Robot control from disparate multiple sensors

Ezio Malis *, Guillaume Morel †, François Chaumette ‡

Abstract

In this paper, we propose a simple and efficient control algorithm that combines several sensors in order to realize the positioning task of a robot end-effector. The multiple sensors control has been designed as a part of the task function approach. A particular choice of the task function allows us to simplify the design of the control law and the stability analysis. This global controller is based on the weighted sum of individual task-functions. The approach has been applied to the control with vision and force sensors. In spite of its simplicity, this approach provides satisfactory experimental behaviour. Improvements in the positioning of cumbersome objects have been obtained using cameras observing different parts of a scene. Moreover, peg in hole insertion experiments involving large initial errors have been performed using a 7 axis robot manipulator without any computation of the peg trajectory by combining vision and force sensors.

*E. Malis is with INRIA, 2004, route des Lucioles, 06902 Sophia-Antipolis, France.
†Guillaume Morel is with EDF R&D - SOCM - Chatou - France.
‡F. Chaumette is with IRISA/INRIA, Campus de Beaulieu, 35042 Rennes, France.

1 Introduction

Most applications of advanced robotics require to provide robot manipulators with the ability of working in environments with unknown location and geometry. Thus, external sensory information has to be integrated in the manipulator control. Over the past decades, force/torque sensors, proximity sensors and video cameras have been particularly used to this aim. Early in robot control development, force sensing capabilities have been considered as a crucial issue, as the robot often interacts with its environment. Many researches have been conducted to understand stability properties, bandwidth limitation, and to emphasize the role of dynamics in force control (Eppinger and Seering, 1989), (Khatib, 1987), (Raibert and Craig, 1981), (Lawrence, 1988). On another hand, vision-based control has recently received a growing interest, as the computational power of commercially available computers became compatible with real time visual feedback (Espiau et al., 1992), (Hutchinson et al., 1996), (Wilson et al., 1996). For both vision and force control, initial drawbacks have been over-passed, and a number of techniques are now available. They should be selected depending on the nature of the task, the robot and sensor design, and the low level controller hardware. At a more general level, the *task function approach* has been proposed in (Samson et al., 1991) as a general framework for sensor-based control of robots.

Combining several sensory data is also an important issue that has been studied considering two fundamentally different approaches. In the first one, the different sensors are considered to complementarily measure the same physical phenomena. Thus, a sensory data fusion strategy is used to extract a pertinent information from the multiple sensory data. The second control approach consists of selecting, among the available sensory signals, a set of pertinent data, which is then servoed. The two approaches will be referred afterwards as *sensory data fusion* and *sensory data selection*.

A typical example of sensory data fusion is stereo vision. With this approach, two images provided by two distinct cameras are used to extract a complete Euclidean information on the observed scene. The fused data can then be used as a measure in a feedback loop, such as in (Allen et al., 1993), where a stereo pair is used to reconstruct the position of a moving object at video rate, and to control the robot's end-effector in order to track and pick the object. Such a 3D visual servoing technique, was also proposed in (Cipolla and Hollinghurst, 1997), where a five d.o.f. robot is controlled using a fixed external stereo head. Data fusion can also be performed in projective space without explicitely reconstructing the position of observed objects. For example, in (Hager, 1997) a stereo system simultaneously tracks the robot end-effector and visual features used to define goal positions. Thus, the error is defined as a function of features directly observed from both cameras and

the robot can be positioned with an accuracy that is independent of errors in hand-eye calibration. Another method, based on the estimation of a (4×4) projective homography matrix which relates two sets of points of the projective space, is also proposed in (Ruf and Horaud, 1999). This method is similar to the previous ones, although the robot is then controlled without doing any Euclidean reconstruction. Sensory data fusion can also be achieved directly by the mean of control, as in (Maru et al., 1993), where 2D visual servoing is performed using a stereo head, while updating the task Jacobian from the information given by the two cameras. Disparate sensory data fusion can also be used in robot control. For example, in (Agapakis, 1990), a proximity sensor is used in combination to a camera for depth reconstruction purposes.

On the other hand, sensory data selection is used when all the different data do not provide the same quality of information. In this case one can use environment models in order to select the appropriate sensor and to switch control between sensor. A typical example of this approach was developed in (Nelson and Khosla, 1996) and (Nelson et al., 1996) through the resolvability concept, which provides a measure of the ability of a sensor to resolve motion. The authors consider an application involving force and vision combination: the resolvability of the two sensors is evaluated on line and the controller automatically switches between force and vision control during a contact

task. A similar idea is developed in (Hosoda et al., 1996) by applying classical hybrid position/force control : the force is servoed along the direction that are mechanically constrained by the environment (e.g. the normal direction of a contact plane) while the image is used to servo robot's motion along the remaining free directions. In this case, the sensory signal selection is operated in advance.

Our approach of multi sensory robot control is drastically different from these two methodologies (Morel et al., 1998) (Malis et al., 2000). It does not pertain to sensory data fusion because we assume that the sensors may observe different physical phenomena from which extracting a single fused information does not make sense. It neither pertains to sensory data selection because we consider potential situations for which it is not possible to select a set of data that would be more pertinent than others. Consequently, the proposed approach addresses a very large spectrum of potential applications, for which the sensory equipment may be disparate and complex. As an improvement over previous approaches, there is no need to provide a model of the environment that would be required to design a switching or fusion strategy.

Consider an example task in nuclear maintenance operations, consisting of placing the nozzle dam in steam generators (see Figure 1). This object is roughly a 1 meter diameter disk, and has to overlap a hole where it precisely

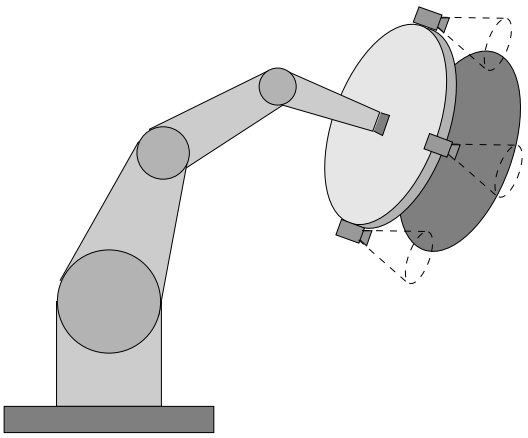


Figure 1: Cover positioning in a steam generator

fits. Since, in the typical maintenance scenarios, the exact location of the hole with respect to the robot is not precisely known, a vision feedback is necessary for an automatic operation. Due to the large size of the object, a single camera observing only one side of the scene is not appropriate, as it could produce a "lever effect". A small positioning error of the camera could produce a larger error on the other side of the cover. Using more cameras is thus desirable in order to observe different parts of the scene. They can be placed on the robot around the cover to observe the edges of the hole, as illustrated in Figure 1. In this case, the fields of view of the cameras do not necessarily overlap, which fundamentally differs from conventional stereo vision. Furthermore, in the final phase of the task, calibration errors in the vision system may cause small displacements from the nominal insertion trajectory. This would involve large forces. A wrist force/torque sensor

can then be used in conjunction with the cameras to perform the placement task while minimising interaction forces. It must be noticed that combining force and vision has been rarely considered and, to our knowledge, only few references are available in the literature (Hosoda et al., 1996) (Nelson and Khosla, 1996) (Nelson et al., 1996) (Morel et al., 1998).

The main contribution of our work is the definition of a global task function as the weighted sum of individual task-functions (one for each sensor). Such a hierarchical structure leads to an increased modularity. A task function for each sub-system is designed and the sensor combination is done at a higher level.

The paper is organized as follows. In the next section, a general formulation of disparate multi sensor control problem is given, based on the task-function concept (Samson et al., 1991). The general control law and the stability analysis of the controller are described in Section 3. In Section 4 we detail the design of the force and vision control sub-systems. Finally, experiments are presented in Section 5.

2 Modeling

In this section, we describe the general model used to control the robot motion from disparate sensory feedback. The robot is supposed to be controlled

by a six dimensional vector $\mathbf{v}_r$ representing the end-effector velocity, whose components are supposed to be expressed in the end-effector frame. It is equipped with N sensors, each of them being rigidly attached to the end-effector. Each sensor $S_i,\ i = 1 \cdots N$ provides an n_i dimensional vector signal $\mathbf{s}_i$. For simplicity, it is assumed in the next that $n_i \geq 6$. However, this hypothesis is not restrictive and our work can be easily extended to the general case. An interaction matrix $\mathbf{L}_i,\ \{i = 1, 2, .., N\}$ is attached to each sensor, such that (Espiau et al., 1992):

$$\dot{\mathbf{s}}_i = \mathbf{L}_i \mathbf{v}_i \tag{1}$$

where $\mathbf{v}_i$ is the absolute velocity of the sensor S_i, expressed in a frame $\mathcal{F}_i$ attached to the sensor.where m is the number of manipulator's degrees of freedom (generally $m = 6$) Furthermore, we introduce the transformation matrix $\mathbf{W}_i$ linking the sensor velocity and the end-effector velocity:

$$\mathbf{v}_i = \mathbf{W}_i \mathbf{v}_r \tag{2}$$

Since the sensor is fixed with respect to the end-effector, the matrix $\mathbf{W}_i$ is constant, and depends on the geometric transform between the end-effector frame and the sensor frame. Using the two previous equations we naturally

get $\dot{\mathbf{s}}_i = \mathbf{L}_i\mathbf{W}_i\mathbf{v}_r$, where $\mathbf{L}_i\mathbf{W}_i$ is called the Jacobian matrix of the i^{th} sensor. Let $\mathbf{s} = \left[\mathbf{s}_1^T \ \mathbf{s}_2^T \ \cdots \mathbf{s}_N^T\right]^T$ be the n dimensional vector (where $n = \sum_{i=1}^{N} n_i$) containing all the signals provided by the N sensors. The relationship between the time derivative of the global sensor signal vector and the end-effector velocity $\mathbf{v}_r$ is:

$$\dot{\mathbf{s}} = \begin{bmatrix} \mathbf{L}_1 & 0 & \dots & 0 \\ 0 & \mathbf{L}_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{L}_N \end{bmatrix} \begin{bmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \\ \vdots \\ \mathbf{W}_N \end{bmatrix} \mathbf{v}_r = \mathbf{L}\mathbf{W}\mathbf{v}_r \tag{3}$$

Now, let $\mathbf{s}^*$ be the desired value of the sensor signal vector $\mathbf{s}$. We use a task function of the form $\mathbf{e} = \mathbf{C}\left(\mathbf{s} - \mathbf{s}^*\right)$ (Espiau et al., 1992), where $\mathbf{C}$ is a full rank constant matrix of dimensions $(6\times n)$, which allows to take into account the information redundancy. The matrix $\mathbf{C}$ being constant, the time derivative of the task function is:

$$\dot{\mathbf{e}} = \mathbf{C}\dot{\mathbf{s}} = \mathbf{C}\mathbf{L}\mathbf{W}\mathbf{v}_r \tag{4}$$

A major concern in designing a task function based controller is to select a suitable constant matrix $\mathbf{C}$, while ensuring that the matrix $\mathbf{CLW}$ has full rank. Usually, $\mathbf{C}$ is designed as a function of the matrices $\mathbf{L}$ and $\mathbf{W}$,

which both depend on unknown parameters. Thus, estimated matrices $\widehat{\mathbf{L}}$ and $\widehat{\mathbf{W}}$ will be used. Different choices of $\mathbf{C}$ are possible. Mainly two solutions involving a constant matrix $\mathbf{C}$ are used in the literature:

- $\mathbf{C} = (\widehat{\mathbf{L}}^* \widehat{\mathbf{W}})^+$ is the pseudo-inverse of the Jacobian matrix calculated at the desired position $\widehat{\mathbf{L}}^* = \mathbf{L}(\mathbf{s}^*, \mathbf{g}^*)$ (where $\mathbf{g}$ are unknown geometric parameters). This choice ensures a better decoupling of the control law near the convergence.

- $\mathbf{C} = (\widehat{\mathbf{L}}^* \widehat{\mathbf{W}})^T$ is the transpose of the Jacobian matrix calculated at the desired position. This choice does not need the inversion of the interaction matrix, but it does not realize the decoupling of the control law near the convergence.

In this paper, we propose to use the block pseudo-inverse of the Jacobian matrix:

$$\mathbf{C} = (\widehat{\mathbf{L}}^* \widehat{\mathbf{W}})^\sharp = \left[\begin{array}{ccc} \kappa_1 \widehat{\mathbf{W}}_1^{-1} \widehat{\mathbf{L}}_1^{*+} & \ldots & \kappa_N \widehat{\mathbf{W}}_N^{-1} \widehat{\mathbf{L}}_N^{*+} \end{array} \right] \tag{5}$$

where $\widehat{\mathbf{L}}_i^{*+}$ is the pseudo-inverse of $\widehat{\mathbf{L}}_i^*$, $\widehat{\mathbf{W}}_i^{-1}$ is the inverse of $\widehat{\mathbf{W}}_i$ and κ_i is a positive weighting factor such that $\sum_{i=1}^{N} \kappa_i = 1$. This choice presents intermediate characteristics with respect to the precedent ones.

The interest of using this new combination matrix for disparate multiple sensory feedback will be shown when computing the control law and analysing

the stability of the system. However, it can already be noticed that it presents some interesting characteristics. Indeed, if we consider for each sensor a task function $\mathbf{e}_i = \mathbf{C}_i (\mathbf{s}_i - \mathbf{s}_i^*)$ with $\mathbf{C}_i = \widehat{\mathbf{W}}_i^{-1} \hat{\mathbf{L}}_i^{*+}$, then the task function of the entire system is a weighted sum of the task functions relative to each sensor:

$$\mathbf{e} = \mathbf{C} (\mathbf{s} - \mathbf{s}^*) = \sum_{i=1}^{N} \kappa_i \mathbf{e}_i = \sum_{i=1}^{N} \kappa_i \mathbf{C_i} (\mathbf{s_i} - \mathbf{s_i^*}) \tag{6}$$

Such a choice of the matrix $\mathbf{C}$ leads to an increased modularity. An individual task function is designed for each sub-system and the sensor combination is done at a higher level. The design of the multi sensor combination simply consists of selecting the positive weights κ_i, $i = 1 \cdots N$. This choice is both task and sensor dependent. The weights κ_i can be set according to the relative precision of the sensors, or more generally to balance the velocity contribution of each sensor: if a sensor S_i plays an important safety role (*e.g.* a force sensor) it will be affected a large weight. A dynamical setting of κ_i can also be implemented. For example, if a sensor S_i fails during a task, then the weight κ_i will be set to zero.

3 Control law and stability analysis

If a simple proportional control law is used by imposing the exponential decreasing of the task function ($\dot{\mathbf{e}} = -\lambda\mathbf{e}$, λ being a positive scalar), the ideally decoupling control law is:

$$\mathbf{v}_r = -\lambda(\mathbf{CLW})^{-1}\mathbf{e} \tag{7}$$

A more realistic control law is the following:

$$\mathbf{v}_r = -\lambda(\mathbf{C}\widehat{\mathbf{L}}\widehat{\mathbf{W}})^{-1}\widehat{\mathbf{e}} \tag{8}$$

In general, we choose to use in the control law the interaction matrix calculated at the desired position: $\widehat{\mathbf{L}} = \widehat{\mathbf{L}}^*(\mathbf{s}^*, \widehat{\mathbf{g}}^*)$. Consequently, choosing $\mathbf{C} = (\widehat{\mathbf{L}}^*\widehat{\mathbf{W}})^\sharp$ defined by equation (5), we get:

$$(\mathbf{C}\widehat{\mathbf{L}}^*\widehat{\mathbf{W}})^{-1} = (\sum_{i=1}^{N} \kappa_i \widehat{\mathbf{W}}_i^{-1}\widehat{\mathbf{L}}_i^{*+}\widehat{\mathbf{L}}_i^*\widehat{\mathbf{W}}_i)^+ = \sum_{i=1}^{N} \kappa_i \mathbf{I}_m = \mathbf{I}_m \tag{9}$$

and the control law is:

$$\mathbf{v}_r = -\lambda\widehat{\mathbf{e}} \tag{10}$$

Similar results can be obtained with different choices of $\widehat{\mathbf{L}}$ as we will see in the case of the force sub-system. Considering equation (6), this control

law is equivalent to a weighted sum of the control laws of each sub-system $\mathbf{v}_{ri} = -\lambda \widehat{\mathbf{e}}_i$:

$$\mathbf{v}_r = -\lambda \sum_{i=1}^{N} \kappa_i \widehat{\mathbf{e}}_i = \sum_{i=1}^{N} \kappa_i \mathbf{v}_{ri} \tag{11}$$

The general structure of this controller is given in Figure 2. In order to

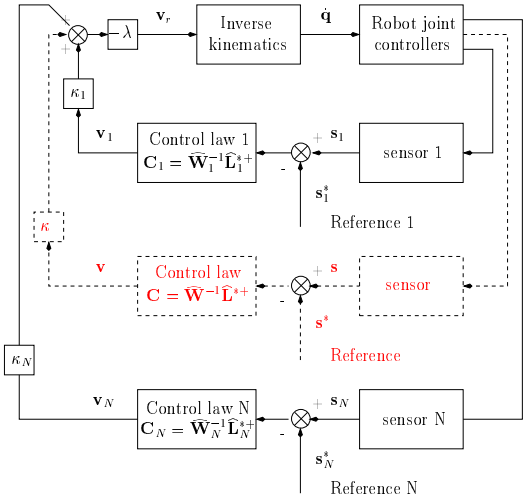


Figure 2: General structure of the controller with N sensors

obtain the closed-loop system equation, the estimated task function must be expressed as a function of the real task function and of the calibration errors of the system. In the general case, when the task function is built directly from sensor data, it is simply $\widehat{\mathbf{e}} = \mathbf{e}$ if the measurement noise is neglected. When the task function is built using parameters reconstructed from the sensor data, a further modeling step should be done as we will see in the next section. If $\widehat{\mathbf{e}} = \mathbf{e}$, the closed-loop system with the control law (10) is

given by:

$$\dot{\mathbf{e}} = -\lambda(\mathbf{CLW})(\mathbf{C}\widehat{\mathbf{L}}^{*}\widehat{\mathbf{W}})^{-1}\mathbf{e} = -\lambda(\widehat{\mathbf{L}}^{*}\widehat{\mathbf{W}})^{\sharp}(\mathbf{LW})\mathbf{e} \tag{12}$$

A sufficient condition for the stability of the system is:

$$(\widehat{\mathbf{L}}^{*}\widehat{\mathbf{W}})^{\sharp}\mathbf{LW} > 0 \tag{13}$$

It is clear now why it is interesting to choose $\mathbf{C}$ as a block pseudo-inverse. Indeed, using equation (5), the condition (13) can be also written as:

$$(\widehat{\mathbf{L}}^{*}\widehat{\mathbf{W}})^{\sharp}\mathbf{LW} = \sum_{i=1}^{N} \kappa_i \widehat{\mathbf{W}}_i^{-1}\widehat{\mathbf{L}}_i^{*+}\mathbf{L}_i\mathbf{W}_i > 0 \tag{14}$$

and this condition is verified if (remember that $\kappa_i > 0$):

$$\widehat{\mathbf{W}}_i^{-1}\widehat{\mathbf{L}}_i^{*+}\mathbf{L}_i\mathbf{W}_i > 0 \qquad \forall i = \{1, 2, 3, ..., N\} \tag{15}$$

The whole system will be thus stable if each sub-system is stable since the sum of positive matrices is a positive matrix. The stability analysis can thus be reduced to the study of the stability of each sub-system separately. Even if the system is not unstable (i.e. $\|\widehat{\mathbf{e}}\|$ is bounded and cannot grow indefinitely) it could happen that $\widehat{\mathbf{e}} = 0$ but $\widehat{\mathbf{e}}_i \neq 0 \forall i$. In this case, the control output will be $\widehat{\mathbf{v}} = 0$ which corresponds to a local minimum. However, that situation

can be easily detected and one can choose a new set of gains κ_i such that $\widehat{\mathbf{e}} \neq 0$.

4 Sub-systems design

In this section we discuss the design of each separate sub-system. The force and vision control sub-systems described in this section can be then combined in the general control scheme described previously.

4.1 Vision sub-system

Vision-based control can be achieved using mainly three different methods. In a 3D visual servoing system, the error to be controlled corresponds to the camera's pose, that is its position and orientation (Wilson et al., 1996). The pose relative to the target is estimated from image features, which requires the precise knowledge of the target geometry. Conversely, 2D visual servoing exploits an error directly computed from the image features, relatively to their desired values (Espiau et al., 1992). Finally, in the $2\frac{1}{2}$D visual servoing approach, the error to be controlled is computed in part in the Cartesian space and in part directly in the image (Malis et al., 1999) (Morel et al., 1999). Any of the three methods could be used in our framework. However, the 3D servoing approach will not be considered here.

4.1.1 2D visual servoing

In the implemented image-based controller, the considered image features are the image coordinates of p points observed on the target $\mathbf{s} = \begin{bmatrix} x_1 & y_1 & \dots & x_p & y_p \end{bmatrix}^T$. The task function is $\mathbf{e} = \mathbf{C}(\mathbf{s} - \mathbf{s}^*)$. Thus, the Jacobian matrix of the sensor signal is:

$$\frac{\partial \mathbf{s}}{\partial \mathbf{r}} = \mathbf{L}(\mathbf{z}, \mathbf{s})\mathbf{W} \tag{16}$$

The interaction matrix $\mathbf{L}$ depends on the distance $\mathbf{z}$ of the points of the target to the camera. When an object is in the field of view of at least two cameras (stereo configuration), the distance can be estimated. Otherwise, it can be approximated for each camera, exactly as for the mono-camera visual servoing (Espiau et al., 1992), when the desired features in the images $\mathbf{s}^*$ are stored. The optimal choice for the combination matrix is $\mathbf{C} = \widehat{\mathbf{W}}^{-1}\widehat{\mathbf{L}}^{+}(\widehat{\mathbf{z}}^*, \widehat{\mathbf{s}}^*)$. The vision system is stable if:

$$\widehat{\mathbf{W}}^{-1}\widehat{\mathbf{L}}^{+}(\widehat{\mathbf{z}}^*, \widehat{\mathbf{s}}^*)\mathbf{L}(\mathbf{z}, \mathbf{s})\mathbf{W} > 0 \tag{17}$$

Unfortunately, the stability domain of the 2D visual servoing is not analytically known to this date and the analysis has to be stopped. However, a 2D mono-camera visual servoing system is known to be robust to calibration errors (Espiau, 1993). The same robustness can thus be expected from a

multi-cameras system using a 2D visual servoing technique.

4.1.2 $2\frac{1}{2}$D visual servoing

In the case of the $2\frac{1}{2}$D visual servoing the task function is reconstructed from the sensor signal (Malis et al., 1999). The Jacobian matrix is an upper block-triangular matrix which is non singular in the whole workspace:

$$\frac{\partial \mathbf{s}}{\partial \mathbf{r}} = \mathbf{L}(d^*, \mathbf{s})\mathbf{W} \tag{18}$$

The only unknown parameter is the distance d^* of the camera to the target but it has not a great influence on the stability of the system. In that case, we have $\widehat{\mathbf{e}}_i = \mathbf{E}_i \mathbf{e}_i$ where $\mathbf{E}_i$ depends on the camera internal parameters and is given in (Malis et al., 1999). The estimated task function is thus given by:

$$\widehat{\mathbf{e}} = \sum_{i=1}^{N} \kappa_i \mathbf{E}_i \mathbf{e}_i \tag{19}$$

Then, $\widehat{\mathbf{e}}$ cannot be written easily as a function of $\mathbf{e}$ under the form $\widehat{\mathbf{e}} = \mathbf{E}\,\mathbf{e}$. However, we can proceed in the following way. The task function can be written as:

$$\mathbf{e} = \begin{bmatrix} \kappa_1 \mathbf{I}_6 & \kappa_2 \mathbf{I}_6 & \dots & \kappa_N \mathbf{I}_6 \end{bmatrix} \mathbf{e}' = \mathbf{K}\mathbf{e}' \tag{20}$$

where $\mathbf{e}^{'}$ is the $6N$ dimensional vector containing the task functions of each sub-system. This vector can be computed as :

$$\mathbf{e}^{'} = \mathbf{K}^{+}\mathbf{e} + (\mathbf{I}_{6N} - \mathbf{K}^{+}\mathbf{K})\phi \tag{21}$$

where $(\mathbf{I}_{6N} - \mathbf{K}^{+}\mathbf{K})\phi$ is an arbitrary vector belonging to the null space of $\mathbf{K}$. Thanks to the simple form of $\mathbf{K}$ its pseudo-inverse is $\mathbf{K}^{+} = (\sum_{i=1}^{N} \kappa_i^2)^{-1}\mathbf{K}^{T}$. Plugging equations (21) and (20) in equation (19), the estimated task function is:

$$\widehat{\mathbf{e}} = \frac{1}{\sum_{i=1}^{N} \kappa_i^2} \sum_{i=1}^{N} \kappa_i^2 \mathbf{E}_i \, \mathbf{e} + \mathbf{b}_e = \mathbf{E} \, \mathbf{e} + \mathbf{b}_e \tag{22}$$

where $\mathbf{b}_e$, which can be considered as an additional "bias" on the task function is given by:

$$\mathbf{b}_e = \sum_{i=1}^{N} \kappa_i \mathbf{E}_i \phi_i - \frac{1}{\sum_{i=1}^{N} \kappa_i^2} \sum_{i=1}^{N} \kappa_i^2 \mathbf{E}_i (\sum_{i=1}^{N} \kappa_i \phi_i) \tag{23}$$

This bias can be neglected. Indeed, this hypothesis is justified since, thanks to the very nice form of matrix $\mathbf{E}_i$ (block diagonal), and with an appropriate choice of $\mathbf{e}_i$ for the rotation ($\mathbf{e}_{\omega i} = \widehat{\mathbf{A}}\mathbf{A}^{-1}\mathbf{u}\theta$, where $\mathbf{A}$ is the matrix of the camera parameters, $\widehat{\mathbf{A}}$ its approximation and $\mathbf{u}\theta$ the rotation vector), then $\mathbf{E}_i \approx \mathbf{I}_6$ and $\mathbf{b}_e \approx 0$. Let us remark that at the convergence, $\mathbf{E}_i = \mathbf{I}_6$ then $\mathbf{b}_e = 0$ and the system is always locally stable (Malis, 1998). Further-

more, thanks to the nice form of matrices $\mathbf{L}_i$ and $\mathbf{W}_i$ (block triangular), the stability analysis of the multi-cameras 2D$\frac{1}{2}$ visual servoing is thus reduced to the stability analysis of each sub-system. Since each sub-system is stable under the conditions given in (Malis et al., 1999), the whole system is also stable, at least under the same conditions.

4.2 Force sub-system

In order to design the force sub-system, it is straightforward to choose $\mathbf{s} = \boldsymbol{\tau}$, where $\boldsymbol{\tau}$ is the interaction wrench. The task function is again $\mathbf{e} = \mathbf{C}(\mathbf{s} - \mathbf{s}^*)$. Then the interaction matrix is:

$$\frac{\partial \mathbf{s}}{\partial \mathbf{r}} = \frac{\partial \boldsymbol{\tau}}{\partial \mathbf{r}} \tag{24}$$

$\mathbf{r}$ being the actual end-effector pose. The interaction matrix depends on both the geometry of the constraint and the contact mechanics. Thus, a general stability condition cannot be derived for any task. We give now an example considering a contact point configuration. Thus, only linear forces τ are considered in the example, instead of the complete wrench $\boldsymbol{\tau}$. In this case, the desired damping is a constant spherical matrix, $\mathbf{B} = b\mathbf{I}_3$. If we assume

elastic deformations and no friction, we get [16]:

$$\frac{\partial \tau}{\partial \mathbf{r}} = k\mathbf{n}\mathbf{n}^T \tag{25}$$

where $\mathbf{n}$ is the unitary vector normal to the contact surface and k is the scalar stiffness. Choosing $\mathbf{C} = \mathbf{B}^{-1}$ and the control law $\dot{\mathbf{r}} = -\lambda \mathbf{e}$ the system will be stable if:

$$\mathbf{B}^{-1}\frac{\partial \boldsymbol{\tau}}{\partial \mathbf{r}} > 0 \tag{26}$$

Since the desired damping is a constant spherical matrix, $\mathbf{B}^{-1} = \frac{1}{b}\mathbf{I}_3$, we finally get:

$$\mathbf{B}^{-1}\frac{\partial \boldsymbol{\tau}}{\partial \mathbf{r}} = \frac{k}{b}\mathbf{n}\mathbf{n}^T \tag{27}$$

which is positive for any positive value of b and k. As already mentioned, this example take into account only linear forces. Similarly, one can consider the general case with the complete wrench $\boldsymbol{\tau}$.

It is interesting to notice that we finally get what was referred in the force control literature as a *position based impedance controller* (Heinrichs et al., 1996) (Caccavale et al., 1999), with a desired impedance corresponding to a pure damper.

5 Experimental Results

The experimental setup used in this work is the following. The robot is a 7 axis redundant electric Mitsubishi PA-10 manipulator (at the EDF-DER). The joint redundancy of the robot is solved by minimizing joint velocities in the inverse kinematics procedure using a conventional pseudo inverse algorithm $\dot{\mathbf{q}} = \mathbf{J}^{+}\mathbf{v}_r$. In addition, a second order differential model is used to solve the inverse kinematics in singular joint configurations (Malis et al., 1996). We used Panasonic cameras and an ATI 6 axis force / torque sensor mounted at the end-effector. Low level joint position control is achieved in the Mitsubishi controller, which communicates with a VME bus controller through an Arcnet communication link. Two CPU boards supporting VxWorks realize the position based impedance control. An additive specific EDIXIA vision dedicated board generates the reference velocity.

In order to perform a task, it is assumed that an off-line learning phase has been done prior to the maintenance operation. It provides the final desired sensory signal value, that consists of the images viewed by the different cameras at the final location. If the end-effector positioning requires an operating force to be applied, the desired force signal can be taught to the system during the same learning phase. Finally, the task consists of servoing the different sensory signals towards their final desired value, assuming that

a sufficient information is available at the initial stage of the task, that is at least one camera can see its tracked target. We describe now first experiments realized coupling two cameras and then experiments realized coupling a camera and a force sensor.

5.1 Coupling two cameras

A system simulating the positioning of the cover using only vision has been tested. In order to demonstrate the practicability of the task, only two cameras were used. The cameras are placed at the extremity of a 50 cm aluminium bar, mounted on the manipulator robot as in Figure 3. Two planar targets with six points each were used. A simple hardware architecture was used in order to reduce the cost of the system: the two cameras are connected to two different entries of the same EDIXIA video-board and the acquisitions of the two images is sequential. The rate of such system is two times slower than the rate of a parallel system (i.e. 80 ms, with a delay between the acquisition of the two images of about 40 ms). Consequently, the velocity of the robot end-effector during the servoing could not be too high. Indeed, the delay between the acquisitions of the images is not taken into account in the control law. The convergence speed of the task was set to $\lambda = 0.1$ for all the experiments. Depending on the stability domain which is possible to obtain

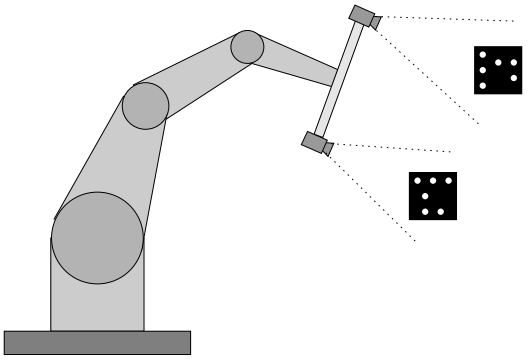


Figure 3: Positioning a cumbersome object

from the control law, two types of algorithms can be considered. When the convergence domain of each sub-system is not large (as for the 2D visual servoing), the following general algorithm is proposed (using N cameras):

1) Test how many cameras have converged (a camera has converged if the error on visual features is lower than ϵ_i: $\|\mathbf{s}_i - \mathbf{s}_i^*\| < \epsilon_i$). If all the cameras have converged the goal is reached, else go to 2).

2) Select all the converged cameras more the nearest to convergence (the nearest camera to convergence is the one minimising $\|\mathbf{s}_j - \mathbf{s}_j^*\|$ under the constraint that $\|\mathbf{s}_j - \mathbf{s}_j^*\| > \epsilon_j$). Go to 3).

3) Do visual servoing using the selected cameras and after their convergence, go to 1).

This iterative algorithm adds a new camera at each step. If the stability domain is very large (as for the $2\frac{1}{2}$D visual servoing), a more satisfactory

algorithm can be considered:

1) Test how many cameras have converged. If all have converged the goal is reached, else go to 2).

2) Select all the converged cameras, more those which have the target in their field of view (and which are in the stability domain). Go to 3).

3) Do visual servoing using the selected cameras and after their convergence go to 1).

If cameras are well distributed around the object, then these two algorithms converge at least in N steps. This implies that:

- a converged camera has also converged at the end of each step. This does not mean that the camera stays at convergence during the servoing. Indeed, only the whole task function decreases.

- If p (with $p < N$) cameras have converged, there exists at least one other "active" camera, i.e. with the target in its field of view and in the stability domain.

5.1.1 Precision comparison between mono and multi-cameras visual servoing

In these experiments, we use image-based visual servoing. The system is robust with respect to the calibration errors since a teaching by showing technique has been used. The system is first shown the target views. The image features are automatically extracted and they are stored. The robot and object are then moved and the system automatically identifies the target and controls the robot end-effector. The visual servoing is stopped when the maximal error on the coordinates of the image points is 1 pixel. Four experiments were carried out, corresponding to four different camera displacements. The first displacement was a pure translation. The second and the third where mainly rotations around the $\vec{x}$ and the $\vec{y}$ axes respectively. Finally, the fourth displacement was a big rotation around the $\vec{x}$ and $\vec{y}$ axes and a big translation. First, the visual servoing is carried out using only one of the two cameras ($\kappa_1 = 0$ and $\kappa_2 = 1$). Then, starting to the same initial position, the servoing is performed using the two cameras with the control law (10) ($\kappa_1 = 0.5$ and $\kappa_2 = 0.5$). In all the experiments, the positioning accuracy (i.e. the error between the reached position and the desired one) is considerably improved using the multi-cameras visual servoing as it is shown in the Table 1 (typically from 2 mm to 0.9 mm). These results demonstrate

the interest of using several cameras to position a cumbersome object.

position	1	2	3	4
translation	1.7 mm - 31 %	2.0 mm - 56 %	1.9 mm - 46 %	2.0 mm - 48 %
rotation	0.3 deg - 36 %	0.4 deg - 58 %	0.3 deg - 27 %	0.4 deg - 44 %

Table 1: Positioning accuracy. The absolute number is the positioning error obtained using one camera. The percentage is the improvement of the positioning using two cameras.

5.1.2 Comparison between 2D and $2\frac{1}{2}$D visual servoing

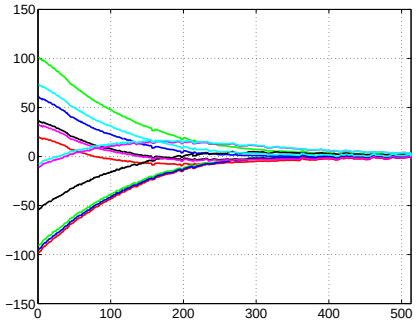
Even if the multi-cameras visual servoing can be used with any scheme (2D, $2\frac{1}{2}$D and 3D) we compare here only the 2D and $2\frac{1}{2}$D methods whose results are illustrated in Figure 4 and 5 respectively. Each figure is divided in two columns, one for each camera. In each column we give respectively the error on the coordinates of the points versus the number of iterations, their trajectory in the image (diamond and circle marks correspond respectively to the initial and final positions). In the last row, we give the control law (i.e. the velocity of the robot end-effector) versus the number of iterations. The error on the coordinates goes to zero and the control law is stable even with the imprecise calibration that was used. Indeed, the camera intrinsic parameters and the transformation matrices between each camera and end-effector frame

were only roughly known.

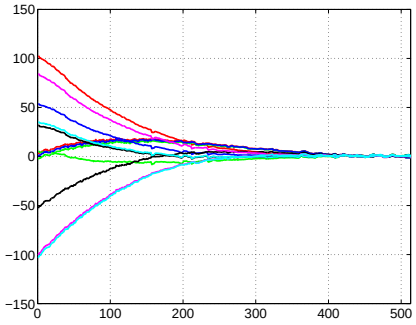
The speed of convergence is relatively slow at the end of the servoing since a simple proportional control law is used. It can be easily improved by simply increasing gain λ as soon as the error becomes small. Obviously, increasing λ too much would lead to an oscillatory behaviour. In our case, λ is chosen small since the rate of the vision system is 80 ms and the velocity of the end-effector must be limited. For both schemes, the trajectories of the points in the images are quite the same (even if the control law is different) since the displacement was not very large. Experiments with bigger displacements were also realized but only one target was initially in the image. In this case, we initially perform the visual servoing using only one camera and once the two targets are in the images, we perform the multi-cameras visual servoing.

5.2 Coupling vision and force sensors

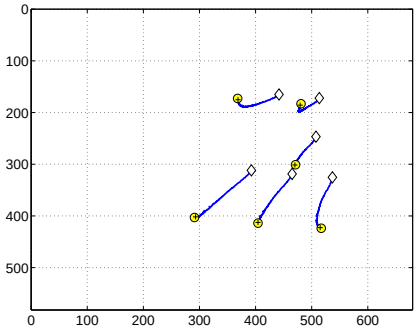
The illustrative task for coupling vision and force sensors is a part of a real nuclear power plant valve maintenance operation. In order to be able to use the different tools involved in this task (see Figure 6(a)), a female interface is mounted on each tool, that matches the end-effector mounted electro-pneumatic male tool changer.



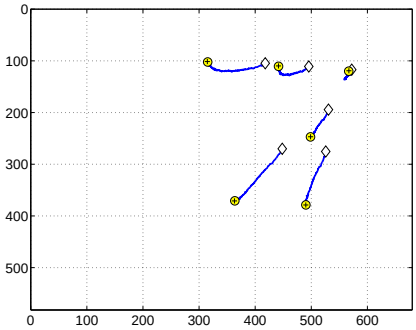
(a) left coordinate errors (pixels vs iteration number)



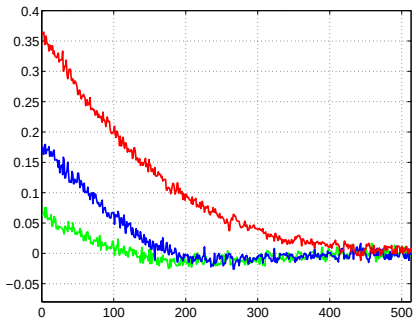
(b) right coordinate errors (pixels vs iteration number)



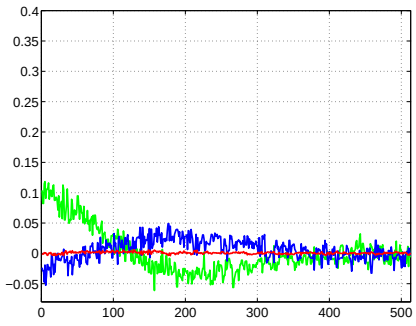
(c) left points trajectory (pixels)



(d) right points trajectory (pixels)

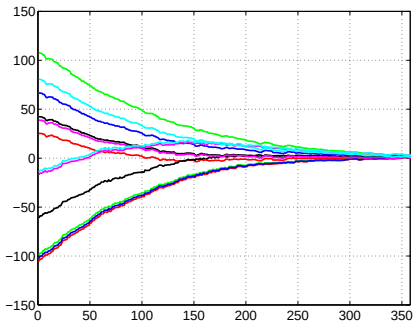


(e) translation velocity (cm/s vs iteration number)

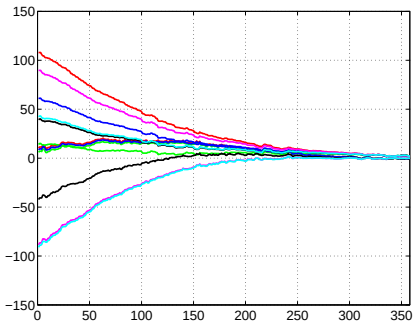


(f) rotation velocity (rad/s vs iteration number)

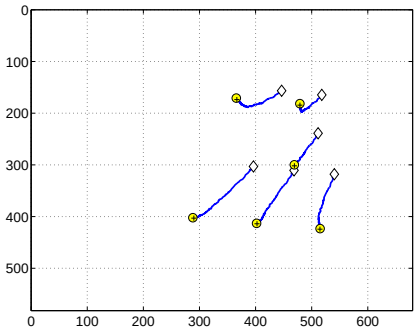
Figure 4: 2D visual servoing results



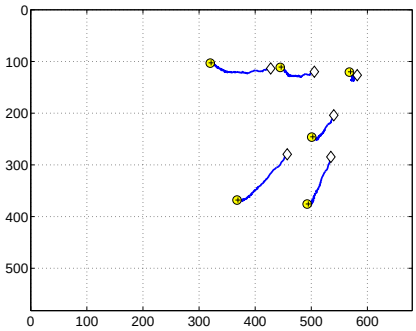
(a) left coordinate errors (pixels vs iteration number)



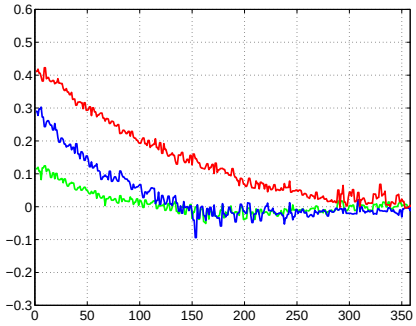
(b) right coordinate errors (pixels vs iteration number)



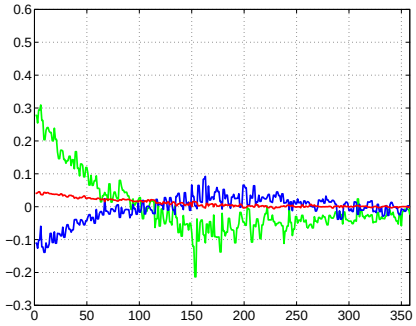
(c) left points trajectory (pixels)



(d) right points trajectory (pixels)

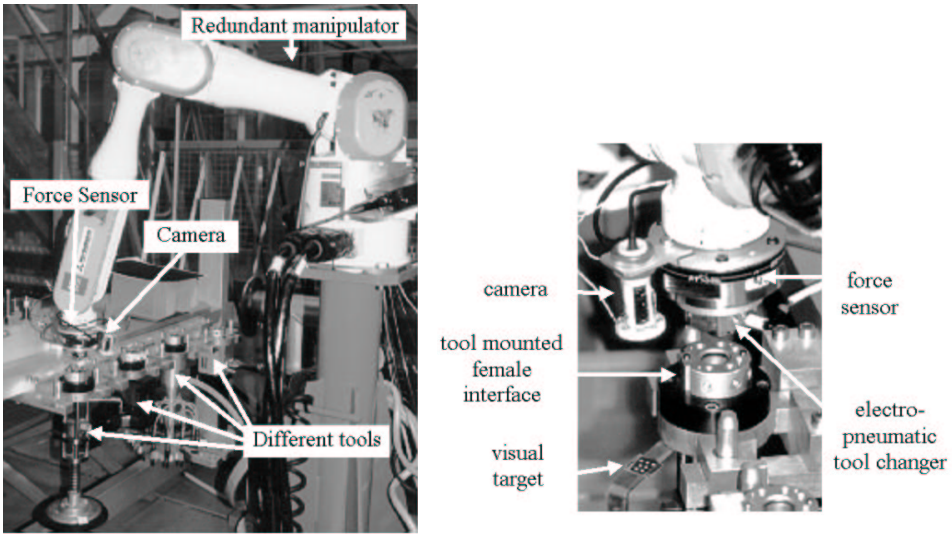


(e) translation velocity (cm/s vs iteration number)



(f) rotation velocity (rad/s vs iteration number)

Figure 5: $2\frac{1}{2}D$ visual servoing results



(a) General view of the task (b) Detailed view of the task

Figure 6: Experimental setup

The clearance is less than a tenth millimeter, and the tool location is unknown. A detailed view of the tool changer mounted on the robot's end-effector is given in Figure 6(b). A system to realize the insertion of a tool changer (which is a task equivalent to the insertion of the cover) using vision and force sensors has been tested. Associated with our controller, a simple programming methodology has been developed. It uses two steps:

- The task is run "manually", that is, only the force sub-system control is running ($\kappa_1 = 1$ and $\kappa_2 = 0$), and the reference trajectory is provided by a device instead of the vision servo controller.

- Once the desired final position is reached, and the force feedback loop

is stabilized to zero ($\tau^* = s^* = 0$), the desired image features are computed and stored.

In order to be able to reach this position through combined visual and force control, one simply sets the desired image features to their memorized value, while the desired force is set to zero. Thus, programming a task as complex as tool changer insertion is extremely simple.

To illustrate the experimental behaviour of our approach, we compare two tool changer insertion attempts. The first one is done with pure visual servoing ($\kappa_1 = 0$ and $\kappa_2 = 1$), while the second one uses combined force and vision control ($\kappa_1 = 0.5$ and $\kappa_2 = 0.5$). The rate of the vision control is 80 ms while the rate of the force control is 10 ms. Thus, the speed of the end-effector is limited by the slow video rate. Improving the vision system will improve the performances of the combined control.

For the experiments shown in this paper, the target impedance is limited to pure damping $\mathbf{B}$ over the six wrench components (i.e. the components of force and torque exerted by the end-effector on objects). The desired wrench is $s^* = 0$. Note that the force sub-system controller is equivalent to an accommodation control (Whitney, 1977). As we anticipated in Section 3, frictions can significantly affect the system behaviour. Consider, for example, that reaching the target requires to move the end effector parallel to the contact

surface. Without friction, the vision system guides the end-effector to the final position while slipping over the surface. However, in the presence of friction, the controller would fail. Close from the final position, the tangential velocity, commanded by the vision-based controller decreases. At some point, it will become smaller than the opposite tangential velocity generated through the force feedback, due to friction component. With a pure damping impedance, the following condition has to be respected to prevent the system from blocking:

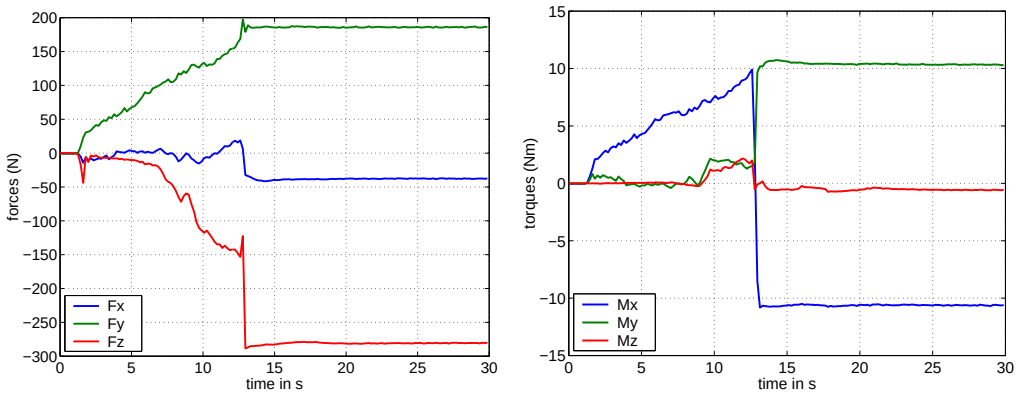
$$|\mathbf{v}_{vision}| > |\mathbf{v}_{force}| = \mathbf{B}|\boldsymbol{\tau}| \tag{28}$$

It is clear that with a proportional visual control law, friction generates static positioning error. A Proportional Integral controller could be used but combining the integral correction with vision feedback and friction non linearities could lead to instability. Furthermore, limit cycles would appear. Rather, a dead zone in the force feedback loop can be used. If the force (in any direction) is smaller than a predetermined limit value, then the velocity commanded by the force loop is zero. This strategy has appeared to be efficient in practice.

5.2.1 Pure visual servoing

In both experiments, the initial position is reached with vision-based control, in order to place the tool changer exactly in front of the female interface. The insertion motion is then a pure translation along the insertion axis, which corresponds to the easiest configuration. In a perfect world, the contact force should stay null. However, the vision-based control does not generate a straight line trajectory. As a matter of fact, the trajectory is generated in order to minimize the image feature error, not the end effector position error. All the geometric modeling errors in both the robot and the sensory system contributes to amplify the deviation from the ideal straight line trajectory. Thus, contact forces appear. Figure 7 shows an example result for pure vision-based insertion. Due to the integral effect of the visual control, combined to the rigidity of the parts to be mated, forces and torques increases rapidly. At $t = 13$ seconds, as the end-effector almost reaches the final position, the electro-pneumatic tool changer is actuated. The sudden change in contact forces is mainly due to the tool changer clamping. Once the tool is grasped, the vision system detects final convergence and the robot stops. One can see that the forces involved during the task are large. Actually, they exceed the maximum value recommended by the robot constructor.

It must be noticed that the control law used for visual servoing is only



(a) Forces (N) (b) Torques (Nm)

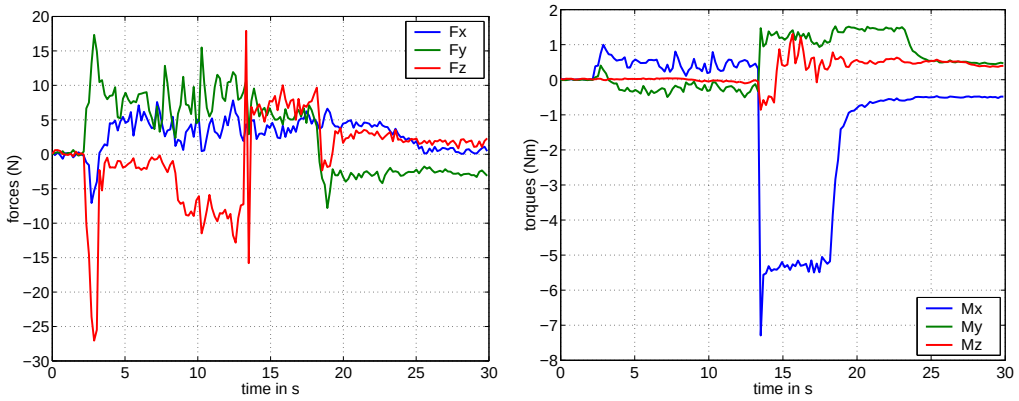
Figure 7: Forces and torques evolution during peg-in-hole insertion with pure visual servo control (Z = insertion axis, X and Y perpendicular plane)

based on the robot kinematics. In order to make our approach work properly at higher speeds (which are however bounded by image acquisition at video rate) it would be necessary to design the vision-based control law by taking into account the dynamics of the manipulator. However, in our experiments the inertial forces are not sufficiently large to be noticeable considering the very slow speeds due to the vision control.

5.2.2 Combined force and vision control

The second experiment combines force and vision control. When the same experiment is performed with combined active compliance, forces and torques remain small. In Figure 8(a), the forces are reduced by a factor 9 (note the change in the coordinates scale). The effects of the tool changer clamping

are now considerably reduced. Similar improvements can be observed in the torques behaviour. In Figure 8(b) the torques are initially reduced by a factor 2. At $t = 13$ seconds, the tool changer is actuated and a sudden step appears on the plot. Contrarily to the previous experiment, the force control allows us to decrease torques by a factor of 9. This experimental result shows



(a) Forces (N) (b) Torques (Nm)

Figure 8: Forces and torques evolution during peg-in-hole insertion with combined visual and force control (Z=insertion axis, X and Y perpendicular plane)

that the force feedback can compensate for the forces undesirably generated by the 2D visual feedback. Additional extensive experiments pointed out that the force and vision combination is not limited to this. In Figure 9, the insertion experiment was realized with a very large initial positioning error, in both position and orientation. It is clear that, since the hole location is completely unknown, force control alone is absolutely not capable of

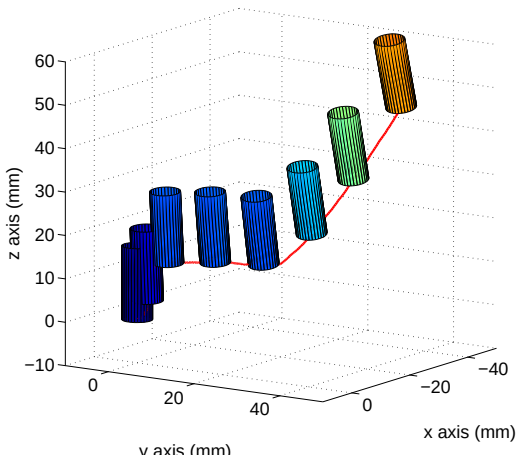


Figure 9: Peg trajectory, reconstructed from experimental data

performing this task. Figure 9 shows that our control scheme is capable of performing low clearance peg-in-hole tasks, with significant initial errors in all the 6 degrees of freedom, and without any knowledge of the hole location, nor on the constraint geometry. No trajectory computation, nor complex insertion strategy is required, as the only input is the final desired image feature vector and the desired force.

6 Conclusion

The control approach proposed in this paper has been designed to take into account several sensors in order to drive the end-effector of a manipulator arm. The particular choice of the task function simplifies the design of the control law and the stability analysis. The stability analysis of the whole system is ensured by the stability of each sub-system. There are obvious

advantages in combining sensors feedbacks in the control of a robot manipulator. Concerning vision sensors, previous work was based on stereo camera configurations. Our paper extends the visual control to several cameras observing different objects or different parts of the same object. The experimental results obtained using two in-hand cameras show that it is possible to improve the positioning accuracy with respect to the use of a single camera. Concerning vision and force sensors, previous work was based on hybrid position / force control, which does not entirely exploit the interest of sensors duality. Our paper has proposed to combine vision and force control within the impedance control approach. The implemented control scheme involves a pure damping position-based impedance control and an external image-based visual controller. It is simple and practical, because the impedance controller and the vision-based controller can be designed separately, as shown by the stability analysis.

Acknowledgements

This work was supported by the national French Company of Electricity Power: EDF. We are grateful to the team manager and the researchers of the DER Chatou, for their participation and help.

References

Agapakis, J. E. (1990). Approaches for recognition and interpretation of workpiece surface using structured lighting. *Int. Journal of Robotics research*, 9(5):3–16.

Allen, P., Timcenko, A., Yoshimi, B., and Michelman, P. (1993). Hand-eye coordination for robotic tracking and grasping. In Hashimoto, K., editor, *Visual servoing*, vol. 7 of *World Scientific Series in Robotics and Automated Systems*, pp. 33–69. World Scientific Press.

Caccavale, F., Natale, C., Siciliano, B., and Villani, L. (1999). 6-dof impedance control based on angle/axis representation. *IEEE Trans. on Robotics and Automation*, 15(2):289–300.

Cipolla, R. and Hollinghurst, N. (1997). Visually guided grasping in unstructured environnement. *Robotics and Autonomous Systems*, 19:337–346.

Eppinger, S. D. and Seering, W. P. (1989). Three dynamic problems in robot force control. In *IEEE Int. Conf. on Robotics and Automation*, pp. 392–397, Scottsdale, Arizona.

Espiau, B. (1993). Effect of camera calibration errors on visual servoing in robotics. In *3rd International Symposium on Experimental Robotics*, Kyoto, Japan.

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

Hager, G. (1997). A modular system for robust positioning using feedback from stereo vision. *IEEE Trans. on Robotics and Automation*, 13(4):582–595.

Heinrichs, B., Sepehri, N., and Thornton-Trump, A. (1996). Position-based impedance control of an industrial hydraulic manipulator. In *IEEE Int. Conf. on Robotics and Automation*, vol. 1, pp. 284–290, Minneapolis, MN.

Hosoda, K., K., I., and Asada, M. (1996). Adaptive hybrid visual servoing / force control in unknown environment. In *IEEE Int. Conf. on Intelligent Robots and Systems*, pp. 1097–1103, Osaka, Japan.

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

Khatib, O. (1987). A unified approach for motion and force control : the operational space formulation. *IEEE Trans. on Robotics and Automation*, 3(3):43–53.

Lawrence, D. (1988). Impedance control stability properties in common implementations. In *IEEE Int. Conf. on Robotics and Automation*, pp. 1185–1192, Philadelphie, Pennsylvannie.

Malis, E. (1998). *Contributions à la modélisation et à la commande en asservissement visuel.* PhD thesis, Université de Rennes I, IRISA.

Malis, E., Chaumette, F., and Boudet, S. (1999). 2 1/2 d visual servoing. *IEEE Trans. on Robotics and Automation*, 15(2):234–246.

Malis, E., Chaumette, F., and Boudet, S. (2000). Multi-cameras visual servoing. In *IEEE Int. Conf. on Robotics and Automation*, vol. 4, pp. 3183–3188, San Francisco, USA.

Malis, E., Morin, L., and Boudet, S. (1996). Control of redundant robots at singularities in degenerate directions. In *12th CISM-IFToMM Symp. on Theory and Practice of Robots and Manipulators*, pp. 319–326.

Maru, N., H. Kase, S. Y., Nishikawa, A., and Miyazaki, F. (1993). Manipulator control by visual servoing with the stereo vision. In *IEEE Int. Conf. on Intelligent Robots and Systems*, vol. 3, pp. 1866–1870, Yokohama, Japan.

Morel, G., Malis, E., and Boudet, S. (1998). Impedance based combination of visual and force control. In *IEEE Int. Conf. on Robotics and Automation*, vol. 2, pp. 1743–1748, Luvein, Belgium.

Morel, G., Szewczyk, J., Boudet, S., and Pot, J. (1999). Explicit incorporation of 2d constraints in vision based control of robot manipulators. In *Proc. ISER'99 : Experimental Robotics IV*, pp. 99–108, Sidney, Australia.

Nelson, B. J. and Khosla, P. K. (1996). Force and vision resolvability for assimilating disparate sensory feedback. *IEEE Trans. on Robotics and Automation*, 12(5):714–731.

Nelson, B. J., Morrow, J. D., and Khosla, P. K. (1996). Robotic manipulation using high bandwidth force and vision feedback. *Mathl. Comput. Modelling*, 24(5/6):11–29.

Raibert, M. H. and Craig, J. J. (1981). Hybrid position/force control of manipulators. *ASME Journal of Dynamic Systems, Measurement and Control*, 102:126–133.

Ruf, A. and Horaud, R. (1999). Rigid and articulated motion seen with an uncalibrated stereo rig. In *IEEE Int. Conf. on Computer Vision*, Corfou, Greece.

Samson, C., Le Borgne, M., and Espiau, B. (1991). *Robot Control: the Task Function Approach*, vol. 22 of *Oxford Engineering Science Series*. Clarendon Press, Oxford, UK.

Whitney, D. E. (1977). Force feedback control of manipulator fine motions. *Journal of Dynamic Systems, Measurement and Control*, pp. 91–97.

Wilson, W. J., Hulls, C. C. W., and Bell, G. S. (1996). Relative end-effector control using cartesian position-based visual servoing. *IEEE Trans. on Robotics and Automation*, 12(5):684–696.