize the distance of the joints from their limits but it dampens any motion in their direction. Thus, it avoids unnecessary self-motion and oscillations. An other approach has been proposed by Nelson and Khosla [9]. It consists in minimizing an objective function which realizes a compromise between the visual task (a target tracking) and the avoidance of internal (kinematic singularities) and external (joint limits) singularities. This function is used by exploiting the robot degrees of freedom which are redundant with respect to the visual task. During the execution of the task, the manipulator moves away from its joint limits and singularities. However, such motions can produce important perturbations in the visual servoing since they are generally not compatible with the regulation to zero of the selected image features.

In our case, we have chosen to use a control scheme based on the task function approach [11][5]. It combines the regulation of the vision-based task with the minimization of a cost function. The visual task is considered as a primary and priority task. The cost function is embedded in a secondary task whose only the components which are compatible with the primary task are taken into account (*i.e.* the minimization of the cost function is performed under the constraint that the visual task is realized). As in [9], our approach uses the robot redundancy with respect to the image constraints, and the cost function to be minimized is based on a measure of the robot manipulability. This measure must be able to compare the different manipulator configurations in order to avoid internal and external singularities. The cost function must reach its maximal value near a singularity and its gradient must be equal to zero when the cost function reaches its minimal value [11].

The next section of this paper recalls the application of the task function approach to visual servoing and the expression of the resulting control law. Section 3 describes the approach proposed to avoid internal and external singularities. We finally present real time experimental results dealing with positioning tasks (with respect to points and straight lines), as well as target tracking. These results have been obtained using an eye-in-hand system composed of a camera mounted on the end-effector of a six d.o.f. robot.

2 Visual Servoing

The *image-based visual servoing* consists in specifying a task as the regulation in the image of a set of visual features[5][7]. Embedding visual servoing in the task function approach [11] allows us to take advantage of general results helpful for the analysis and the synthesis of efficient

closed loop control schemes.

Let us denote $\underline{P}$ the set of selected visual features used in the visual servoing task. To ensure the convergence of $\underline{P}$ to its desired value $\underline{P}_d$, we need to know the interaction matrix $L_{\underline{P}}^T$ defined by the classical equation [5] :

$$\dot{\underline{P}} = L_{\underline{P}}^T(\underline{P}, \underline{p})T_c \quad (1)$$

where $\dot{\underline{P}}$ is the time variation of $\underline{P}$ due to the camera motion T_c. The parameters $\underline{p}$ involved in $L_{\underline{P}}^T$ represent the depth information between the considered objets and the camera frame.

Control laws in visual servoing are generally expressed in the operational space (*i.e.*, in the camera frame), and then computed in the articular space using the robot inverse Jacobian [6][5][10][9]. However, in order to combine a visual servoing with the avoidance of either internal and external robot singularities, it is more interesting to directly express the control law in the articular space. Indeed, manipulator joint limits and kinematic singularities are defined in this space. Furthermore, as far as kinematic singularities are concerned, it has the supplementary advantage to be able to perform a visual servoing (or any task) in a kinematic singularity as long as the task is not singular in such configurations. Thus, we first present a modified version of the control law presented in [5] in order to directly control the manipulator in the articular space.

We have to define the interaction between the motion of the considered features in the image and the robot motion. This leads to the definition of a new interaction matrix such that:

$$\dot{\underline{P}} = H_{\underline{P}}^T \, \dot{\underline{q}} \quad (2)$$

where $\underline{q} = (q_1 \cdots q_n)$ represents the manipulator position in the joint space. Since we have $T_c = J(\underline{q}) \, \dot{\underline{q}}$, where $J(\underline{q})$ is nothing but the robot Jacobian, we simply obtain:

$$H_{\underline{P}}^T(\underline{P}, \underline{p}, \underline{q}) = L_{\underline{P}}^T(\underline{P}, \underline{p}) \; J(\underline{q}) \quad (3)$$

A vision-based task $\underline{e}_1$ is defined by:

$$\underline{e}_1 = C(\underline{P} - \underline{P}_d) \quad (4)$$

where $\underline{P}_d$ is the desired value of the selected visual features, $\underline{P}$ is their current value (measured from the image at each iteration of the control law), and C, called combination matrix, has to be chosen such that $CH_{\underline{P}}^T$ is full rank along the desired trajectory $q_r(t)$. It can be defined as:

- $C = WH_{\underline{P}}^{T+}(\underline{q}, \underline{P}, \widehat{\underline{p}})$ if the 3D parameters $\underline{p}$, involved in the interaction matrix, can be estimated on-line (using for example the 3D structure estimation method presented in [3]). In that case, W is defined as a full rank matrix such that Ker W = Ker $H_{\underline{P}}^T(\underline{q}, \underline{P}, \widehat{\underline{p}})$.
- $C = WH_{\underline{P}}^{T+}(\underline{q}, \underline{P}_d, \underline{p}_d)$ if the value of the interaction matrix can not be updated at each iteration of the control law. Assumptions on the shape and on the geometry of the considered objects in the scene have thus generally to be done in order to compute the desired values $\underline{P}_d$ and $\underline{p}_d$. Such a choice allows us to avoid the on-line estimation of parameters $\underline{p}$. In that case, we set W as a full rank matrix such that Ker W = Ker $H_{\underline{P}}^{T+}(\underline{q}, \underline{P}_d, \underline{p}_d)$.

If the vision-based task does not constrain all the n robot degrees of freedom, a secondary task can be performed and we obtain the following task function:

$$\underline{e} = W^+\underline{e}_1 + (I\!I_n - W^+W)\underline{g}_s^T \quad (5)$$

where

- W^+ and $I\!I_n - W^+W$ are two projection operators which guarantee that the camera motion due to the secondary task is compatible with the regulation of $\underline{P}$ to $\underline{P}_d$. Indeed, thanks to the choice of matrix W, $I\!I_n - W^+W$ belongs to Ker $H_{\underline{P}}$, which means that the realization of the secondary task will have no effect on the vision-based task ($H_{\underline{P}}^T(I\!I_n - W^+W)\underline{g}_s^T = 0$). On the other hand, if errors are introduced in $H_{\underline{P}}^T$, $I\!I_n - W^+W$ no more exactly belongs to Ker $H_{\underline{P}}$, which will induce perturbations on the visual task due to the secondary task. Let us finally note that, if the visual task constrains all the n degrees of freedom of the manipulator, we have $W = I\!I_n$, which leads to $I\!I_n - W^+W = 0$. It is thus impossible in that case to consider any secondary task.
- $\underline{g}_s$ is the gradient of a cost function h_s to be minimized ($\underline{g}_s = \frac{\partial h_s}{\partial \underline{q}}$). This cost function is minimized under the constraint that $\underline{e}_1$ is realized.

For making $\underline{e}$ exponentially decreases and then behaves like a first order decoupled system, we get:

$$\dot{\underline{q}}_d = -\lambda\underline{e} - W^+\frac{\widehat{\partial \underline{e}_1}}{\partial t} - (I\!I_n - W^+W)\frac{\partial \underline{g}_s^T}{\partial t} \quad (6)$$

where:

- $\dot{\underline{q}}_d$ is the joint velocity given as input to the robot controller;
- λ is the proportional coefficient involved in the exponential convergence of $\underline{e}$;
- $\frac{\widehat{\partial \underline{e}_1}}{\partial t}$ represents an estimation of a possible autonomous target motion. If the target moves, this estimation has to be introduced in the control law in order to suppress tracking errors. It can be obtained using classical filtering techniques such as Kalman filter [8][1]. On the other hand, if the scene is static, we can assume that $\frac{\partial \underline{e}_1}{\partial t} = \frac{\widehat{\partial \underline{e}_1}}{\partial t} = 0$.

3 Avoiding joint limits and kinematic singularities using the task function approach

As already stated, when the vision-based task does not constrain all the six camera degrees of freedom, a secondary task can be combined with $\underline{e}_1$. Thus we can use

the redundant degrees of freedom to perform a singularities (external and internal) avoidance task. Several cost functions h_s which reflect this desired behavior are now presented.

3.1 Joint limits avoidance

Joint position specification. While performing the visual servoing task, we would like to keep the manipulator as far as possible from its joint limits. Thus, the final desired joint position $q_{i_{end}}$ of the manipulator on each axis is given by:

$$q_{i_{end}} = \frac{q_{i_{max}} + q_{i_{min}}}{2} \tag{7}$$

where $q_{i_{min}}$ and $q_{i_{max}}$ are the minimum and maximum allowable joint values for the i^{th} joint. This means that the manipulator should be located at the middle of each axis extension. Obviously, such a position generally does not correspond to a correct position of the vision-based task. However, it is possible to design a cost function which is minimal when the manipulator reaches this desired position and which is maximal in the vicinity of joint limits. The projection operator $I\!I_n - W^+W$ will make the robot moving the nearest of that position under the constraint that the visual task is realized.

The corresponding cost function h_s to be minimized is thus defined by:

$$h_s = \frac{\beta}{2} \sum_{i=1}^{n} \frac{(q_i - q_{i_{end}})^2}{q_{i_{max}} - q_{i_{min}}} \tag{8}$$

where β is a scalar constant which sets the amplitude of the control law due to the secondary task. The components of $\underline{g}_s$ and $\frac{\partial \underline{g}_s}{\partial t}$ involved in (5) and (6) are then:

$$g_{s_i} = \frac{\partial h_s}{\partial q_i} = \beta \frac{(q_i - q_{i_{end}})}{q_{i_{max}} - q_{i_{min}}}, \qquad \frac{\partial g_{s_i}}{\partial t} = 0 \tag{9}$$

We can note that values of g_{s_i} belong to the interval $[-\frac{\beta}{2}; \frac{\beta}{2}]$, which allows to easily tune β.

Joint velocities specification. An other approach consists in specifying the behavior of the manipulator in term of velocities. It introduces more constraints on the performance of the system. The final joint position is again given by $q_{i_{end}}$ (see (7)), but we also specify an exponential decrease toward this position such that:

$$\dot{q}_i^*(t) = -\alpha \frac{q_i^*(t) - q_{i_{end}}}{q_{i_{max}} - q_{i_{min}}} \tag{10}$$

where α is a constant which reflects the desired speed of the decay. The solution $q_i^*(t)$ of this differential equation is:

$$q_i^*(t) = (q_{i_0} - q_{i_{end}})\, e^{-\frac{\alpha t}{q_{i_{max}} - q_{i_{min}}}} + q_{i_{end}} \tag{11}$$

where $q_{i_0} = q_i(0)$. Thus, a new cost function is given by:

$$h_s = \frac{\beta}{2} \sum_{i=1}^{n} [q_i(t) - q_i^*(t)]^2 \tag{12}$$

g_{s_i} and $\frac{\partial g_{s_i}}{\partial t}$ are now given by:

$$g_{s_i} = \beta\,(q_i(t) - q_i^*(t)) \tag{13}$$

$$\frac{\partial g_{s_i}}{\partial t} = -\alpha\beta \frac{q_i^*(t) - q_{i_{end}}}{q_{i_{max}} - q_{i_{min}}} \tag{14}$$

Activation thresholds Reaching a position near $q_{i_{end}}$ on each axis is generally not necessary. In fact, the secondary task needs to be used only if one (or several) joint is in the vicinity of a joint limit. We thus define activation thresholds of the secondary task by $\tilde{q}_{i_{min}}$ and $\tilde{q}_{i_{max}}$ such that:

$$\begin{aligned} \tilde{q}_{i_{min}} &= q_{i_{min}} + \rho\,(q_{i_{max}} - q_{i_{min}}) \\ \tilde{q}_{i_{max}} &= q_{i_{max}} - \rho\,(q_{i_{max}} - q_{i_{min}}) \end{aligned} \tag{15}$$

where $0 < \rho < 1/2$. The secondary task function is now given by:

$$h_s = \frac{\beta}{2} \sum_{i=1}^{n} \frac{s_i^2}{q_{i_{max}} - q_{i_{min}}} \tag{16}$$

$$\text{where} \quad s_i = \begin{cases} q_i - \tilde{q}_{i_{max}} & \text{if } q_i > \tilde{q}_{i_{max}} \\ q_i - \tilde{q}_{i_{min}} & \text{if } q_i < \tilde{q}_{i_{min}} \\ 0 & \text{else} \end{cases} \tag{17}$$

and components of $\underline{g}_s$ and $\frac{\partial \underline{g}_s}{\partial t}$ take the form:

$$g_{s_i} = \begin{cases} \frac{\beta(q_i - \tilde{q}_{i_{max}})}{(q_{i_{max}} - q_{i_{min}})} & \text{if } q_i > \tilde{q}_{i_{max}} \\ \frac{\beta(q_i - \tilde{q}_{i_{min}})}{(q_{i_{max}} - q_{i_{min}})} & \text{if } q_i < \tilde{q}_{i_{min}} \\ 0 & \text{else} \end{cases}, \quad \frac{\partial g_{s_i}}{\partial t} = 0 \tag{18}$$

This last cost function to avoid joint limits is similar to the Tsai's manipulability measure used in [9]. It is however more simple since it directly sets the activation thresholds with ρ. Let us finally note that, in all cases, $\underline{g}_s$ and $\frac{\partial \underline{g}_s}{\partial t}$ are continuous, which will ensure a continuous control law.

3.2 Avoiding kinematic singularities

When the robot is in a singular configuration, it looses one or more of its degrees of freedom, which makes impossible to inverse its Jacobian (which is no more full rank). Using a control law expressed in the operational space, such as in [9], it is inevitable to avoid the robot kinematics singularities in order to correctly realize a visual servoing. In our case, as explained before, such an avoidance is not necessary if the interaction matrix H_P^T remains full rank (which is generally the case, even if the robot is in a singular configuration). However, in some cases, and especially if the vision-based task needs a number of degrees of freedom greater than the available one, it is important to be able to avoid the robot kinematic singularities. Like for the problem of joint limits avoidance, it is possible to define a corresponding cost function. When the robot is in a singularity, the determinant of its Jacobian is equal to zero. This well known property allows us to define the cost function $h_{s_{sing}}$ to be minimized as:

$$h_{s_{sing}} = \frac{1}{\det(J(\underline{q}))} \tag{19}$$

In this case, g_{s_i} and $\frac{\partial g_{s_i}}{\partial t}$ are given by:

$$g_{s_i} = -\frac{1}{\det^2(J(\underline{q}))}\frac{\partial \det(J(\underline{q}))}{\partial q_i}\ ,\ \frac{\partial g_{s_i}}{\partial t} = 0 \qquad (20)$$

Since we do not want that $\det(J(\underline{q})) = 0$ (in order to never obtain an infinite value for the cost function and its gradient), we use a saturate determinant such that:

$$\det_{sat}\left(J(\underline{q})\right) = \begin{cases} \det\left(J(\underline{q})\right) & \text{if } \left|\det\left(J(\underline{q})\right)\right| \geq \varepsilon \\ \pm\varepsilon & \text{if } \left|\det\left(J(\underline{q})\right)\right| < \varepsilon \end{cases} \qquad (21)$$

3.3 A global task function

The two tasks described above, joint limits and kinematic singularities avoidance, can be easily combined into an unique cost function as:

$$h_s = h_{s_{joint}} + K h_{s_{sing}} \qquad (22)$$

where $h_{s_{joint}}$ is one of the three cost functions dedicated to joint limits avoidance and $h_{s_{sing}}$ is the cost function dedicated to kinematic singularities avoidance. K is a constant to be tuned which allows to normalize the two tasks and to fix the relative importance of one of the sub-task with respect to the other. In [9] the combination of these two tasks is performed by a simple product ($h_s = h_{s_{joint}} h_{s_{sing}}$), thus this relative importance is impossible to determine.

4 Experimental results

The application presented in this paper has been implemented on an experimental testbed composed of a CCD camera mounted on the end effector of a six degrees of freedom cartesian robot (see Figure 1). The image processing part is performed at the video rate on a commercial image processing board (EDIXIA IA 1000). The implementation of the control law runs on a SPARC Station 10.

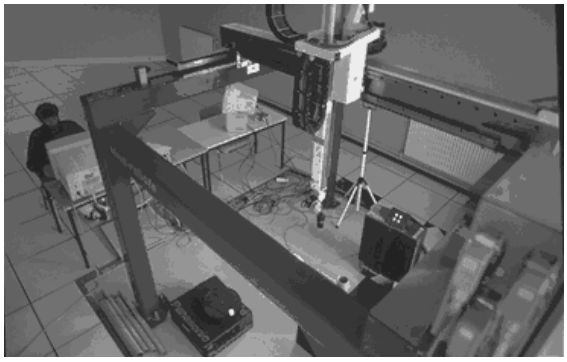


Figure 1: Experimental cell (camera mounted on a 6 dof robot)

4.1 Positioning with respect to a point

For this first experiment, we consider the case of a point. If $\underline{P} = (X, Y)^T$ describes the position of the point in the image plane and if z is its depth, the interaction matrix which links the motion of the object in the image to the camera motion T_c is given by:

$$L_{\underline{P}}^T = \begin{pmatrix} -1/z & 0 & X/z & XY & -(1+X^2) & Y \\ 0 & -1/z & Y/z & 1+Y^2 & -XY & -X \end{pmatrix} \qquad (23)$$

The goal is to observe this point at the center of the image: $\underline{P}_d = (0, 0)^T$ (see Figure 2.b).

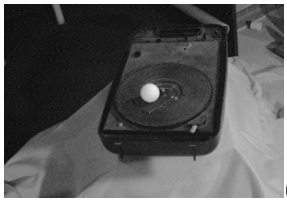

(a)

(b)

Figure 2: Positioning with respect to a point (a) initial image (b) final image

Avoiding joint limits. The initial position of the camera, corresponding to the image depicted on Figure 2.a, is located in the vicinity of three joint limits (q_1, q_3 and q_5, see Table 1). If none strategy is used to avoid joint limits, the visual task cannot be achieved. Only two degrees of freedom are necessary to perform the vision-based task, thus four motion components are redundant and can be used to realize the secondary task. Table 1 presents the obtained results according to the different strategies presented in Section 3.1. In each case, the vision task has been achieved and the joint limits have been avoided (see Figure 3, 4 and 5).

The second method (velocity specification) is more efficient than the first one (position specification), (see Figure 3.c and 4.c). For the same initial position, it allows the robot to reach faster a final position closer to the middle of the joint limits. As seen on Figure 3.a and 4a, convergence of the visual task is slow when the vision-based task is nearly achieved (we stop the experiment when $|\ \underline{P} - \underline{P}_d\ |$ is less than 1 pixel). It is due to the fact that the depth z involved in the interaction matrix has been set to 2 m while its real value was approximatively 1 m. This coarse approximation implies that the secondary task is not perfectly compatible with the visual task (due to the error in $I\!I_n - W^+W$). This introduces the observed small perturbation, which ends when the robot is far away from joint limits since the secondary task $\underline{g}_s$ then has no more influence. We can note that the secondary cost function h_s increases using the velocity specification (see Figure 4.b) while it logically decreases using the position specification (see Figure 7.b). This is due to the fact that the robot moves away from the specified trajectory $q_i^*(t)$ (see (12))

	$q_1(mm)$	$q_2(mm)$	$q_3(mm)$	$q_4(dg)$	$q_5(dg)$	$q_6(dg)$
Joint limit max $q_{i_{max}}$	750	640	500	162	141	90
Joint limit min $q_{i_{min}}$	-740	-750	-496	-171	-5	-90
Initial joint position $q_{i_{init}}$	740	-241	-495	22	3	-45
Final position (position spec.)	244	-110	-90	0	19	-41
Final position (velocity spec.)	135	-84	-35	0	22	-38
Act. Threshold Max	601	501	400	129	126	72
Act. Threshold Min	-591	-611	-396	-138	9	-72
Final position	601	-241	-396	127	8	-42

Table 1: Positioning with respect to a point: Joint limits avoidance

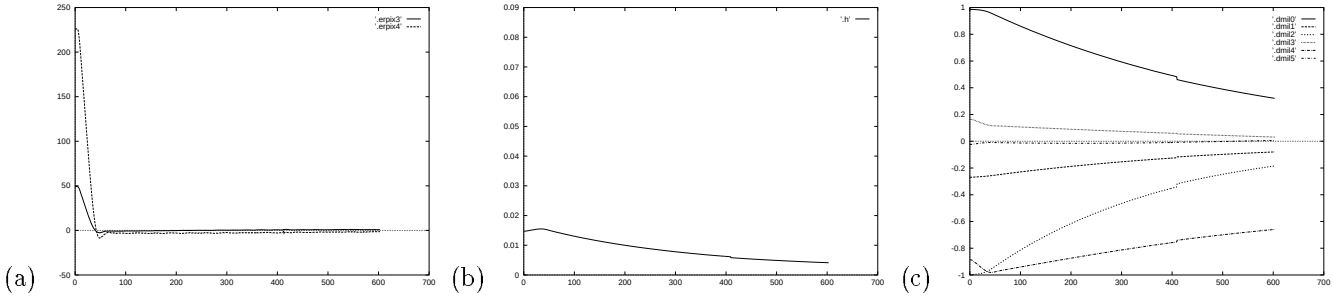


Figure 3: Positioning with respect to a point: Joint limits avoidance (position specification) (a) Error $(\underline{P} - \underline{P}_d)$ (in pixels) (b) secondary cost function h_s (c) distance from the joint limits $\frac{\underline{q} - \underline{q}_{end}}{\underline{q}_{max} - \underline{q}_{end}}$, all plots versus number of frames.

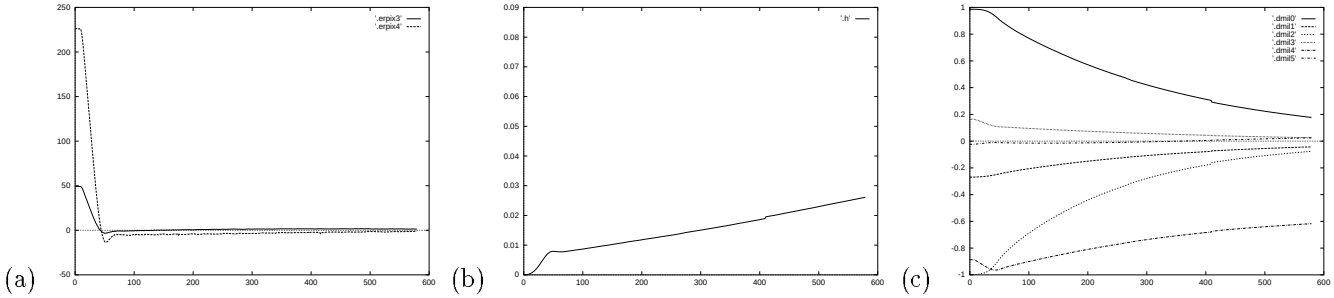


Figure 4: Same experiment using velocity specification

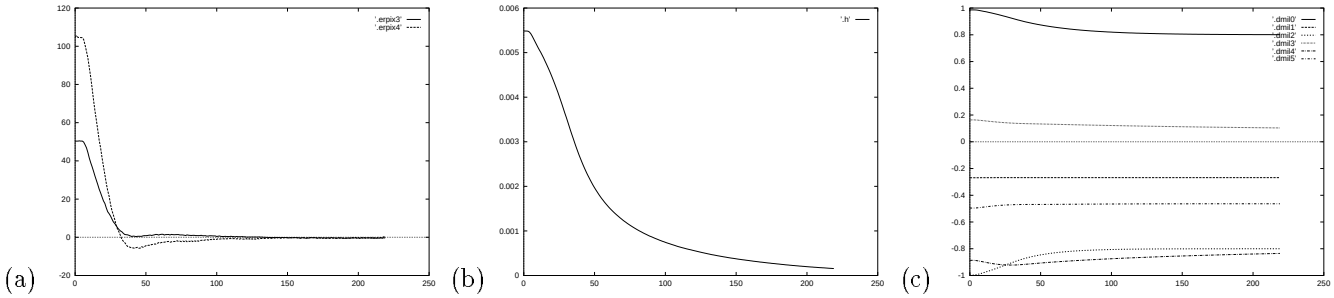


Figure 5: Same experiment using the activation thresholds

	$q_1(mm)$	$q_2(mm)$	$q_3(mm)$	$q_4(dg)$	$q_5(dg)$	$q_6(dg)$
Joint limit max $q_{i_{max}}$	750	640	500	162	141	90
Joint limit min $q_{i_{min}}$	-740	-750	-496	-171	-5	-90
Initial joint position $q_{i_{init}}$	741	634	170	-145	90	83
Act. Threshold Max	601	501	400	129	126	72
Act. Threshold Min	-591	-611	-396	-138	9	-72
Final position	601	501	171	-121	101	52

Table 2: Positioning with respect to an object: Joint limits and singularity avoidance

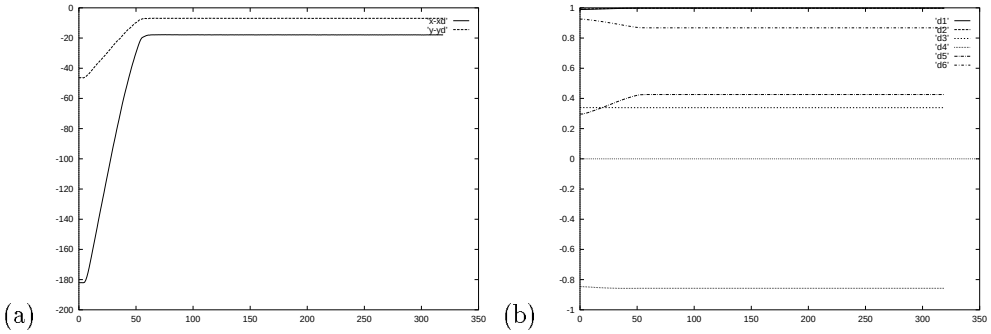


Figure 6: Positioning with respect to an objet without joint limits avoidance : (a) Error ($\underline{P} - \underline{P}_d$) (in pixels) (b) Distance from the activation thresholds limits $\frac{q_i - q_{i_{end}}}{q_{i_{max}} - q_{i_{end}}}$

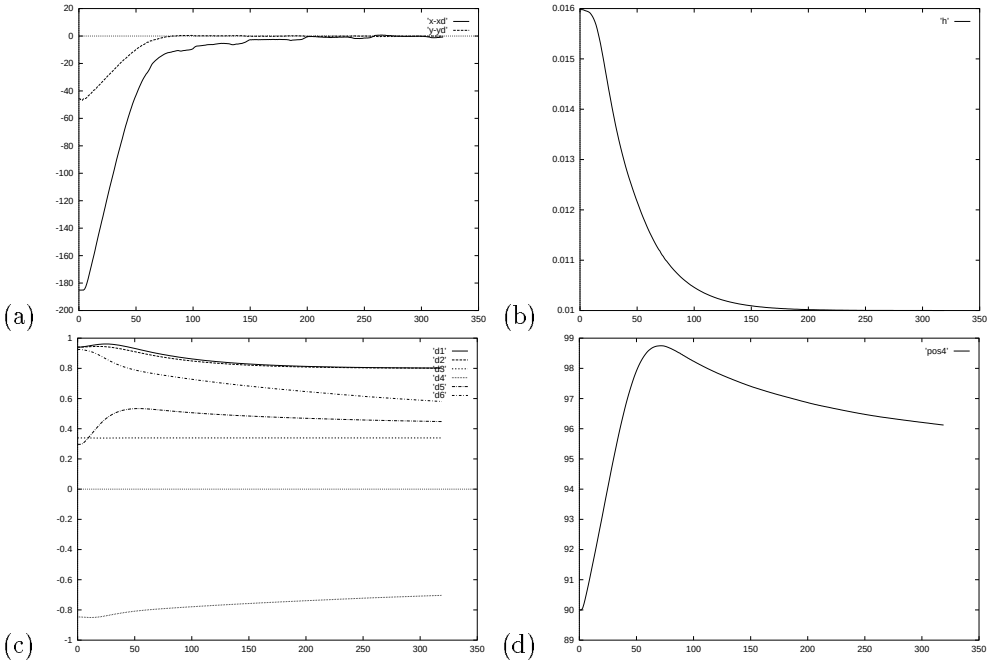


Figure 7: Positioning with respect to an objet with joint limits and singularity avoidance (a) Error ($\underline{P} - \underline{P}_d$) (in pixels) (b) secondary cost function h_s (c) Distance from the activation thresholds limits $\frac{q_i - q_{i_{end}}}{q_{i_{max}} - q_{i_{end}}}$ (d) position on q_5 (in dg), singularity is for $q_5 = 90^o$

because of the motions implied by the realization of the primary vision-based task.

The third method (activation thresholds) is the most efficient one from the point of view of the convergence speed, because only three axes are initially involved in the joint limits avoidance (none when the activation limits were reached). We used here an activation threshold equal to 10% of the distance between each axis ($\rho = 0.1$). We can see on Figure 5.c that the robot moves away from its joint limits and reaches the region limited by the activation thresholds (whose corresponding value is ± 0.8 on Figure 5.c).

Global task function: joint limits and kinematic singularities avoidance This experiment is similar to the previous one ; we have only changed the initial camera position.

The initial position of the camera, is located in the vicinity of four joint limits (q_1, q_2, q_4 and q_6) and the manipulator is in singularity $q_5 = 90\ dg$ (see Table 2). If none strategy is used to avoid joint limits, the visual task cannot be achieved. The error $\underline{P} - \underline{P}_d$ remains equal to (17,7) pixels (see Figure 6.a) since two axes reach their maximal position (see Figure 6.b). Table 2 presents the obtained results according to the strategy using the activation thresholds presented in Section 3. We can see on Figure 7.c that the robot moves away from its joint limits and reaches the region limited by the activation thresholds (± 0.8 on Figure 7.c). The fact that the robot starts in singularity does not perturb the visual servoing task which is always of full rank 2. Corresponding axis q_5 (see Figure 7.d) moves away from the singularity according to the chosen secondary task. In this experiment we have set $K = 0.0005$ and $\beta = 0.4$.

4.2 Target tracking

Let us now consider the tracking of the point used in the previous experiment. In the presented results, the target has translational motions with different amplitudes and directions. To supress tracking errors, we have used a Kalman filter with constant acceleration and colored noise state model (see [1] for details). This task is not achievable if we do not consider a peculiar strategy to avoid joint limits, since the initial robot position is near three joint limits (see Figure 8.c). The secondary cost function used here is again based on the activation thresholds approach. Once again, the joint limits avoidance is correctly performed and does not perturb the behavior of the target tracking.

4.3 Positioning with respect to a straight line

Generally, the image features which are used in visual servoing are composed with a set of image points coordinates which have to reach particular values in the image. This approach has been generalized to more complex visual features such as circle or straight line [5]. This experiment consists in positioning the camera with respect to a straight line. Its desired position is such that it appears vertical and centered in the image (see Figure 9). As in the previous experiment, only two degrees of freedom are necessary to perform the visual task. Initially, the manipulator is located near a joint limit (q_4). Without a secondary task, the robot encounters this joint limit and visual servoing fails. Figure 10 depicts the results of this experiment using the activation thresholds. Lines $Q4_{min}$ and $Q4_{max}$ on Figure 10.c represent the activation threshold: if $q_4 \geq q_{4_{max}}$ or $q_4 \leq q_{4_{min}}$, the avoidance process is activated. Using this method, the vision-based task is rapidly achieved, and the secondary task guides joint q_4 away from the joint limit region.

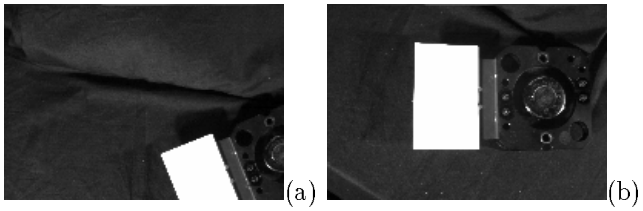

(a) (b)

Figure 9: Positioning with respect to a straight line (a) initial image (b) final image

5 Conclusion

We have proposed a solution to avoid the joint limits and kinematics singularities in visual servoing. We have chosen to use a control scheme based on the task function approach which combines the regulation of the selected vision-based task with the minimization of a secondary cost function. The secondary task reflects the manipulability of the robot in the vicinity of internal or external singularities. The third secondary task that we have proposed (using activation thresholds) is the most efficient one and allows a faster convergence of the visual task. We have demonstrated on various experiments (positioning and target tracking) the validity of our approach even when the initial robot position is near joint limits or when the robot starts in singularity. Finally let us note that, as far as kinematics singularities are concerned, if all the d.o.f of the robot are constrained by the vision-based task, an other approach based on a damped least square method can be used [4][12]. It will realize, as best as possible, a compromise between the feasibility and the precision of the tasks.

References

[1] F. Bensalah and F. Chaumette. Compensation of abrupt motion changes in target tracking by visual servoing. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'95*, volume 1, pages 181–187, Pittsburgh, Pennsylvania, August 1995.

[2] T.F. Chang and R.V. Dubey. A weighted least-norm solution based scheme for avoiding joints limits for redundant manipulators. *IEEE Trans. on Robotics and Automation*, 11(2):286–292, April 1995.

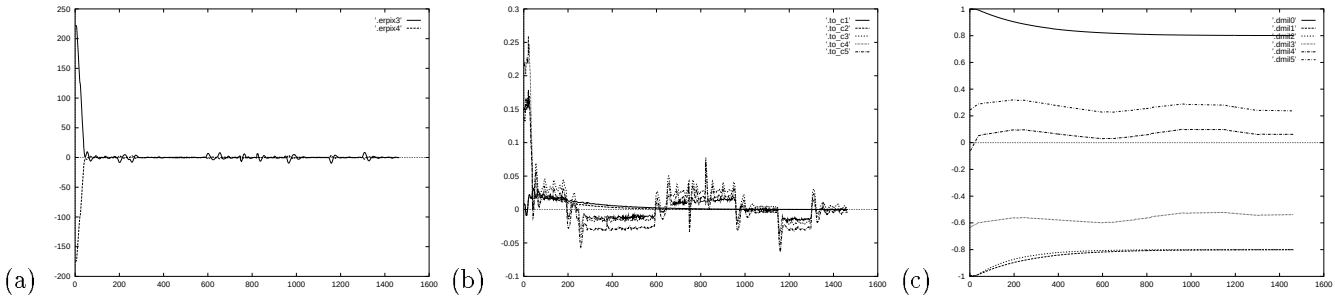


Figure 8: Target tracking and joint limits avoidance (a) error $(\underline{P} - \underline{P}_d)$ (b) joint velocities $\dot{\underline{q}}$ $(m/s$ and $rad/s)$ (c) Distance from the joint limits $\frac{\underline{q} - \underline{q}_{end}}{\underline{q}_{max} - \underline{q}_{end}}$

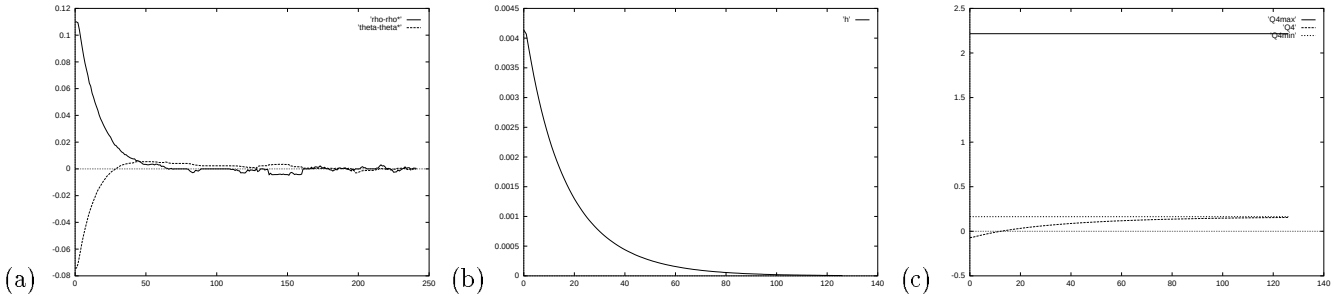


Figure 10: Positioning with respect to a straight line (a) Errors $\rho - \rho_d$ and $\theta - \theta_d$ (b) secondary cost function h_s, (c) joint position q_4 and activation thresholds

[3] F. Chaumette, S. Boukir, P. Bouthemy, and D. Juvin. Optimal estimation of 3D structures using visual servoing. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR'94*, pages 347–354, Seattle, Washington, June 1994.

[4] O. Egeland, M. Ebdrup, and S. Chiaverini. Sensory control in singular configurations-application to visual servoing. In *IEEE Int. Workshop on Intelligent Motion Control*, pages 401–405, Istanbul, August 1990.

[5] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Trans. on Robotics and Automation*, 8(3):313–326, June 1992.

[6] J.T Feddema, C.S.G. Lee, and O.R. Mitchell. Weighted selection of image features for resolved rate visual feedback control. *IEEE Trans. on Robotics and Automation*, 7(1):31–47, February 1991.

[7] K. Hashimoto, editor. *Visual Servoing : Real Time Control of Robot Manipulators Based on Visual Sensory Feedback*. World Scientific Series in Robotics and Automated Systems, Vol 7, World Scientific Press, Singapour, 1993.

[8] A.E. Hunt and A.C. Sanderson. Vision-based predictive robotic tracking of a moving object. Technical Report CMU-RI-TR-82-15, Carnegie-Mellon University, January 1982.

[9] B. Nelson and P.K. Khosla. Strategies for increasing the tracking region of an eye-in-hand system by singularity and joint limits avoidance. *International Journal of Robotics Research*, 14(3):255–269, June 1995.

[10] N. Papanikolopoulos, B. Nelson, and P.K. Khosla. Full 3D tracking using the controlled active vision paradigm. In *7th IEEE Symposium on Intelligent Control, Th. Henderson editor*, pages 267–274, Glasgow, Scotland, August 1992.

[11] C. Samson, M. Le Borgne, and B. Espiau. *Robot Control: the Task Function Approach*. Clarendon Press, Oxford, Royaume Uni, 1991.

[12] C.W. Wampler. Manipulator inverse kinematic solutions based on vector formulations and damped least squared method. *IEEE Trans. on Systems, Man, and Cybernetics*, 16(1):93–101, January 1986.