



HAL
open science

A redundancy based control law for executing a coordinated vision-based task using a dual arm robotic system

Renliw Fleurmond, Viviane Cadenat

► To cite this version:

Renliw Fleurmond, Viviane Cadenat. A redundancy based control law for executing a coordinated vision-based task using a dual arm robotic system. International Workshop IEEE Electronics, Control, Measurement, Signals and their application to Mechatronics (ECMSM), Jun 2015, Liberec, Czech Republic. hal-01959499

HAL Id: hal-01959499

<https://hal.science/hal-01959499>

Submitted on 18 Dec 2018

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

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

A redundancy based control law for executing a coordinated vision-based task using a dual arm robotic system

Renliw Fleurmond
CNRS, LAAS,
7 Avenue du Colonel Roche
F-31400 Toulouse, France
Email: rfleurmo@laas.fr

Viviane Cadenat
CNRS, LAAS,
7 Avenue du Colonel Roche
F-31400 Toulouse, France
Email: cadenat@laas.fr

Abstract—This paper deals with the problem of coordinating a dual arm robot equipped with several cameras. Our goal is to use the redundancy formalism to realize a real coordination of the two arms with an image based control, while satisfying additional constraints such as avoidance of the joint limits and occlusions. Simulation results validate our approach.

I. INTRODUCTION

Recent evolutions have shown that robots are more and more expected to intervene in environments designed by humans for humans. This trend is visible both in industrial and service settings with various applications ranging from domestic tasks to manufacturing missions such as parts assembly [1]. To perform these human oriented tasks, the robots mechanical structure has evolved, leading to the development of anthropomorphic or bi-manual systems. This has led to an increased interest for the domain of dual arm manipulation. Although basic robot manipulation has been a widely investigated field, the higher complexity induced by the use of two or several arms introduces new challenges at control, planning, modeling and reasoning levels [1].

Dual arm manipulation tasks are generally divided into two classes [1]: non-coordinated manipulation where the two arms are moving independently and coordinated manipulation where the two end-effectors are performing different parts of a common task. We focus on this last case where both end effectors are cooperating to perform a common task, for example open a bottle [2]. This problem has been tackled at different levels and through many approaches [1]. See [1] for a survey. Here we consider the specific problem of coordinated control. We can mention the pure position feedback which is sensitive to modeling errors [3], hybrid position/force based control laws [4] or active compliance control laws [5]. With this approach, the object position must be accurately known. Vision as exteroceptive data can be also exploited to perform coordination with dual arms robots. The robot is then controlled using visual servoing. Although it offers nice properties (possibility of using several cameras [6], good robustness with respect to modelling errors [7]), to the best of our knowledge, there are only few works which have used it for dual arm manipulation [1]. We may nonetheless mention the works by Miyabe [8], Hynes [9], Zereik [10] and Vahrenkamp [11] where this control approach is used to grasp

an object with one or both arms [8], [10], [11] or tie surgical nodes [9]. However, in these works, the arms are separately or alternatively controlled, which means that the coordination problem is not really addressed.

In this paper, our objective is to perform a coordinated task using image-based visual servoing because of its above mentioned nice properties. In previous works [12], we have already developed such a control strategy using the informations provided by several cameras mounted on our dual arm robotic system. The task to be realized was to recap a pen. However, some constraints such as the visibility or the joint limits have not been taken into account. Here, we aim at extending our approach to integrate these constraints in our control law. To do so, our idea is to benefit from redundancy. Indeed, on the one hand our robot is equipped with two 7 degrees of freedom arms. On the other hand, it has been shown in [2] that coordinated manipulation tasks such as the assembly task considered above can be expressed by the absolute pose of the object and/or the relative pose between both end-effectors. Thus its realization depends on the relative pose between the cap and the pen, which means that its maximal dimension is six and that a strong redundancy is available. Our idea is then to analyze and compare the efficiency of several projectors to define our own solution.

This paper is organized as follows: the next section briefly presents the modeling of the coordinated task and our main previous results. The third part proposes a brief overview of the main approaches allowing to deal with redundancy in the context of visual servoing. We then compare these approaches to choose our own solution. The additional constraints are described in the fourth section, while some simulation results validating our approach are finally presented in the last one.

II. MODELING THE PROBLEM

A. The robot

Our robotic platform is the PR2 developed by Willow Garage. It consists of an omni-directional mobile base equipped with two 7-DOF robotic arms. We will consider in our study that only the arms are moving. This robot has four cameras on the head and a camera at each forearm. We will only use one camera in the head and one in a forearm to get

complementary points of view. We assume that the cap and the pen are respectively gripped by the right and left arms and that all the objects can be seen by both cameras. We also assume that the latter objects are cylindrical and can be described by the same set of parameters. We denote by q_r and

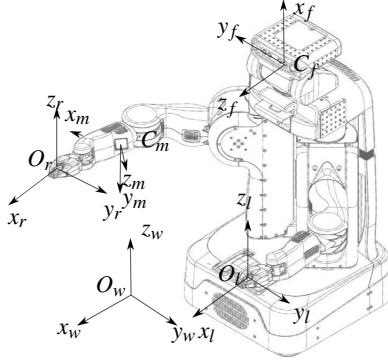


Fig. 1: The robotic platform and the different frames.

q_l (respectively \dot{q}_r and \dot{q}_l) the joint coordinates (respectively the joint velocities) of the right and left arms. We introduce the different frames which will be necessary to model our problem (see figure 1). We denote by F_w the world frame and F_r and F_l are respectively the frames linked to the left and right end effectors. F_f and F_m are respectively the frames attached to the head and forearm cameras. The kinematic screws of the right and left end effectors T_r and T_l , and of the mobile camera on the forearm T_m with respect to world frame are given by the direct differential kinematic model as follows:

$$T_r = J_r \dot{q}_r, \quad T_l = J_l \dot{q}_l, \quad T_m = J_{5r} \dot{q}_r \quad (1)$$

where J_r , J_l and J_{5r} are 6×7 matrices which have been already determined.

B. The visual features

As previously mentioned, the cap and the pen are supposed to be cylindrical objects. To perform the task, it is necessary to position them in the image and to monitor the translation of each cylinder along its axis. Three visual features suffices to respect these requirements [12]: the polar parameters $(\rho; \theta)$ of the inertia axis of the cylinder (the part coloured in grey in figure 2) and parameter k which expresses the position of one end point $E(x_e; y_e)$ of the cylinder on the straight line as shown in figure 2. Thus our image cues vector s is made of

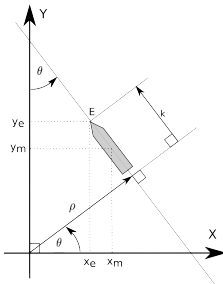


Fig. 2: Visual features used: ρ , θ and k .

three components $S = [\rho \ \theta \ k]^T$. We denote by S_{fc} and S_{mc} the visual features corresponding to the cap seen by respectively the fixed and the moving cameras, S_{fp} and S_{mp} the visual features corresponding to the pen seen by respectively the fixed and the moving cameras. More details about the computation of the visual features can be found in [12].

C. Visual servoing task

To recap the pen, the robot has to align both cylinders before connecting them. We then divide the task into three subtasks: the first one consists in making both cylinders axes be coplanar while ensuring a sufficient distance between the two objects. The second one is intended to align the latter axes while the last one allows to bring the cap near the pen. Thus, it suffices to sequence these three subtasks to fulfill our initial objective: recap the pen. Now, it remains to model these three subtasks and to design a control law allowing to perform them. To do so, we have chosen to use the task function formalism [13]. In this framework, the task is modeled by a function $e(q, t)$ chosen so that its regulation will ensure the realization of the mission. It often expresses as an error between the current vector of visual features S and the desired ones S^* , S^* being often constant. However, such a choice would lead to the independent control of both arms, which prevents from executing the task in a coordinated manner [12]. This is the reason why we have chosen to express the task functions as a relative error between the visual features representing the cap and the pen. We have shown in [12] that the task functions e_i ($i \in \{1, 2, 3\}$) representing the above mentioned subtasks can be written using the following general form :

$$e_i = H_i \cdot \left(\begin{bmatrix} S_{mc} - S_{mp} \\ S_{fc} - S_{fp} \end{bmatrix} - A_i \right) \quad (2)$$

where H_i is an activation matrix which allows to select only the necessary visual features, and A_i a constant vector. Their expressions can be found in [12]. In this way we have built a multi-cameras visual servoing task to implement the coordination between two arms.

D. The control law

We have considered the two arms as a single robotic system to be controlled as in [12]. Thus, we get:

$$q = [q_r^T \quad q_l^T]^T \quad (3)$$

Therefore the dimension of our system is $n = 7 + 7$. As classically done in the visual servoing area [7], it is possible to impose an exponential decay to make each task function e_i vanish. The corresponding controller is given by [13] :

$$\dot{q} = -J_i^+ \lambda_i e_i \quad (4)$$

where λ_i is a positive gain or a positive-definite matrix, J_i the Jacobian of the task function and J_i^+ its Moore-Penrose inverse. The expression of J_i and its computation are detailed in [12]. However, the above mentioned control law only makes the current task function e_i decrease to zero. From this, it follows that: (i) it is necessary to sequence the different control laws to perform successively the three subtasks and recap the pen; and (ii) the constraints related to visibility or to joint limits

are not handled. The first problem has already been solved in [12] using the following control law:

$$\dot{q}(t) = \dot{q}_C(t) - \exp(-\mu(t - t_s))(\dot{q}_C(t) - \dot{q}_O(t_s)) \quad (5)$$

where $\dot{q}_O(t)$ and $\dot{q}_C(t)$ are respectively the old and current control to be applied to the robot, t_s is the switching instant and μ is a real positive number which regulates the transition delay. We now focus on the second point: how to take into account constraints in our problem?

III. REDUNDANCY WITH TASK FUNCTION APPROACH

This section presents and analyzes the approaches allowing to deal with redundancy within the task function formalism. Redundancy occurs when the execution of the main task does not require to use all the robot degrees of freedom. Some of them can then be exploited to realize other objectives. If all these objectives can be proven to be compatible, that is if they can be simultaneously realized, the problem can be easily solved. Indeed, it suffices to associate a task function e_i to each subtask, define a global task function e by stacking the k different elements e_i and finally apply equation (6) to obtain the control law.

$$\dot{q} = \begin{bmatrix} J_1 \\ J_2 \\ \vdots \end{bmatrix}^+ \cdot \begin{bmatrix} \dot{e}_1^* \\ \dot{e}_2^* \\ \vdots \\ \dot{e}_k^* \end{bmatrix} \quad (6)$$

J_i are the jacobian matrices of task functions e_i and \dot{e}_i^* their desired dynamics. Let us note that, in this case, all the tasks are perfectly performed. Unfortunately, it is often difficult to prove the compatibility of the tasks, especially for a highly redundant robotic system as ours. In such a case, the literature proposes to define a hierarchy between the different tasks, that is to identify the main objective to perform and the missions which are secondary with respect to it. For the sake of clarity, we consider the case of two tasks to be realized with a given priority. Let us denote by e_1 and e_2 the primary and secondary tasks which may be or not compatible. We also introduce the velocities allowing to make them vanish separately $\dot{q}_1 = J_1^+ \dot{e}_1^*$ and $\dot{q}_2 = J_2^+ \dot{e}_2^*$ where J_1 and J_2 are their respective Jacobian matrices, while \dot{e}_1^* and \dot{e}_2^* represent their desired dynamics. The basic redundancy framework, as proposed in [13], aims at minimizing the secondary objective e_2 under the constraint that the primary task e_1 is perfectly performed. The corresponding control law is known to be given by [13]:

$$\dot{q} = \dot{q}_1 + P_1 \dot{q}_2 \quad (7)$$

where $P_1 = I - J_1^+ J_1$ is a projector onto the kernel of J_1 so that the execution of e_2 does not disturb the execution of e_1 . This controller thus guarantees the proper convergence of e_1 , while e_2 is realized at best using the remaining degrees of freedom [14]. It is then clear that e_2 might not be perfectly performed. In order to improve the execution of the secondary objective, Siciliano and Slotine have proposed to take into account the effect of the realization of e_1 on e_2 . The corresponding control law expresses as [15]:

$$\dot{q} = \dot{q}_1 + (J_2 P_1)^+ (\dot{e}_2^* - J_2 \dot{q}_1) \quad (8)$$

More recently, the concept of directional redundancy has been introduced [16]. The goal is to integrate in the control law all

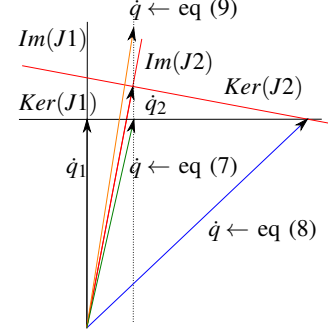


Fig. 3: The tasks e_1 and e_2 seems to be compatible.

the motions which do not impede the execution of the main task. The corresponding controller is defined by [16]:

$$\dot{q} = \dot{q}_1 + \tilde{P}(J_1, \dot{q}_2) \cdot \dot{q}_2 \quad (9)$$

where \tilde{P} depends on both J_1 and \dot{q}_2 (see [16] for more details). This projector is less restrictive than the classical one. It enables not only components of \dot{q}_2 belonging to $Ker(J_1)$ as the classical one, but also components which can make the task function e_1 decrease. It must be mentioned that, to avoid potential instability problems, the norm of enabled components is limited inside the projector. Finally, we may also mention the works by Marey [17] who proposes to reduce the dimension of the main task by considering the norm of e_1 instead of the complete vector. A new projection operator can then be defined, but it becomes singular when the main task is close to zero. It is then necessary to switch back to one of the cases considered above. We have then chosen not to use this last projector. Now let us analyze the behaviour of each previously mentioned controller.

Controller (7) relies on the classical projector P_1 which guarantees that the execution of e_2 does not disturb the convergence of e_1 . The realization of the primary task is then always ensured, the second one being performed at best. Controller (8) tries to execute perfectly both objectives, which is an interesting property. However, it introduces a great drawback: some instability might arise when these objectives are not fully compatible. Finally, controller (9), based on directional redundancy, leads to higher velocities, which allows to reduce the task convergence time. However, it appears to be sensitive to noise and some oscillations or even instability might occur if the speeds values are too high. On the base of these results, we believe that controllers (7) and (8) present interesting properties. The first one because it always offers a solution to fulfill several objectives at the same time; the second one because it is more efficient than (7) when the tasks are compatible. Therefore, we propose to build our control strategy on these two controllers. Our idea is to switch between them depending on the compatibility of the tasks. The incompatibility condition is generally fulfilled when the directions of \dot{q}_1 and \dot{q}_2 induce opposite motions, that is when the scalar product $\dot{q}_1^T \dot{q}_2$ is negative. However, this condition does not allow to discriminate all the cases where the tasks seem to be compatible whereas they are not. As one can see in the figure 3, the control vector provided by equation (8) might tend towards infinity although the scalar product $\dot{q}_1^T \dot{q}_2 > 0$, which means that \dot{q}_1 and \dot{q}_2 do not a priori induce

opposite motions. To take this kind of problem into account, we have decided to keep the solution which provides the smallest velocity. Our control strategy is finally given by:

$$\begin{aligned} \dot{q}_p &= \dot{q}_1 + P_1 \dot{q}_2, \quad \dot{q}_o = \dot{q}_1 + (J_2 P_1)^+ J_2 (\dot{q}_2 - \dot{q}_1) \\ \dot{q} &= \begin{cases} \dot{q}_o & \text{if } \dot{q}_1^T \dot{q}_2 > 0 \text{ and } \|\dot{q}_o\| < \|\dot{q}_p\| \\ \dot{q}_p & \text{otherwise} \end{cases} \end{aligned} \quad (10)$$

In this way, we combine the safety of the classical projector with the optimum convergence of the secondary task whenever it is possible. We can also ensure that the control law will be continuous, because we switch between the two chosen controllers when they become equal.

It is possible to extend the computation to more than two tasks. In this case, the computation of the control law is iterative. We denote by J_i the jacobian of the i^{th} task in descending order of priority and by \dot{q}_i the corresponding control law.

$$\begin{aligned} P_0^a &= I, \quad \dot{q}_0^a = 0 \\ \dot{q}_{p(i+1)} &= \dot{q}_i^a + P_i^a \cdot \dot{q}_{i+1} \\ \dot{q}_{o(i+1)} &= \dot{q}_i^a + (J_{i+1} P_i^a)^+ \cdot J_{i+1} \cdot (\dot{q}_{i+1} - \dot{q}_i^a) \\ \dot{q}_{i+1}^a &= \begin{cases} \dot{q}_{o(i+1)} & \text{if } \dot{q}_{o(i+1)}^T \dot{q}_{i+1} > 0 \text{ and} \\ & \|\dot{q}_{o(i+1)}\| < \|\dot{q}_{p(i+1)}\| \\ \dot{q}_{p(i+1)} & \text{otherwise} \end{cases} \\ P_{i+1}^a &= P_i^a - (J_{i+1} P_i^a)^+ (J_{i+1} P_i^a) \end{aligned} \quad (11)$$

The expression of the total control law of k tasks is given by:

$$\dot{q} = \dot{q}_k^a \quad (12)$$

To apply this formalism in our case, it is necessary to isolate the different tasks to be executed and decide a hierarchy between them. As previously mentioned, our goal is to perform the vision based task while avoiding joint limits and image cues losses. For the sake of the robot safety and because our robot is highly redundant, the joint limit avoidance task is awarded the highest priority. Then follows the vision-based task which allows to recap the pen and finally the visibility task which is given the lowest priority. Now, we focus on the definitions of the task functions allowing to represent these two constraints.

IV. MODELLING THE JOINT LIMIT AND THE IMAGE FEATURES LOSS AVOIDANCE TASKS

A. Joint limits

First let us focus on the joint limits avoidance task. This task must be only enabled when a joint is close to its bounds. If we denote the physical limits of a joint i by $[q_{imin}, q_{imax}]$, a safe interval $[q_{si-}, q_{si+}]$ is defined by:

$$\begin{cases} q_{si-} &= q_{imin} + (1 - \tau) \cdot (q_{imax} - q_{imin}) \\ q_{si+} &= q_{imax} - (1 - \tau) \cdot (q_{imax} - q_{imin}) \end{cases}$$

where $\tau \in]0, 0.5[$ is a tuning parameter. In order to avoid limits and retain q_i in the safe interval, it is usual to build a task function F_{bi} which acts as a virtual repulsive force [16] on the system. For each joint i , we have chosen to make this force evolve as shown below and in figure 4.

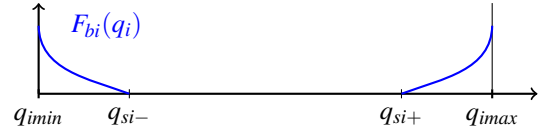


Fig. 4: Repulsive field outside the safe interval.

$$F_{bi} = \begin{cases} (q_i - q_{si+})^2 / (q_{imax} - q_i) & \text{if } q_i \geq q_{si+} \\ (q_{si-} - q_i)^2 / (q_i - q_{imin}) & \text{if } q_i \leq q_{si-} \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

From this, we define the task function allowing to avoid the joint limits as $F_b = [F_{b1} \dots F_{bn}]^T$. It tends to infinite when the joint value reaches its physical limits. In this way, the closer the joint is to its bounds, the greater the control law induced by the force is. We nonetheless limit the control to avoid actuator saturation. To make F_b vanish, we propose the following control law:

$$\dot{q}_b = -J_b^+ \lambda_b F_b(q) \quad (14)$$

where λ_b is a positive gain or a positive definite n by n matrix. As F_b is continuous, the control law is continuous too. In order to integrate this task in the redundancy scheme, it is necessary to express its jacobian. It is given by:

$$J_b = \begin{bmatrix} J_{b1} & 0 & \dots & 0 \\ 0 & J_{b2} & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & J_{bn} \end{bmatrix}$$

where J_{bi} expresses as:

$$J_{bi} = \begin{cases} (q_i - q_{si+})(2q_{imax} - q_{si+} - q_i) / (q_{imax} - q_i)^2 & \text{if } q_i \geq q_{si+} \\ (q_{si-} - q_i)(q_i - 2q_{imin} - q_{si-}) / (q_i - q_{imin})^2 & \text{if } q_i \leq q_{si-} \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

B. Visibility constraint

Now let us model the visibility constraint. Given the dimensions of the images provided by the cameras, we can specify a safe zone for the visual features as it has been done for the joints. Although it is easy to express the visibility

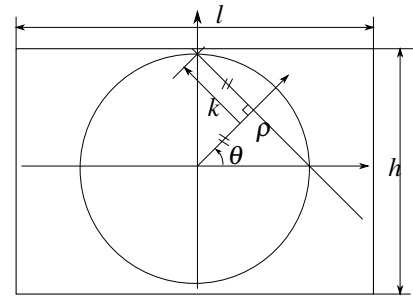


Fig. 5: Bounds for chosen image cues.

constraints for point-wise visual features, such a task is more complex for the image cues we have chosen ρ , k and θ . The two first ones correspond to distances measured in the image

plane, while the last one represents the orientation of the line. Therefore we have chosen to define a safe interval only for parameters k and ρ . To do so, it is first necessary to compute their admissible bounds k_{min} , k_{max} , ρ_{min} and ρ_{max} . We propose the following definition (see figure 5):

$$\rho_{max} = k_{max} = 0.5 \cdot \min(h, l) \cdot \cos(\pi/4) \quad (16)$$

$$\rho_{min} = k_{min} = -\rho_{max} \quad (17)$$

where $l \times h$ is the dimension of the image.

Now, let us denote by S_a the 12×1 vector containing all visual features for each camera and each object, and by S_t , the 8×1 vector containing the values of k and ρ for each camera and each object. The relation between S_a and S_t expresses as:

$$S_a = [S_{mc}^T \ S_{mp}^T \ S_{fc}^T \ S_{fp}^T]^T \quad (18)$$

$$S_t = H_t S_a = [\rho_{mc} \ k_{mc} \dots \ \rho_{fp} \ k_{fp}]^T \quad (19)$$

where H_t is an 8×12 constant matrix allowing to select the lines corresponding to k and ρ in S_a . To preserve the visibility, it is necessary that the i^{th} component s_i of S_t belongs to a safe interval $[s_{si-}, s_{si+}]$ defined as shown below:

$$s_{si-} = s_{imin} + (1 - \tau_s) \cdot (s_{imax} - s_{imin}) \quad (20)$$

$$s_{si+} = s_{imax} - (1 - \tau_s) \cdot (s_{imax} - s_{imin}) \quad (21)$$

where $[s_{imin}, s_{imax}]$ respectively correspond to the minimal and maximal values of s_i and $\tau_s \in]0, 0.5[$ is a tuning parameter. From this, it is possible to determine the task function $F_v^T = [F_{v1} \dots F_{v8}]$ so that its regulation to zero preserve the visibility. Following a similar reasoning to the one developed for the joint limits avoidance, we propose to define its components F_{vi} as follows:

$$F_{vi} = \begin{cases} (s_i - s_{si+})^2 / (s_{imax} - s_i) & \text{if } s_i \geq s_{si+} \\ (s_{si-} - s_i)^2 / (s_i - s_{imin}) & \text{if } s_i \leq s_{si-} \\ 0 & \text{otherwise} \end{cases} \quad (22)$$

Now, it remains to compute the control law \dot{q} making F_v vanish. To do so, it is first necessary to determine the time derivative of S_t . In [12], we have already shown that $\dot{S}_a = J_s \dot{q}$ (see [12] for the proof details and the expression of matrix J_s). From this, we can deduce that: $\dot{S}_t = H_t \cdot J_s \cdot \dot{q}$. Furthermore, $\dot{F}_v = J_v \cdot \dot{S}_t$, where J_v is given by:

$$J_v = \begin{bmatrix} J_{v1} & 0 & \dots & 0 \\ 0 & J_{v2} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & J_{vn} \end{bmatrix}$$

where J_{vi} expresses as:

$$J_{vi} = \begin{cases} (s_i - s_{si+})(2s_{imax} - s_{si+} - s_i) / (s_{imax} - s_i)^2 & \text{if } s_i \geq s_{si+} \\ (s_{si-} - s_i)(s_i - 2s_{imin} - s_{si-}) / (s_i - s_{imin})^2 & \text{if } s_i \leq s_{si-} \\ 0 & \text{otherwise} \end{cases} \quad (23)$$

Finally the control law allowing to perform this task is defined as shown below:

$$\dot{q} = -(J_v H_t J_s)^+ \lambda_v F_v \quad (24)$$

where λ_v is a positive gain or a positive definite matrix.

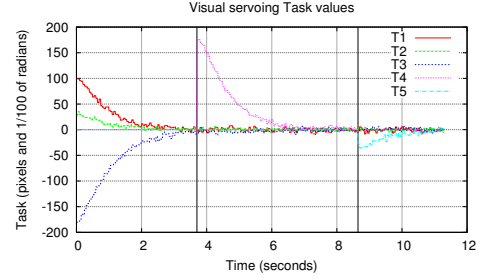


Fig. 6: Evolution of the visual servoing task functions e_1 , e_2 and e_3 .

V. SIMULATION RESULTS

We now present simulation results to validate our approach. A Gaussian noise has been introduced on the visual features (mean = 0, standard deviation = 2 pixels)¹. The control law is updated at 15 Hertz rate which is the minimal frame rate of the cameras on the robot. At the beginning of the task, it is assumed that the cap and the pen have already been grasped by the robot. The initial configurations of both arms have been chosen so that problems of visibility and joint limits avoidance can occur. The simulation has been conducted using ROS² and Gazebo³ which offer conditions closer to the ones of experimental tests. The trajectory realized by the robot has been recorded. The corresponding video is available at <http://homepages.laas.fr/rfleurgo/>.

Now, let us analyze the obtained results. Figures 6, 7 and 8 respectively show the evolution of the components of the task functions e_i (visual servoing), F_b (joint limits avoidance) and F_v (visibility tasks). As we can see, the components of F_b vanish more rapidly than the ones of F_v . Indeed, it has the lowest priority and it is not always fully compatible with the two other ones. Now, let us consider the case of the visual servoing task. Let us recall that in this case we have to execute successively the three above mentioned subtasks. Therefore, we have to make e_1 , then e_2 and finally e_3 vanish. This is exactly what occurs on figure 6 where the two switching instants are materialized by two vertical black lines. We can then conclude that the corresponding objectives are fulfilled and that our pen is correctly recapped, which is shown by our video. In addition, we have also plotted the evolution of one robot joint with and without the joint limits avoidance task. The results are shown in figure 9. As we can see, if this task is not enabled, the joint reaches its bounds and the task is more hardly achieved. This result clearly demonstrates the interest and the efficiency of our control strategy.

VI. CONCLUSION

In this paper, we have developed a multi cameras vision-based control strategy allowing to coordinate two manipulator arms while taking into account additional constraints. The task to be performed has been defined by visual features

¹The sensors provide 640×480 pixels images.

²<http://www.ros.org/>

³<http://gazebosim.org/>

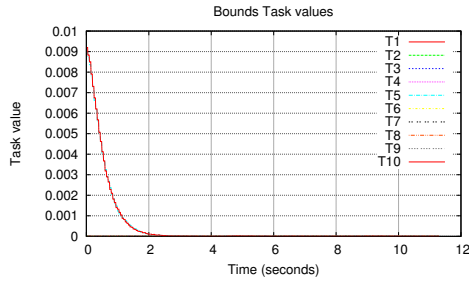


Fig. 7: Evolution of the task function F_b for avoiding joints limits.

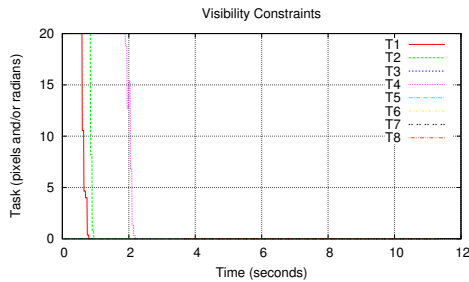


Fig. 8: Evolution of the visibility task function F_s .

characterizing the relative pose between the end-effectors so that a coordination between both arms has been achieved, and the task dimension is minimal. We have also taken advantage of the system redundancy to integrate constraints on the visual features visibility and the joint limits avoidance. To do so, we have developed a method allowing to perform at best the different tasks to be executed. The control strategy has been evaluated in conditions close to the experimental ones (introduction of noises, use of ROS and Gazebo, ...) and the obtained simulation results have demonstrated its interest and its efficiency. Now, to go further and improve our approach, it is necessary to take into account the problems of singularities and collisions which may still hamper the execution of the task. Finally, we also plan to experimentally validate our approach on the LAAS PR2 robot and to perform more complex coordination tasks.

REFERENCES

- [1] C. Smith, Y. Karayiannidis, L. Nalpantidis, X. Gratal, P. Qi, D.V. Dimarogonas, and D. Kragic. Dual arm manipulation - a survey. *Robotics and Autonomous Systems*, 60(10):1340 – 1353, 2012.
- [2] B.V. Adorno, P. Fraitse, and S. Druon. Dual position control strategies using the cooperative dual task-space framework. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 3955–3960, 2010.
- [3] F. Caccavale, N. Ciro, Bruno Siciliano, and Luigi Villani. Achieving a cooperative behavior in a dual-arm robot system via a modular control structure. *Journal of Robotic Systems*, 18(12):691–699, 2001.
- [4] T. Watanabe, K. Harada, Zhongwei Jiang, and T. Yoshikawa. Object manipulation under hybrid active/passive closure. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 1013–1020, April 2005.

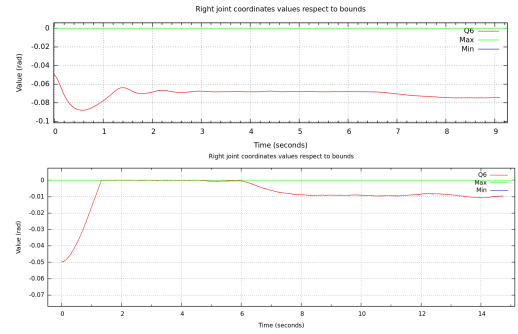


Fig. 9: Evolution of the considered axis when the joint limits avoidance task is enabled (top) and disabled (bottom).

- [5] Fabrizio Caccavale and Masaru Uchiyama. Cooperative manipulators. In Bruno Siciliano and Oussama Khatib, editors, *Springer Handbook of Robotics*, pages 701–718. Springer Berlin Heidelberg, 2008.
- [6] O. Kermorgant and F. Chaumette. Avoiding joint limits with a low-level fusion scheme. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 768–773, Sept 2011.
- [7] F. Chaumette and S. Hutchinson. Visual servo control part 1: Basic approaches. *IEEE Robotics and Automation Magazine*, 13(4):82–90, December 2006.
- [8] T. Miyabe, A. Konno, and M. Uchiyama. Automated object capturing with a two-arm flexible manipulator. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, volume 2, pages 2529–2534 vol.2, 2003.
- [9] P. Hynes, G.I. Dodds, and A. J. Wilkinson. Uncalibrated visual-servoing of a dual-arm robot for mis suturing. In *The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechanics*, pages 420–425, 2006.
- [10] E. Zereik, A. Sorbara, G. Casalino, and F. Didot. Autonomous dual-arm mobile manipulator crew assistant for surface operations: force/vision-guided grasping. In *Recent Advances in Space Technologies, 2009. RAST'09. 4th International Conference on*, pages 710–715, June 2009.
- [11] N. Vahrenkamp, C. Boge, K. Welke, T. Asfour, J. Walter, and R. Dillmann. Visual servoing for dual arm motions on a humanoid robot. In *9th IEEE-RAS International Conference on Humanoid Robots*, pages 208–214, 2009.
- [12] Renliw Fleurmond and Viviane Cadenat. Multi-cameras visual servoing to perform a coordinated task using a dual arm robot. In *International Conference on Informatics in Control, Automation and Robotics*, September 2014.
- [13] C. Samson, M. Le Borgne, and B. Espiau. *Robot control: the task function approach*. Oxford engineering science series. Clarendon Press, 1991.
- [14] Viviane Cadenat, Phillippe Soueres, and Michel Courdesses. Using system redundancy to perform a sensor-based navigation task amidst obstacles. *International Journal of Robotics & Automation*, 16(2):61–73, 2001.
- [15] B. Siciliano and J.-J. E. Slotine. A general framework for managing multiple tasks in highly redundant robotic systems. In *Advanced Robotics, 1991. 'Robots in Unstructured Environments', 91 ICAR, Fifth International Conference on*, pages 1211 –1216 vol.2, June 1991.
- [16] N. Mansard and F. Chaumette. Directional redundancy for robot control. *Automatic Control, IEEE Transactions on*, 54(6):1179–1192, 2009.
- [17] M. Marey and F. Chaumette. A new large projection operator for the redundancy framework. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 3727–3732, May 2010.