Jacobian Learning Methods for Tasks Sequencing in Visual Servoing

Nicolas Mansard[1], Manuel Lopes[2], José Santos-Victor[2], François Chaumette[1]

[1] *IRISA - INRIA Rennes, France*
E-Mail : {nmansard,chaumett}@irisa.fr
[2] *Instituto de Sistemas e Robótica, Instituto Superior Técnico, Lisboa, Portugal*
E-Mail : {macl,jasv}@isr.ist.utl.pt

Abstract— In this paper, the coupling between Jacobian learning and task sequencing through the redundancy approach is studied.

It is well known that visual servoing is robust to modeling errors in the jacobian matrices. This justifies why jacobian estimation does not usually degrade the system convergence. However, we show that this is not true anymore when the redundancy formalism is used. In this case the jacobian matrix is also necessary to compute projection operators for task decomposition, which is quite sensitive to errors. We show that learning improves the servoing performance, when task sequencing is used. Conversely, sequencing improves the convergence of learning, especially for tasks involving several degrees of freedom. Eye-in-hand and eye-to-hand experiments have been performed on two robots with six degrees of freedom.

I. INTRODUCTION

Visual servoing methods provide very efficient and robust solutions to control robot motions [7]. It provides high accuracy for the final pose and good robustness to camera calibration and other setting parameters. The redundancy formalism [10], [14] extends the task-function approach [16] to compute a control law that realizes a main task, while simultaneously taking supplementary constraints into account. It can be used when the main task does not constrain all the robot degrees of freedom (DOF). A secondary task can then be added to meet a second objective without disturbing higher priority tasks. The control law for the second task is projected into the set of motions constituting the null space of the first task, thus leaving the first tasks unmodified. The computation of the projection operator is based on the jacobian of the first task.

This approach involves the computation of the task jacobian, linking the evolution of the visual features to the robot articular motion. It thus requires knowledge about the camera-world and world-actuator transformations that influence the interaction matrix (relating image and camera velocities) and the robot jacobian (relating end-effector and joint velocities). Such transformations are usually obtained during an offline calibration phase.

However, full system calibration (and even a coarse one) is not always possible and/or desirable. Some robots may lack proprioceptive sensors to provide the necessary information and some parameters may vary over time, due to malfunction, changes in mechanical parts or modification in the camera lenses. Even when calibration information is available, the analytic computation of the interaction matrix often requires an estimate of the depth of the tracked features. For all these reasons, a perfect computation of the task jacobian can be very difficult or even impossible in practice.

Several methods have already been proposed to estimate the interaction matrix, the robot jacobian or the task jacobian. One of the first works was [6], where robust learning rule was derived and a convergence proof given. This method is based on the Broyden update rule, well known from optimization theory [5] and has been widely used in robotic applications with visual control. In [8], it was used for visually-guided grasping. Ad hoc task sequencing was used in [3] to separate the reaching phase and the grasping phase. Another object-grasping task is presented in [12], where the estimation algorithm is used to provide an approximation of a highly non-linear mapping, using several local linear models. An error function for the jacobian approximation is defined. By minimizing this function with a Newton method, a time-varying system is obtained. This work was applied to both eye-in-hand and eye-to-hand systems with moving targets [15]. It was also suggested to learn the inverse jacobian directly, instead of the jacobian [9], although only an offline formulation was proposed in that work. A complete discussion about adaptive identification methods for slowly varying parameters is presented in [2]. It also presents a new method to improve the robustness of parameter identification, by combining directions with new information with those where the information had been lost and had to be recovered.

All these works are only focused in the jacobian estimation itself. It is well known that visual servoing offers a high level of robustness to jacobian errors. As a consequence, learning the jacobian usually provides satisfactory results. However, when computing the control law for a set of tasks using the redundancy formalism, the jacobian is also used to compute a projection operator. As will be shown in this paper, this computation is not robust at all, and the highest priority task may be strongly disturbed by the projector estimation errors. **Learning the jacobian for a redundancy-based servo scheme turns out to be more demanding than for single-task scheme** and has to be studied with care, which is one of the scope of this paper.

In [13] a solution was presented to stack redundant tasks on top of the others, until all degrees of freedom of the robot

are constrained, and the desired pose is reached. It is based on a generalization of the redundancy approach to several tasks [17].

In this work, we apply the sequencing method to coarsely calibrated systems. Hence, errors in the jacobian will cause errors in the computation of the projection operators, disturbing higher priority tasks. To overcome this problem, we use the disturbance itself as an error signal used to improve the jacobian online. We compare several estimation methods, in order to access their quality in terms of real-parameter estimation and online behavior of the system.

Our results show that it is possible to estimate the jacobian online, in the context of task sequencing for visual servoing. In addition, the learning stability is greatly improved by the task sequencing approach. Hence, **task sequencing and learning are intertwined, mutually constrained processes that bring additional performance and flexibility** to the control of complex robotic systems, when calibration information is unavailable or highly uncertain.

In Section II the stack of tasks structure used to sequence redundant tasks is defined, together with a control law that maintains all the tasks. Section III recalls several jacobian estimation methods, that are discussed in the experiments. Several sets of experiments have been realized to support our conclusions using two different six DOF robots (Section IV).

II. VISUAL SERVOING USING A STACK OF TASKS

In this section, we recall how to sequence redundant tasks and to maintain the tasks already achieved [13].

A. Redundancy formalism for two tasks

Let $\mathbf{q}$ be the articular vector of the robot. Let $\mathbf{e_1}$ and $\mathbf{e_2}$ be two tasks, $\mathbf{J_i} = \frac{\partial \mathbf{e_i}}{\partial \mathbf{q}}$ $(i = 1, 2)$ their jacobian, defined by:

$$\dot{\mathbf{e}}_{\mathbf{i}} = \frac{\partial \mathbf{e_i}}{\partial \mathbf{q}} \dot{\mathbf{q}} = \mathbf{J_i} \dot{\mathbf{q}} \qquad (1)$$

Since the robot is controlled using its articular velocity $\dot{\mathbf{q}}$, (1) has to be inverted. The general solution (with $i = 1$) is:

$$\dot{\mathbf{q}} = \mathbf{J_1^+} \dot{\mathbf{e}}_{\mathbf{1}} + \mathbf{P_1} \mathbf{z} \qquad (2)$$

where $\mathbf{P_1}$ is the orthogonal projection operator on the null space of $\mathbf{J_1}$ and $\mathbf{J_1^+}$ is the pseudo-inverse of $\mathbf{J_1}$. Vector $\mathbf{z}$ can be used to apply a secondary command, that will not disturb $\mathbf{e_1}$. Here, $\mathbf{z}$ is used to carry out at best a task $\mathbf{e_2}$. Introducing (2) in (1) (with $i = 2$) gives:

$$\dot{\mathbf{e}}_{\mathbf{2}} = \mathbf{J_2} \mathbf{J_1^+} \dot{\mathbf{e}}_{\mathbf{1}} + \mathbf{J_2} \mathbf{P_1} \mathbf{z} \qquad (3)$$

By inverting this last equation, and introducing the computed $\mathbf{z}$ in (2), we finally get:

$$\dot{\mathbf{q}} = \mathbf{J_1^+} \dot{\mathbf{e}}_{\mathbf{1}} + \mathbf{P_1} (\mathbf{J_2} \mathbf{P_1})^+ (\dot{\mathbf{e}}_{\mathbf{2}} - \mathbf{J_2} \mathbf{J_1^+} \dot{\mathbf{e}}_{\mathbf{1}}) \qquad (4)$$

Since $\mathbf{P_1}$ is Hermitian and idempotent (it is a projection operator), (4) can be written:

$$\dot{\mathbf{q}} = \mathbf{J_1^+} \dot{\mathbf{e}}_{\mathbf{1}} + \widetilde{\mathbf{J_2}}^+ \widetilde{\dot{\mathbf{e}}_{\mathbf{2}}} \qquad (5)$$

where $\widetilde{\mathbf{J_2}} = \mathbf{J_2} \mathbf{P_1}$ is the limited jacobian of the task $\mathbf{e_2}$, giving the available range for the secondary task to be performed without affecting the first task, and $\widetilde{\dot{\mathbf{e}}_{\mathbf{2}}} = \dot{\mathbf{e}}_{\mathbf{2}} - \mathbf{J_2} \mathbf{J_1^+} \dot{\mathbf{e}}_{\mathbf{1}}$ is the secondary task function, after subtracting the part $\mathbf{J_2} \mathbf{J_1^+} \dot{\mathbf{e}}_{\mathbf{1}}$ already accomplished by the first task. A very good intuitive explanation of this equation is given in [1].

B. Extending the redundancy formalism for several tasks

Let $(\mathbf{e_1}, \mathbf{J_1}) \dots (\mathbf{e_n}, \mathbf{J_n})$ be n tasks so that Task $\mathbf{e_i}$ should not disturb task $\mathbf{e_j}$ if $i > j$. A recursive extension of (5) is proposed in [17]:

$$\dot{\mathbf{q}}_{\mathbf{i}} = \dot{\mathbf{q}}_{\mathbf{i-1}} + (\mathbf{J_i} \mathbf{P_{i-1}^A})^+ (\dot{\mathbf{e}}_{\mathbf{i}} - \mathbf{J_i} \dot{\mathbf{q}}_{\mathbf{i-1}}) \qquad (6)$$

where $\mathbf{P_i^A}$ is the projector onto the null-space of the augmented Jacobian $\mathbf{J_i^A} = (\mathbf{J_1}, \dots \mathbf{J_i})$. The recursion is initialized by $\dot{\mathbf{q}}_{\mathbf{0}} = 0$. The robot velocity is $\dot{\mathbf{q}} = \dot{\mathbf{q}}_{\mathbf{n}}$.

Using this recursive equation directly, a projector has to be computed on each step of the computation. A recursive formula for the computation of the projector is proposed in [1]. We recall this equation here

$$\mathbf{P_i^A} = \mathbf{P_{i-1}^A} - \widetilde{\mathbf{J_i}}^+ \widetilde{\mathbf{J_i}} \qquad (7)$$

where $\widetilde{\mathbf{J_i}} = \mathbf{J_i} \mathbf{P_{i-1}^A}$ is the limited jacobian of the task i. The recursion is initialized by $\mathbf{P_0^A} = \mathbf{I}$ (identity matrix).

The control law is finally obtained by setting an exponential decrease for each task:

$$\forall i = 1..n, \quad \dot{\mathbf{e}}_{\mathbf{i}} = -\lambda_i \mathbf{e_i} \qquad (8)$$

where the parameters λ_i are used to tune the convergence speed of each task.

In this control law, two matrices have to be learned: $\mathbf{J_i}$ and $\mathbf{P_i^A}$. Since $\mathbf{P_i^A}$ can be computed from $\mathbf{J_i}$, we will learn only the jacobian and then compute the projection operator from it. In this sense, the two matrices are learned. The effect of the learning can be considered from two different points of view: convergence of a task while the jacobian is learned online and the disturbances caused upon higher priority tasks due to the effect of learning the projection operator.

C. Application to visual servoing

In this article, we propose an implementation of this control law using visual servoing. The task functions $\mathbf{e_i}$ used in the remainder of the text are computed from visual features:

$$\mathbf{e_i} = \mathbf{s_i} - \mathbf{s_i^*} \qquad (9)$$

where $\mathbf{s_i}$ is the current value of the visual features for task $\mathbf{e_i}$ and $\mathbf{s_i^*}$ their desired value.

The interaction matrix $\mathbf{L_{s_i}}$ related to $\mathbf{s_i}$ is defined so that $\dot{\mathbf{s}}_{\mathbf{i}} = \mathbf{L_{s_i}} \mathbf{v}$, where $\mathbf{v}$ is the camera kinematic screw. From (9), it is clear that the interaction matrix $\mathbf{L_{s_i}}$ and the task jacobian $\mathbf{J_i}$ are linked by the relation:

$$\mathbf{J_i} = \mathbf{L_{s_i}} \mathbf{J_q} \qquad (10)$$

where the matrix $\mathbf{J_q}$ denotes the robot jacobian ($\mathbf{v} = \mathbf{J_q} \dot{\mathbf{q}}$). In the following section we will see several methods that can be used to learn the jacobian matrix.

III. Jacobian Estimation Methods

A large amount of information is required to recompute the jacobian at each iteration. As presented just above, the jacobians can be divided in two parts: the articular jacobian $\mathbf{J_q}$, and the interaction matrix $\mathbf{L_s}$ (see (10)). The articular jacobian can only be computed if the full arm-eye calibration is available. Instead, the interaction matrices require some 3D parameters of the observed object. This can be estimated using pose computation (if the object model is available) or using homographies [4] between the image and relevant scene planes. In this last case, the scale parameters cannot be estimated, and the object depth has to be fixed *a priori* or estimated using other methods. All these parameters can lead to errors in the computation of the interaction matrix.

Different approaches can be used to estimate the tasks jacobians. Learning the task jacobian, $\mathbf{J_i}$, is quite difficult, due to the numerous non-linear terms involved in its analytical form. The experiments have proved the extreme difficulty in obtaining a good estimation of $\mathbf{J_i}$. In [12], a mixture of several linear models was described to tackle this problem.

Instead of trying to estimate $\mathbf{J_i}$, we have chosen to use an approximation of the articular jacobian, $\widehat{\mathbf{J_q}}$, computed from coarse robot calibration data. Then, since an approximation of $\mathbf{v}$ can be computed, only the matrices $\widehat{\mathbf{L_{s_i}}}$ need to be estimated. Furthermore, estimating $\mathbf{L_{s_i}}$ will allow to "absorb" some of the errors in $\mathbf{J_q}$, caused by the coarse robot calibration. This solution is also able to take into account the uncertainties in the target model, yielding better results, even when using a properly calibrated robot, as it will be shown in Section IV-B. In the following sub-sections, we present the methods used in the experiments to learn the jacobian.

A. Broyden Update

The first work presented for jacobian estimation in visual servoing was [6]. This method is based on the Broyden update rule. The Jacobian estimation is given by:

$$\hat{\mathbf{J}}(t+1) = \hat{\mathbf{J}}(t) + \alpha \frac{\left(\Delta\mathbf{e} - \hat{\mathbf{J}}(t)\Delta\mathbf{x}\right)\Delta\mathbf{x}^\top}{\Delta\mathbf{x}^\top\Delta\mathbf{x}} \qquad (11)$$

After observing some image motion, $\Delta\mathbf{e}$, caused by a motor command $\Delta\mathbf{x}$, the Jacobian is updated directly, with α defining the update speed. This method has several positive aspects: low memory usage because only the last observation is used; low computational cost and a single parameter to be tuned. When the motions are too small, this computation may become unstable. One solution consists in including a regularization term in the denominator to prevent singularities. Alternatively, the learning can simply be switched off, whenever the motion falls below a certain threshold.

In practice, it can be used to compute offline an estimation of the jacobian from a set of simple motions. Or the estimation of the jacobian can be updated online, using at the first iteration the result of the offline learning or an estimation of the analytical jacobian.

B. Correlation

A different approach can be made with least squares estimation [2]. Considering the cost function l as:

$$l = \sum_{i=0}^{t} \gamma^{i-t}(\Delta\mathbf{e} - \mathbf{J}\Delta\mathbf{x})^\top(\Delta\mathbf{e} - \mathbf{J}\Delta\mathbf{x})$$

The minimization off-line gives the usual least squares solution:

$$\mathbf{J} = \mathbf{Q}\mathbf{R}^+$$

where $\mathbf{Q}$ and $\mathbf{R}$ are:

$$\begin{aligned}\mathbf{Q} &= \sum_{i=0}^{t} \lambda^{(t-i)}\Delta\mathbf{e}_i^\top\Delta\mathbf{x}_i \\ \mathbf{R} &= \sum_{i=0}^{t} \lambda^{(t-i)}\Delta\mathbf{x}_i^\top\Delta\mathbf{x}_i\end{aligned}$$

or in an online formulation;

$$\begin{aligned}\mathbf{Q} &= \lambda\mathbf{Q} + \Delta\mathbf{e}_t^\top\Delta\mathbf{x}_t \\ \mathbf{R} &= \lambda\mathbf{R} + \Delta\mathbf{x}_t^\top\Delta\mathbf{x}_t\end{aligned} \qquad (12)$$

Like the *Broyden* update, the correlation learning can be used offline or online. However, it can not be used to update an approximation of the analytical jacobian since the two matrices $\mathbf{Q}$ and $\mathbf{R}$ are required. In practice, it is difficult to start the learning online without any offline training. In the experiment, a short offline training composed of some simple motions has always been used to initialize the online learning.

C. Direct-Inverse

Learning the jacobian boils down to minimizing the prediction error of the image velocities. Yet, for control purposes, we need the inverse map $\mathbf{J}^+$, that corresponds to the "reconstruction" of the robot joint velocities from image velocities. Thus, in [9], it was suggested that one should learn the inverse Jacobian directly, instead of the Jacobian. The cost function becomes:

$$l = \sum_{i=0}^{t} \gamma^{i-t}(\Delta\mathbf{x} - \mathbf{H}\Delta\mathbf{e})^\top(\Delta\mathbf{x} - \mathbf{H}\Delta\mathbf{e})$$

where $\mathbf{H} = \mathbf{J}^+$. This cost function can be seen as a reconstruction error, as opposed to the prediction error used before. The main advantage is that we no longer need to invert the Jacobian for computing the control law. An additional benefit is that the least-squares fitting requires the inversion of a smaller, possibly better conditioned, information matrix, R. This is particularly relevant for the task sequencing approach whereby the (sub-)task dimension is much smaller than the control space.

Using the online formulation (12) it is possible to propose an online version of the *direct-inverse* learning method. In the experiment, both the offline and the online version of the method were used. As for the classical correlation method, the online learning can not be initialized from an analytical value of the jacobian, and requires thus a short offline training.

IV. EXPERIMENTS AND RESULTS

In this section we present results comparing the quality of tasks sequencing using several methods of jacobian estimation. Two robots with different kinematics and servoing architectures were used for the experiments. We first describe quickly the selected visual features used for the experiments. Four representative experiments are then presented in detail.

A. Visual features for vision-based control

In order to have a better and easier control over the robot trajectory, approximately decoupled tasks were chosen. We have used visual features derived from the image moments. To simplify the image processing as we mainly focus on the control part, we have used a simple white-points-on-blackboard target as shown Fig 1. The first task $\mathbf{e_g}$ - *centering* - is based on the position of the center of gravity of the four points. The second task $\mathbf{e_Z}$ - *zooming* - uses the area of the object in the image to control the range between the robot and the target. The third task $\mathbf{e}_\alpha$ - *Z-rotation* - rotates the camera around the optical axis, so that the object will be correctly oriented in the image. It uses the orientation of the object in the image, which can be obtained from the second order moments. The last task $\mathbf{e_R}$ - *perspective correction* - uses third order moments to decouple v_x from ω_y and v_y from ω_x. The reader is invited to refer to [18] for more details.

B. Results

The three first experiments were realized with the robot Baltazar. Baltazar is an anthropomorphic robotic torso [11] equipped with a six DOF arm, an eleven DOF hand and a four DOF head. In the presented experiments, the target was attached to the robot hand. An eye-to-hand visual servo was used to position the hand parallel to the eye image plane, centered at a distance of $20cm$. This robot has a high payload/weight ratio causing some elasticities, its motors are equipped with position sensors but the lack of an home sensor causes some errors if a precise calibration is needed. The camera is coarsely calibrated and for the experiments a $4.5mm$ lens had to be used. Figure 1 presents the robot and the initial and final hand position. The total motion is about $30cm$ corresponding to a maximum joint translation of $90\,dg$.

The first experiment includes a comparison of the estimation methods presented in Section III. The second experiment shows that it is possible to correct the robot (coarse) calibration by learning only the interaction matrix when realizing a full sequencing. The third experiment is a first step to show (experimentally) that the trajectory obtained when applying a sequencing control law provides a very good dataset for learning. The last experiment was realized on an accurately calibrated robot (see Section IV-F). It shows that a small uncertainty can result in a big perturbation, and that the online estimation is able to provide a nearly perfect behavior.

C. Experiment 1 - online learning

In this first experiment, the stack has two tasks: centering ($\mathbf{e_g}$) and Z-rotation ($\mathbf{e}_\alpha$). The goal consists in testing how the
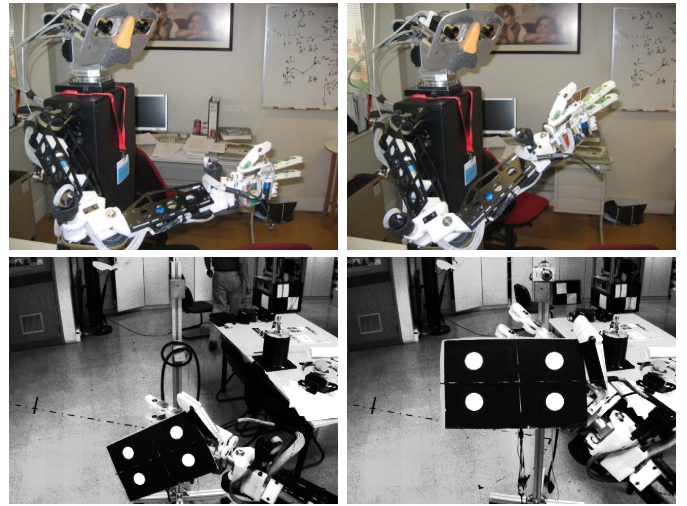

Fig. 1. Initial and final position. Top: outside view, Down: camera view

jacobian estimation errors influence the task sequencing, due to errors introduced in the projection operator. When the jacobian of the first task is mis-estimated, the centering is lost with the activation of the second task. When the error increases, the target moves further away from the image center, and could even leave the image if the disturbance is too strong (which results of course in the visual servoing failure).

Figure 2 presents the evolution of the error for the first task using analytic/offline learning versus online estimation methods. Figure 3 shows the result for the second (rotation) task. Offline learning relies on simple motions of the arm, done during approximately 250 iterations. Online learning was carried out at every frame.

As can be seen in Fig. 2, analytic or offline learning are worse, in terms of having a larger perturbation and longer convergence times. The perturbation is very important and it can not be reduced before the secondary task completion.

On the opposit, online estimation methods lead to much better results, outperforming the results with the analytical jacobian. Although a large disturbance appears when the second task is added, it is quickly reduced afterward. For online estimation, the amplitude of the perturbation ranged from 20 to 30 pixels. *Broyden* and *Correlation* methods were able to eliminate the error after 30 iterations. The maximal perturbation is equivalent to the one obtained with analytic computation, but the duration is much shorter.

The online *Direct-inverse* method was unable to reduce the error as fast as the two other methods or the analytic version, whatever the tuning realized. The advantage of directly minimizing the reconstruction errors instead of the prediction error does not appear significant in this setting. Indeed, to compute the projection it is necessary to have the direct map and the result, in the end, is worst.

It is also interesting to see that the task-error convergence is very similar for all methods (for Task $\mathbf{e_g}$, Fig. 2 before Iteration 50, and for Task $\mathbf{e}_\alpha$, Fig. 3). This emphasizes that the
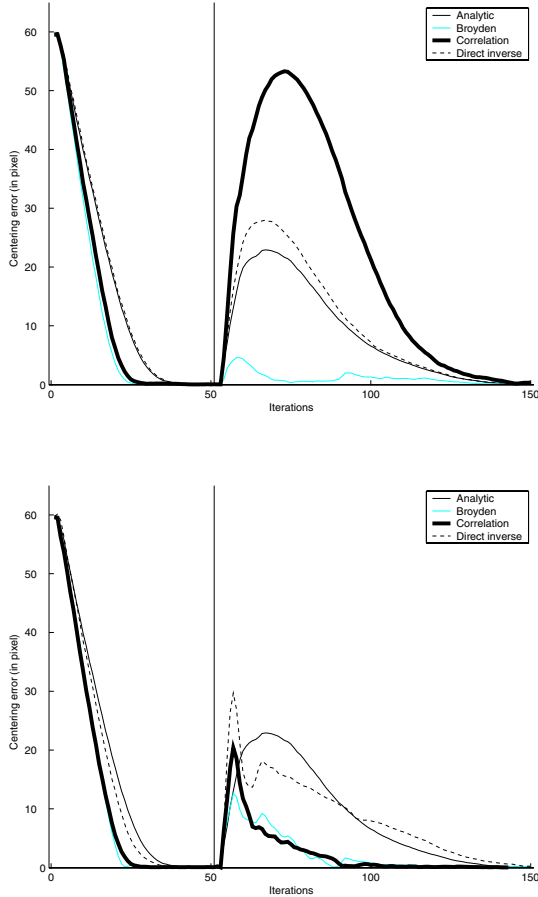
Fig. 2. Temporal evolution of the image error during servoing using offline (top) or online (bottom) learning methods. The vertical line shows the time instant where the second task started.
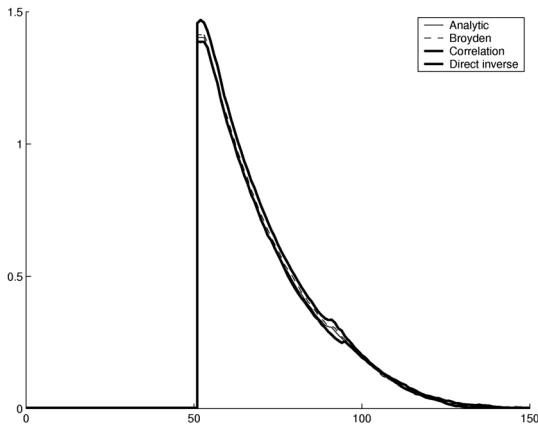

Fig. 3. Convergence rates for the second task (rotation) while trying to keep the centering at zero (refer to Figure 2). Due to robustness to jacobian error, the convergences are the same for all the tasks.

reduction of the perturbation is not made at the cost of worse convergence. The convergence is very robust to jacobian error, since all the task convergences are the same. It is nevertheless not true for the projection operator estimation, which is very sensitive and requires an accurate estimation.

Finally, we can note from several experiments starting from different initial positions and using different tasks that the online *Correlation* method produced better results in sense of perturbation amplitude, perturbation average and perturbation-correction time, when properly tuned. However, it is not as robust to gain-tuning as the *Broyden* approach that could solve the task in all situations with the same parameters settings (note the *Broyden* performances for offline learning).

D. Experiment 2 - generalization to more tasks

To verify that the method it is possible to learn all tasks, a complete sequence was done, consisting in: centering, zooming, Z-rotation and perspective. All online learning methods have been able to achieve the sequencing. The lowest perturbations were obtained using the *Correlation* method. Figure 4 cares the results of this method (on the bottom) with the analytical one (on the top). The vertical lines represent the time instant where a new task is added to the stack, after all the tasks already in the stack have converged to zero. It is interesting to see that in a badly calibrated system, the learning scheme yields better results than using the analytic solution, in terms of convergence speed, amplitude and average of the perturbations.

E. Experiment 3 - better learning through task sequencing

A very important point was to note that learning improves the sequencing quality by reducing convergence time and the size of the perturbations. At the same time, the sequencing generates more efficient trajectories for learning. This experiment tests the hypothesis that learning four simpler tasks in sequence is easier than learning four tasks at the same time.

We compared the learning when running the robot under three different control laws. During the first run, task sequencing was used, in the same way as in previous experiments. In the second trial, all tasks are active at the same time. In other words, the same formalism is used but every task is active from the beginning, as opposed to starting a new task only after all the previous ones are completed. The last trial consisted on classical visual servoing, using only one single task of full rank. The condition number of the full-rank jacobian matrix was then estimated at each iteration. When a sequencing was used, the jacobians of all tasks were piled up and the overall condition number evaluated.

For *Correlation* and *Direct-inverse* methods, the condition number was the same for the three experiments. For these learning methods, the sequencing does not improve the learning.

On the opposite, the hypothesis has turned out to be correct when using *Broyden* algorithm. Figure 5 shows that for the *Broyden* method, the condition number of the matrices are much worse for the full task and convergence cannot be attained.
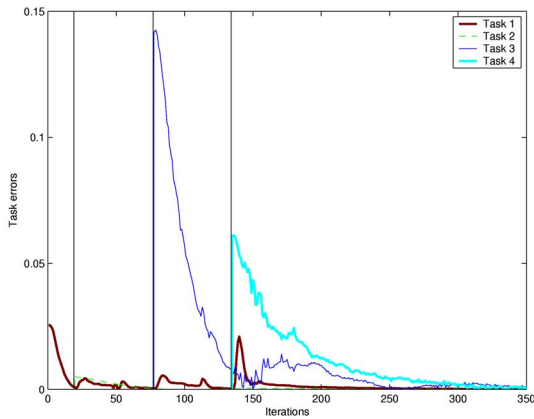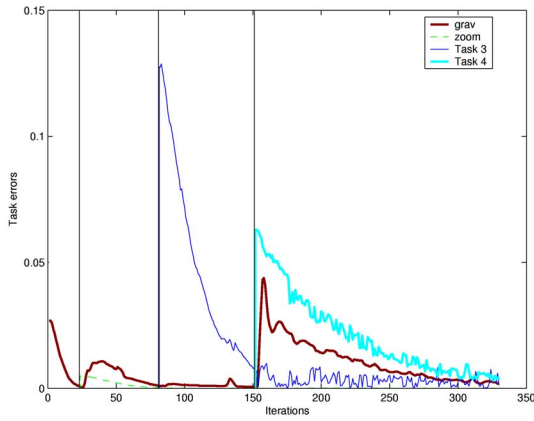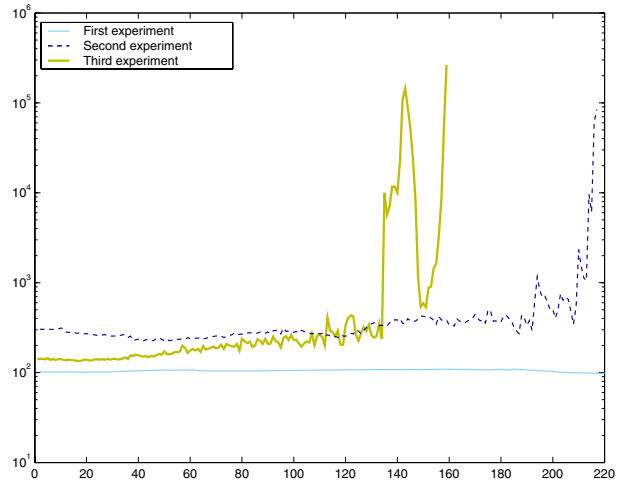
Fig. 5. Condition-number evolution of the estimated interaction matrix during the servo. The matrix is learned from three different trajectories. The first one is a sequencing as done above (first experiment). The second uses the stack of task (control law (**??**)), but all the tasks were activated at the same time at the first iteration (that is to say, no task sequencing). The last one is obtained from a classical visual servoing using a six-DOF task composed of all the visual features (third experiment). The matrix learned from a classical servo has a very large condition number. It increases until the servo becomes impossible. The learning realized from sequencing provides a properly conditioned matrix.

Fig. 4. Results for a sequence of four tasks. Top: Analytic method. Bottom: *Correlation*. The vertical line shows the time instant where a new task is added in the stack. Due to calibration errors, the analytical solution is not able to ensure the stack priorities. When a perturbation appears, it is not corrected until the active task has converged. The online estimation is able to quickly correct the perturbations. The perturbation average is thus much lower. The perturbation amplitudes are also lower.

F. Experiment 4 - calibrated robot

The last experiment was realized with an industrial robot. This robot is a six-DOF eye-in-hand robot with a very low payload/weight ratio. It has position and home sensors and its high repeatability allows to do a very precise calibration. Full, accurate calibration is available and the articular jacobian is no longer an approximation. However, the depth of the target is unknown and must be estimated to compute the correct jacobians. Mis-estimating depth induces scale errors in the interaction matrices.

The experiments consist in a sequencing of the four tasks. The tasks were introduced in the same order as before, at fixed time for a better comparison ($\mathbf{e_g}$ at $t = 0$, $\mathbf{e_\alpha}$ at $t = 150$, $\mathbf{e_Z}$ at $t = 60$, and $\mathbf{e_R}$ at $t = 110$). For each experiment, we vary the method used to compute the interaction matrix. The evolution of the perturbation norm is shown on Fig. 6 for each estimation scheme. The complete results are detailed for the *Broyden* method in Fig. 7.

The first test ("current") shows that a perfect behavior is obtained when all the required knowledge is available. Depth was estimated with a pose computation algorithm, using the object geometric model and camera calibration. In the second trial ("misestimated"), depth was mis-estimated by a factor of 2. In the third experiment ("desired"), the interaction matrices were computed at the desired position using the desired depth. As can be shown on the top of Fig. 6, inaccuracies in the projection operators introduce large perturbations.

In a second set of experiments presented on the bottom of Fig. 6, we analyze the use of estimated jacobian matrices. The first two trials use both the online and offline versions of the *Correlation*. The third one starts with the analytical matrix, that is updated with the *Broyden* rules. These methods are compared with the same analytic version ("curent") as before.

Figure 6 shows the perturbations for each estimation scheme. As expected, the use of the "perfect" analytical solution leads to better results than the versions with estimated jacobians. However, the use of an estimation of the jacobian matrix (such as the matrix computed at the desired value, or from a misestimated depth parameter) produces large perturbations during a long time. It is also true when used the results of some offline learning. On the opposite, the use of online estimations always outperforms approximations the analytical jacobian. The best behavior is obtained with the *Broyden* algorithm, using the true interaction matrix as initialization. It provides a robust and fast online estimation without the need of a first offline learning.

Finally, Fig. 7 shows the results using the *Broyden* method. We can note that the perturbation amplitudes are very small (almost invisible) for all tasks. They are also quickly reduced.
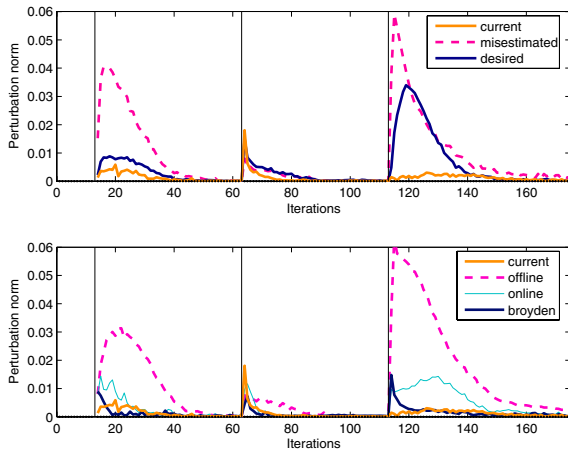
Fig. 6. Perturbation norm (i.e. norm of the error of the tasks already completed) during the positioning. Top: results obtained with the true jacobian ("current") and two approximate versions where the depth is misestimated by a factor of 2 or set to the desired value in the goal position. Bottom: comparison of the true jacobian version ("current") with learned jacobians (Correlation offline and online) and Broyden. Approximations in the analytic solution (on the top) can generate disturbance that are reduced using a learning scheme (on the bottom).
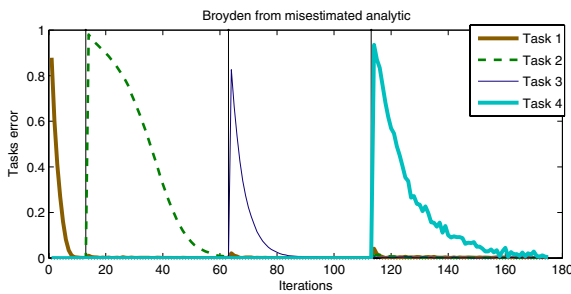

Fig. 7. Correcting a depth mis-estimation by an online estimation of the interaction matrix. Results for a sequence of four tasks sequencing on the full-calibrated robot Afma6. The jacobian estimation method is *Broyden*, using the analytical solution as an initialization of the learning.

V. CONCLUSION

In this paper several learning methods have been tested for jacobian estimation in task sequencing. The tested methods were *Broyden* method, *Correlation* method and *Direct-inverse* method. The *Direct-inverse* method was adapted to have an online update rule. All methods were able to learn the jacobian and accomplish the sequence of tasks, both in eye-in-hand and eye-to-hand configuration.

Of course, if all the necessary knowledge is present (that is if the system is perfectly modeled and calibrated), the control scheme using the analytical jacobian performs better than any learning-based scheme. However, as soon as some errors or misestimations appear in the model, the performances of the analytical solution become lower while the learning improves the quality of the servo. If the calibration is not accurate, the learning methods perform always better.

When comparing the learning schemes, an online update of the jacobian produces better results than an offline learning when considering the number of iterations required to nullify the perturbation, and also when considering the perturbation amplitude. In terms of comparison, the best estimation was obtained using the *Correlation* method. But it requires a tuning and a training step. Another very interesting solution is to use the analytical approximation of the jacobian to initialize the *Broyden* online learning. This solution is very robust, it does not require any tuning, and can perform all the tasks with the same parameters. Moreover, it does not require any offline learning phase.

It was verified that learning improves the task sequencing by reducing the perturbations. It was also shown that the sequencing helps the learning by estimating a mixture of smaller subtasks successively. The condition number of the obtained jacobian, and the stability of the global system (control+learning) are more satisfactory.

REFERENCES

[1] P. Baerlocher and R. Boulic. An inverse kinematic architecture enforcing an arbitrary number of strict priority levels. *The Visual Computer*, 6(20), 2004.

[2] M. De Mathelin and R. Lozano. Robust adaptive identification of slowly time-varying parameters with bounded disturbances. *Automatica*, 35:1291–1305, 1999.

[3] Z. Dodds, M. Jägersand, G. Hager, and K. Toyama. A hierarchical vision architecture for robotic manipulation tasks. In *Proc. of Int. conf. on Computer Vision Systems*, 1999.

[4] O. Faugeras. *Three-dimensional computer vision: a geometric viewpoint*. MIT Press, Cambridge, Massachusetts, 1993.

[5] R. Fletcher. *Practical Methods of Optimization*. Chichester, 2nd edition, 1987.

[6] K. Hosoda and M. Asada. Versatile visual servoing without knowledge of true jacobian. In *IROS*, pages 186–193, Munchen, Germany, Sep. 1994.

[7] S. Hutchinson, G. Hager, and P. Corke. A tutorial on visual servo control. *IEEE Trans. on Robotics and Automation*, 12(5):651–670, October 1996.

[8] M. Jagersand and R.C. Nelson. On-line estimation of visual-motor models using active vision. In *ARPA Image Understanding Workshop*, pages 677–682, 1996.

[9] J. T. Lapresté, F. Jurie, M. Dhome, and F. Chaumette. An efficient method to compute the inverse jacobian matrix in visual servoing. In *ICRA*, 2004.

[10] A. Liegeois. Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE Trans. on Systems, Man and Cybernetics*, 7(12):868–871, December 1977.

[11] M. Lopes, R. Beira, M. Praça, and J. Santos-Victor. An anthropomorphic robot torso for imitation: design and experiments. In *IROS*, Sendai, Japan, 2004.

[12] M. Lopes, A. Bernardino, and J. Santos-Victor. A developmental roadmap for task learning by imitation in humanoid robots. In Y. Demiris, editor, *AISB - Third Int. Symp. on Imitation in Animals and Artifacts*, Hatfield, Uk, April 2005.

[13] N. Mansard and F. Chaumette. Visual servoing sequencing able to avoid obstacles. In *ICRA*, Barcelona, Spain, April 2005.

[14] E. Marchand and G. Hager. Dynamic sensor planning in visual servoing. In *IROS*, volume 3, pages 1988–1993, Leuven, Belgium, May 1998.

[15] J. A. Piepmeier, B. A. Gumpert, and H. Lipkin. Uncalibrated eye-in-hand visual servoing. In *ICRA*, 2002.

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

[17] B. Siciliano and J-J. Slotine. A general framework for managing multiple tasks in highly redundant robotic systems. In *ICAR'91*, pages 1211–1216, 1991.

[18] O. Tahri and F. Chaumette. Point-based and region-based image moments for visual servoing of planar objects. *IEEE Transactions on Robotics*, 21(6):1116–1127, December 2005.