A new sensor self-calibration framework from velocity measurements

Olivier Kermorgant, David Folio and François Chaumette

***Abstract*— In this paper we propose a new on-line sensor self-calibration framework. The approach is to consider the sensor/robot interaction that links the sensor signal variations to the robot velocity. By on-line calibration, we mean only the actual measurements are used to perform calibration under the condition that the interaction matrix is analytically known. This allows us to propose a very simple and versatile formulation of sensor parameter calibration. Various sensors can be considered, and calibration from different sensory data may be considered within the same process. Intrinsic and extrinsic parameters estimation are formulated as a non-linear minimization problem the jacobian of which can be expressed analytically from the sensor model. Simulations and experiments are presented for a camera observing four points, showing good results in the case of separated intrinsic and extrinsic calibration, and illustrating the possible limitations in the case of simultaneous estimation.**

***Index Terms*— self-calibration, visual servoing**

I. Introduction

Sensors calibration is a fundamental problem in many robotic research fields. Without proper calibration, sensor devices produce data that may not be useful or can even be misleading or meaningless. Hence, to maintain an up-to-date model, the calibration has to be repeated at regular intervals. In this work, we focus on on-line self-calibration approach, which means we assume the robot is in its environment, with sensors acquiring information that are not made for the calibration in the first time.

The literature provides many works related to the calibration problem. For example, Borenstein and Feng [1] introduce near-optimal calibration methods for reducing odometry errors in mobile robotics. Recently, an Extended Kalman Filter has been introduced to simultaneously estimate the robotic system configuration and the parameters characterizing the systematic error of sensor (i.e., to solve the Simultaneous Localization and Auto Calibration (SLAC) problem) [8], [12]. Another way to address the problem is to consider it as a mobile sensor network. For instance, in [10] a robot with calibrated sensors gathers samples within the sensor field to allow already deployed static sensors to be calibrated/re-calibrated as required.

Furthermore, since one most used sensor in robotics is vision, camera self-calibration has been widely investigated [14], [9]. For example, in [14] the authors introduce a camera calibration method using epipolar transformation. The camera is calibrated simply by evolving in the environment, selecting and tracking points of interest in the image as the camera moves.

We propose in this work a new on-line sensor self-calibration framework using the sensor/robot interaction. We first present in Section II some preliminaries and introduce the sensor/robot motion model. We describe in Sections III, IV and V our on-line parameters estimation method. In Section VI, we describe an application based on a camera, and present some simulation and experimental results.

II. Sensor-Based Modeling

We consider a sensor mounted on a robotic system and we assume the robot velocity can be measured. The sensor motion can be characterized by its instantaneous velocity $\mathbf{v_s} = (\boldsymbol{v_s}, \boldsymbol{\omega_s})$, where $\boldsymbol{v_s} = (v_x, v_y, v_z)$ and $\boldsymbol{\omega_s} = (\omega_x, \omega_y, \omega_z)$ represent the translational and rotational velocity of the sensor frame $\mathcal{F}_\mathbf{s}$ (Fig. 1). The sensor motion $\mathbf{v_s}$ is related to the robot motion $\mathbf{v}_e$ by:

$$\mathbf{v_s} = {}^\mathbf{s}\mathbf{W}_e \mathbf{v}_e \tag{1}$$

where ${}^\mathbf{s}\mathbf{W}_e$ allows to transform the velocity screw from the sensor frame $\mathcal{F}_\mathbf{s}$ to the end-effector frame $\mathcal{F}_e$. It is given by:

$${}^\mathbf{s}\mathbf{W}_{e(\boldsymbol{\xi}_{\text{Ex}})} = \left[\begin{array}{c|c} {}^\mathbf{s}\mathbf{R}_e & [{}^\mathbf{s}\mathbf{t}_e]_\times \, {}^\mathbf{s}\mathbf{R}_e \\ \hline \mathbf{0}_{3\times 3} & {}^\mathbf{s}\mathbf{R}_e \end{array}\right] \tag{2}$$

where ${}^\mathbf{s}\mathbf{R}_e \in SO(3)$ and ${}^\mathbf{s}\mathbf{t}_e \in \mathbb{R}^3$ are respectively the rotation and the translation between $\mathcal{F}_e$ and $\mathcal{F}_\mathbf{s}$. $[{}^\mathbf{s}\mathbf{t}_e]_\times$ is the 3×3 skew-symmetric matrix related to ${}^\mathbf{s}\mathbf{t}_e$. Using the θu representation for rotation, 6 extrinsic parameters can be represented by:

$$\boldsymbol{\xi}_{\text{Ex}} = (t_x, t_y, t_z, \theta u_x, \theta u_y, \theta u_z)$$

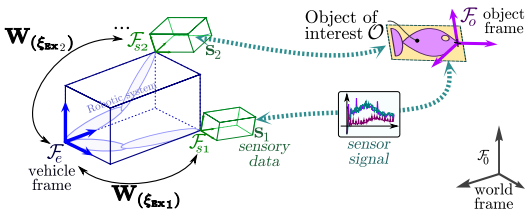

Fig. 1. Sensor-based model.

Now, let $\mathbf{s}$ be a set of data provided by a sensor mounted on a robot. Assuming that only the sensor motion $\mathbf{v_s}$ affects the sensor signal variations $\dot{\mathbf{s}}$ (i.e. motionless environment), the variation of the sensor features $\dot{\mathbf{s}}$ can be expressed by:

$$\dot{\mathbf{s}} = \mathbf{L}_{\mathbf{s}(\mathbf{s},\boldsymbol{\xi}_{\text{In}},\mathbf{a})} \mathbf{v_s} \tag{3}$$

where $\mathbf{L}_{\mathbf{s}(\mathbf{s},\boldsymbol{\xi}_{\text{In}},\mathbf{a})}$ is named the interaction matrix of $\mathbf{s}$ [4]. It can be derived for many features coming from exteroceptive sensors. It depends mainly on the type of considered sensory data $\mathbf{s}$ and on the intrinsic parameters $\boldsymbol{\xi}_{\text{In}}$. It may also depend

O. Kermorgant is with INRIA Rennes-Bretagne Atlantique, Campus de Beaulieu, Rennes, France; olivier.kermorgant@irisa.fr

D. Folio is with PRISME, Université d'Orléans, Orléans, France david.folio@ensi-bourges.fr

F. Chaumette is with INRIA Rennes-Bretagne Atlantique, Campus de Beaulieu, Rennes, France; francois.chaumette@irisa.fr

on other data: for instance the interaction matrix of an image point observed by a camera depends on the depth $\mathbf{z}$ of that point, which is not actually known. Therefore, for a single sensor, we have the following general sensor-based definition:

$$\dot{\mathbf{s}} = \mathbf{L}_{\mathbf{s}(\mathbf{s},\boldsymbol{\xi}_{\text{In}})}\mathbf{v}_{\mathbf{s}} = \mathbf{L}_{\mathbf{s}(\mathbf{s},\boldsymbol{\xi}_{\text{In}})}{}^{\mathbf{s}}\mathbf{H}_{e(\boldsymbol{\xi}_{\text{Ex}})}\mathbf{v}_e = \mathbf{H}_{(\mathbf{s},\boldsymbol{\xi}_{\text{In}},\boldsymbol{\xi}_{\text{Ex}})}\mathbf{v}_e \quad (4)$$

This model can be extended to multiple sensors, and in case of n sensors we get:

$$\left.\begin{array}{l} \dot{\mathbf{s}}_1 = \mathbf{L}_{\mathbf{s}(\mathbf{s}_1,\boldsymbol{\xi}_{\text{In}\mathbf{1}})}\mathbf{v}_{\mathbf{s}_1} = \mathbf{H}_{(\mathbf{s}_1,\boldsymbol{\xi}_{\text{In}1},\boldsymbol{\xi}_{\text{Ex}1})}\mathbf{v}_e \\ \vdots \qquad\qquad \vdots \\ \dot{\mathbf{s}}_n = \mathbf{L}_{\mathbf{s}(\mathbf{s}_n,\boldsymbol{\xi}_{\text{In}\mathbf{n}})}\mathbf{v}_{\mathbf{s}_n} = \mathbf{H}_{(\mathbf{s}_n,\boldsymbol{\xi}_{\text{In}n},\boldsymbol{\xi}_{\text{Ex}n})}\mathbf{v}_e \end{array}\right\} \dot{\mathbf{s}} = \mathbf{H}_{(\mathbf{s},\boldsymbol{\xi}_{\text{In}},\boldsymbol{\xi}_{\text{Ex}})}\mathbf{v}_e \quad (5)$$

Our idea is to exploit the above sensor/robot motion link to estimate the calibration parameters: $\{\boldsymbol{\xi}_{\text{In}}, \boldsymbol{\xi}_{\text{Ex}}\}$.

III. Intrinsic parameters ($\boldsymbol{\xi}_{\text{IN}}$)

If the intrinsic parameters are wrong, then we obtain an error $\boldsymbol{\varepsilon}_{(\widetilde{\boldsymbol{\xi}_{\text{In}}})}$ which can be formulated as follow:

$$\dot{\mathbf{s}} = \mathbf{L}_{\mathbf{s}(\mathbf{s},\boldsymbol{\xi}_{\text{In}})}\mathbf{v}_{\mathbf{s}} = \mathbf{L}_{\mathbf{s}(\mathbf{s},\widetilde{\boldsymbol{\xi}_{\text{In}}})}\mathbf{v}_{\mathbf{s}} + \boldsymbol{\varepsilon}_{(\widetilde{\boldsymbol{\xi}_{\text{In}}})}$$

Hence, the objective is to minimize the following error:

$$\left\|\boldsymbol{\varepsilon}_{(\widetilde{\boldsymbol{\xi}_{\text{In}}})}\right\|^2 = \left\|\dot{\mathbf{s}} - \mathbf{L}_{\mathbf{s}(\mathbf{s},\widetilde{\boldsymbol{\xi}_{\text{In}}})}\mathbf{v}_{\mathbf{s}}\right\|^2 \quad (6)$$

A. General formulation

We assume here that we perfectly measure the sensor velocity $\mathbf{v}_{\mathbf{s}}$ and the sensor feature variation $\dot{\mathbf{s}}$. The idea is to minimize the error with a classical non-linear least square approach, that requires the computation of the error jacobian $\boldsymbol{J}_{(\mathbf{s},\widetilde{\boldsymbol{\xi}_{\text{In}}})}$. If we consider a set of m intrinsic parameters $\boldsymbol{\xi}_{\text{In}} = (\boldsymbol{\xi}_{\text{In}1}, \ldots, \boldsymbol{\xi}_{\text{In}k})$, from Eq. (6) we easily deduce the error jacobian, and we get:

$$\begin{aligned} \boldsymbol{J}_{(\mathbf{s},\widetilde{\boldsymbol{\xi}_{\text{In}}})} &= \frac{\partial \boldsymbol{\varepsilon}_{(\widetilde{\boldsymbol{\xi}_{\text{In}}})}}{\partial \boldsymbol{\xi}_{\text{In}}} = -\frac{\partial\left(\mathbf{L}_{\mathbf{s}(\mathbf{s},\widetilde{\boldsymbol{\xi}_{\text{In}}})}\mathbf{v}_{\mathbf{s}}\right)}{\partial \boldsymbol{\xi}_{\text{In}}} \\ &= -\begin{bmatrix} \frac{\partial \mathbf{L}_{\mathbf{s}1}}{\partial \boldsymbol{\xi}_{\text{In}}}\mathbf{v}_{\mathbf{s}} \\ \vdots \\ \frac{\partial \mathbf{L}_{\mathbf{s}k}}{\partial \boldsymbol{\xi}_{\text{In}}}\mathbf{v}_{\mathbf{s}} \end{bmatrix} \end{aligned} \quad (7)$$

where $\mathbf{L}_{\mathbf{s}i}$ is the 1×6 interaction matrix of the i-th sensor data.

Finally, the error jacobian $\boldsymbol{J}_{(\mathbf{s},\widetilde{\boldsymbol{\xi}_{\text{In}}})}$ only depends on the derivative of the interaction matrix $\mathbf{L}_{\mathbf{s}(\mathbf{s},\widetilde{\boldsymbol{\xi}_{\text{In}}})}$ wrt. the intrinsic parameters $\boldsymbol{\xi}_{\text{In}}$. Thus, the error jacobian can be known for any set of sensory data $\mathbf{s}$ for which we are able to compute the analytical form of its interaction matrix.

The minimization is done through a gradient descent in (6), that is:

$$\Delta\boldsymbol{\xi}_{\text{In}} = -\lambda_{\boldsymbol{\xi}_{\text{In}}}\boldsymbol{J}^{+}_{(\mathbf{s},\widetilde{\boldsymbol{\xi}_{\text{In}}})}\boldsymbol{\varepsilon}_{(\widetilde{\boldsymbol{\xi}_{\text{In}}})} \quad (8)$$

where $\lambda_{\boldsymbol{\xi}_{\text{In}}} > 0$ tunes the decay, and $\boldsymbol{J}^{+}_{(\mathbf{s},\widetilde{\boldsymbol{\xi}_{\text{In}}})}$ denotes the Moore-Penrose pseudo-inverse of the error jacobian $\boldsymbol{J}_{(\mathbf{s},\widetilde{\boldsymbol{\xi}_{\text{In}}})}$. Afterward, the intrinsic parameters are iteratively updated with:

$$\widetilde{\boldsymbol{\xi}_{\text{In}}}(t_{k+1}) = \widetilde{\boldsymbol{\xi}_{\text{In}}}(t_k) \oplus \Delta\boldsymbol{\xi}_{\text{In}} \quad (9)$$

where $\widetilde{\boldsymbol{\xi}_{\text{In}}}(t_0)$ is an initial guess of the intrinsic parameters, and $\oplus$ is an update operator (for instance a simple weighted sum).

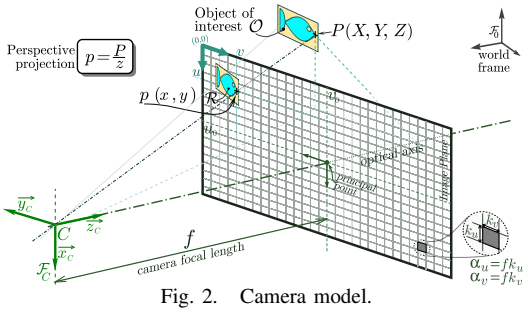

Fig. 2. Camera model.

B. Application to camera self-calibration

We apply here the proposed formalism to the camera self-calibration case. We use image features for which the interaction matrix $\mathbf{L}_{\mathbf{s}(\mathbf{s},\boldsymbol{\xi}_{\text{In}})}$ can be determined analytically. Such expressions are available for different kinds of visual features such as points, straight lines, circles [6], and image moments [2]. The point is a very simple primitive which can be easily extracted from the image. It is then widely used in robot vision. This is the reason why we first address this case by considering a visual landmark made of n points.

Let us recall that, using the pinhole camera model, a 3D point P of coordinates (X, Y, Z) in the camera frame $\mathcal{F}_C$ is projected into a 2D point with coordinates $p = (x, y)$ in the image plane (see Fig. 2) with a perspective projection model:

$$x = \frac{X}{Z}, \qquad y = \frac{Y}{Z} \quad (10)$$

If we denote (x_p, y_p) the pixel coordinates of the corresponding point in the image, it is related to the normalized coordinates (x, y) by:

$$\left\{\begin{array}{l} x_p = x_c + \alpha_x x \\ y_p = y_c + \alpha_y y \end{array}\right. \quad (11)$$

where (α_x, α_y) is the ratio between the focal length and the pixel size, and (x_c, y_c) is the principal point coordinate in pixel (see Fig. 2). These four parameters define the camera intrinsic parameters, that is $\boldsymbol{\xi}_{\text{In}} = (\alpha_x, \alpha_y, x_c, y_c)$.

We have to use the interaction matrix $\mathbf{L}_{\mathbf{s}(\mathbf{p},\boldsymbol{\xi}_{\text{In}})}$ that links the motion $\dot{p} = (\dot{x_p}, \dot{y_p})$ of a point $p = (x_p, y_p)$ in the image to the camera motion $\mathbf{v}_{\mathbf{s}}$. This interaction matrix is deduced from the optic flow equations, and is given by [6]:

$$\begin{aligned} \mathbf{L}_{\mathbf{s}(\mathbf{p}\boldsymbol{\xi}_{\text{In}}\mathbf{Z})} &= \begin{bmatrix} \mathbf{L}_{x_p}(p,\boldsymbol{\xi}_{\text{In}},Z) \\ \mathbf{L}_{y_p}(p,\boldsymbol{\xi}_{\text{In}},Z) \end{bmatrix} \\ &= \begin{bmatrix} \frac{\alpha_x}{Z} & 0 & \frac{x_p - x_c}{Z} & \frac{(x_p-x_c)(y_p-y_c)}{\alpha_y} & -\left(\alpha_x + \frac{(x_p-x_c)^2}{\alpha_x}\right) & \frac{\alpha_x}{\alpha_y}(y_p - y_c) \\ 0 & -\frac{\alpha_y}{Z} & \frac{y_p - y_c}{Z} & \left(\alpha_y + \frac{(y_p-y_c)^2}{\alpha_y}\right) & -\frac{(x_p-x_c)(y_p-y_c)}{\alpha_x} & -\frac{\alpha_y}{\alpha_x}(x_p - x_c) \end{bmatrix} \end{aligned} \quad (12)$$

As one can see, the interaction matrix $\mathbf{L}_{\mathbf{s}(\mathbf{p},\boldsymbol{\xi}_{\text{In}})}$ induces the value of the depth Z of each considered point p. Several approaches may be used to determine it. The most obvious solution is to measure it using dedicated sensors. However, if the robotic platform is not equipped with such sensors, it is possible to use structure from motion (SFM) techniques [3], [15], signal processing methods [13], or even pose relative estimation [16]. Recent work has also shown that Z-depth can be retrieved in the case of partially uncalibrated camera [5].

All these techniques requiring calibrated sensors, we propose to treat the depths as additional unknown parameters of the interaction matrix: we use the sensor/robot motion model to estimate the Z-depth [7].

From the analytical interaction matrix, we can express the error jacobian for this visual feature. With $\boldsymbol{\xi}_{\text{In}} = (\alpha_x, \alpha_y, x_c, y_c)$, (7) yields:

$$\boldsymbol{J}_{(p,\widetilde{\boldsymbol{\xi}_{\text{In}}})} = -\begin{bmatrix} \frac{\partial \mathbf{L}_{x_p}}{\partial \alpha_x}\mathbf{v_s} & \frac{\partial \mathbf{L}_{x_p}}{\partial \alpha_y}\mathbf{v_s} & \frac{\partial \mathbf{L}_{x_p}}{\partial x_c}\mathbf{v_s} & \frac{\partial \mathbf{L}_{x_p}}{\partial y_c}\mathbf{v_s} \\ \frac{\partial \mathbf{L}_{y_p}}{\partial \alpha_x}\mathbf{v_s} & \frac{\partial \mathbf{L}_{y_p}}{\partial \alpha_y}\mathbf{v_s} & \frac{\partial \mathbf{L}_{y_p}}{\partial x_c}\mathbf{v_s} & \frac{\partial \mathbf{L}_{y_p}}{\partial y_c}\mathbf{v_s} \end{bmatrix}$$

with, from (12):

$$\begin{array}{ll} \frac{\partial \mathbf{L}_{x_p}}{\partial \alpha_x} = \begin{bmatrix} -\frac{1}{Z} & 0 & 0 & 0 & -\left(1-x^2\right) & y \end{bmatrix} \\ \frac{\partial \mathbf{L}_{x_p}}{\partial \alpha_y} = \begin{bmatrix} 0 & 0 & 0 & -\frac{\alpha_x}{\alpha_y}xy & 0 & -\frac{\alpha_x}{\alpha_y}y \end{bmatrix} \\ \frac{\partial \mathbf{L}_{x_p}}{\partial x_c} = \begin{bmatrix} 0 & 0 & -\frac{1}{Z} & -y & 2x & 0 \end{bmatrix} \\ \frac{\partial \mathbf{L}_{x_p}}{\partial y_c} = \begin{bmatrix} 0 & 0 & 0 & -\frac{\alpha_x}{\alpha_y}x & 0 & -\frac{\alpha_x}{\alpha_y} \end{bmatrix} \end{array} \tag{13}$$

$$\begin{array}{ll} \frac{\partial \mathbf{L}_{y_p}}{\partial \alpha_x} = \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{\alpha_y}{\alpha_x}xy & \frac{\alpha_y}{\alpha_x}x \end{bmatrix} \\ \frac{\partial \mathbf{L}_{y_p}}{\partial \alpha_y} = \begin{bmatrix} 0 & -\frac{1}{Z} & 0 & \left(1-y^2\right) & 0 & -x \end{bmatrix} \\ \frac{\partial \mathbf{L}_{y_p}}{\partial x_c} = \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{\alpha_y}{\alpha_x}y & \frac{\alpha_y}{\alpha_x} \end{bmatrix} \\ \frac{\partial \mathbf{L}_{y_p}}{\partial y_c} = \begin{bmatrix} 0 & 0 & -\frac{1}{Z} & -2y & x & 0 \end{bmatrix} \end{array} \tag{14}$$

where $x = \frac{x_p - x_c}{\alpha_x}$ and $y = \frac{y_p - y_c}{\alpha_y}$.

As we consider Z as an additional unknown parameter, we have to express Z error jacobian $\frac{\partial \mathbf{L}_{\mathbf{s}(\mathbf{p},\widetilde{\boldsymbol{\xi}_{\text{In}}})}}{\partial Z}$, by deriving Eq. (6) wrt. Z, and we get from (12):

$$\begin{array}{l} \frac{\partial \mathbf{L}_{x_p}}{\partial Z} = \begin{bmatrix} \alpha_x \frac{1}{Z^2} & 0 & -\alpha_x \frac{x}{Z^2} & 0 & 0 & 0 \end{bmatrix} \\ \frac{\partial \mathbf{L}_{y_p}}{\partial Z} = \begin{bmatrix} 0 & \alpha_y \frac{1}{Z^2} & -\alpha_y \frac{y}{Z^2} & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Finally, the complete jacobian wrt. $(\alpha_x, \alpha_y, x_c, y_c, Z)$ can be written:

$$\boldsymbol{J}_{(p,\widetilde{\boldsymbol{\xi}_{\text{In}}})} = \begin{bmatrix} -\frac{v_x}{Z} - \left(1-x^2\right)\omega_y + y\omega_z & \frac{\alpha_y}{\alpha_x}x\left(y\omega_y + \omega_z\right) \\ -\frac{\alpha_x}{\alpha_y}y\left(x\omega_x + \omega_z\right) & -\frac{v_y}{Z} + \left(1-y^2\right)\omega_x - x\omega_z \\ -\frac{v_z}{Z} - y\omega_x + 2x\omega_y & \frac{\alpha_y}{\alpha_x}\left(y\omega_y + \omega_z\right) \\ -\frac{\alpha_x}{\alpha_y}\left(x\omega_x + \omega_z\right) & -\frac{v_z}{Z} - 2y\omega_x + x\omega_y \\ \frac{\alpha_x}{Z^2}\left(v_x - xv_z\right) & \frac{\alpha_y}{Z^2}\left(v_y - yv_z\right) \end{bmatrix}^\top \tag{15}$$

Under the condition that this jacobian is of rank 2, we have to use at least 4 points to get a full rank jacobian in the error minimization: the 8 features will allow estimating the 4 intrinsic parameters and the 4 depths.

IV. Extrinsic parameters ($\boldsymbol{\xi}_{\text{EX}}$)

If the extrinsic parameters are not correctly estimated, we have an error $\boldsymbol{\varepsilon}_{(\widetilde{\boldsymbol{\xi}_{\text{EX}}})}$ which can be formulated as follow:

$$\dot{\mathbf{s}} = \mathbf{L}_{\mathbf{s}(\mathbf{s},\boldsymbol{\xi}_{\text{In}})}\,{}^{\mathbf{s}}\mathbf{W}_{e(\widetilde{\boldsymbol{\xi}_{\text{EX}}})}\mathbf{v}_e + \boldsymbol{\varepsilon}_{(\widetilde{\boldsymbol{\xi}_{\text{EX}}})} \tag{16}$$

and the goal is then to minimize the following error:

$$\left\|\boldsymbol{\varepsilon}_{(\widetilde{\boldsymbol{\xi}_{\text{EX}}})}\right\|^2 = \left\|\dot{\mathbf{s}} - \mathbf{L}_{\mathbf{s}(\mathbf{s},\boldsymbol{\xi}_{\text{In}})}\,{}^{\mathbf{s}}\mathbf{W}_{e(\widetilde{\boldsymbol{\xi}_{\text{EX}}})}\mathbf{v}_e\right\|^2 \tag{17}$$

A. Approach formulation

If we recall the extrinsic parameters denote the pose of the sensor wrt. the end-effector frame, we notice this minimization problem takes place in SE_3 and then we can define the velocity screw $\mathbf{v}$ of the estimated end-effector frame $\widetilde{\mathcal{F}_e}$ that virtually moves as the parameters are being updated. We denote $\mathbf{L}_{\boldsymbol{\varepsilon}_{(\widetilde{\boldsymbol{\xi}_{\text{EX}}})}}$ the error interaction matrix that links the error time derivative to the screw of the estimated frame:

$$\dot{\boldsymbol{\varepsilon}}_{(\widetilde{\boldsymbol{\xi}_{\text{EX}}})} = \mathbf{L}_{\boldsymbol{\varepsilon}_{(\widetilde{\boldsymbol{\xi}_{\text{EX}}})}}\mathbf{v} \tag{18}$$

This relation allows minimizing the error in (17): we can indeed impose in (18) an exponential decrease for the error during the minimization process, which leads to the expression of the instantaneous velocity:

$$\mathbf{L}_{\boldsymbol{\varepsilon}_{(\widetilde{\boldsymbol{\xi}_{\text{EX}}})}}\mathbf{v} = \dot{\boldsymbol{\varepsilon}}_{(\widetilde{\boldsymbol{\xi}_{\text{EX}}})} = -\lambda\boldsymbol{\varepsilon}_{(\widetilde{\boldsymbol{\xi}_{\text{EX}}})}$$

from which we deduce:

$$\mathbf{v} = -\lambda\mathbf{L}^{+}_{\boldsymbol{\varepsilon}_{(\widetilde{\boldsymbol{\xi}_{\text{EX}}})}}\boldsymbol{\varepsilon} = -\lambda\mathbf{L}^{+}_{\boldsymbol{\varepsilon}_{(\widetilde{\boldsymbol{\xi}_{\text{EX}}})}}\left(\mathbf{L}_{\mathbf{s}(\mathbf{s},\boldsymbol{\xi}_{\text{In}})}\mathbf{W}_{(\widetilde{\boldsymbol{\xi}_{\text{EX}}})}\mathbf{v}_e - \dot{\mathbf{s}}\right) \tag{19}$$

where $\mathbf{L}^{+}_{\boldsymbol{\varepsilon}_{(\widetilde{\boldsymbol{\xi}_{\text{EX}}})}}$ is the Moore-Penrose pseudo-inverse of the interaction matrix $\mathbf{L}_{\boldsymbol{\varepsilon}_{(\widetilde{\boldsymbol{\xi}_{\text{EX}}})}}$. This virtual screw $\mathbf{v}$ allows to compute the values of ${}^{\mathbf{s}}\mathbf{t}_e$ and ${}^{\mathbf{s}}\mathbf{R}_e$ (hence ${}^{\mathbf{s}}\mathbf{W}_{e(\widetilde{\boldsymbol{\xi}_{\text{EX}}})}$) using the exponential map. If ${}^{\mathbf{s}}\widetilde{\mathbf{M}}_{e(t)}$ is the homogeneous matrix expressing the position of $\widetilde{\mathcal{F}_e}$ in $\mathcal{F}_{\mathbf{s}}$ at t, it is updated following:

$$\begin{array}{rll} {}^{\mathbf{s}}\widetilde{\mathbf{M}}_{e(t+1)} = & {}^{\mathbf{s}}\widetilde{\mathbf{M}}_{e(t)}\,{}^{e(t)}\mathbf{M}_{e(t+1)} \\ \text{with} \quad {}^{e(t)}\mathbf{M}_{e(t+1)} = & exp(\mathbf{v}, \delta t) \end{array} \tag{20}$$

The iteration is then repeated until the error vanishes.

B. The error interaction matrix

In order to minimize the error in (19), one must find the expression of the interaction matrix defined in (18). According to (17), the only element that depends on the parameters to be estimated is the transformation matrix ${}^{\mathbf{s}}\mathbf{W}_e$ which leads to the relation:

$$\dot{\boldsymbol{\varepsilon}}_{(\widetilde{\boldsymbol{\xi}_{\text{EX}}})} = \mathbf{L}_{\mathbf{s}(\mathbf{s},\boldsymbol{\xi}_{\text{In}})}\,{}^{\mathbf{s}}\dot{\mathbf{W}}_{e(\widetilde{\boldsymbol{\xi}_{\text{EX}}})}\mathbf{v}_e \tag{21}$$

$$\text{with} \quad {}^{\mathbf{s}}\dot{\mathbf{W}}_e = \left[\begin{array}{c|c} {}^{\mathbf{s}}\dot{\mathbf{R}}_e & \dot{[{}^{\mathbf{s}}\mathbf{t}_e]}_\times\,{}^{\mathbf{s}}\mathbf{R}_e + [{}^{\mathbf{s}}\mathbf{t}_e]_\times\,{}^{\mathbf{s}}\dot{\mathbf{R}}_e \\ \hline \mathbf{0}_{3\times3} & {}^{\mathbf{s}}\dot{\mathbf{R}}_e \end{array}\right] \tag{22}$$

$$\text{where} \quad \left\{\begin{array}{ll} {}^{\mathbf{s}}\dot{\mathbf{t}}_e = & -\boldsymbol{v} + [{}^{\mathbf{s}}\mathbf{t}_e]_\times\,\boldsymbol{\omega} \\ {}^{\mathbf{s}}\dot{\mathbf{R}}_e = & {}^{\mathbf{s}}\mathbf{R}_e\,[\boldsymbol{\omega}]_\times \end{array}\right. \tag{23}$$

$\mathbf{v} = (\boldsymbol{v}, \boldsymbol{\omega})$ being, as already said, the instantaneous velocity of the virtual end-effector frame $\widetilde{\mathcal{F}_e}$ corresponding to the current estimation of the extrinsic parameters. There exist three $\mathbf{v}$-dependent 3×3 matrices $\mathbf{A}(\boldsymbol{v})$, $\mathbf{B}(\boldsymbol{\omega})$ and $\mathbf{C}(\boldsymbol{\omega})$ such that:

$$\dot{\mathbf{W}}_{(\boldsymbol{\xi}_{\text{EX}})} = \left[\begin{array}{c|c} \mathbf{B} & \mathbf{A}+\mathbf{C} \\ \hline \mathbf{0}_{3\times3} & \mathbf{B} \end{array}\right] \tag{24}$$

with, by injecting (23) in (22):

$$\begin{array}{rcl} \mathbf{B} & = & {}^{\mathbf{s}}\dot{\mathbf{R}}_e = {}^{\mathbf{s}}\mathbf{R}_e[\boldsymbol{\omega}]_\times \\ \mathbf{A}+\mathbf{C} & = & \dot{[{}^{\mathbf{s}}\mathbf{t}_e]}_\times\,{}^{\mathbf{s}}\mathbf{R}_e + [{}^{\mathbf{s}}\mathbf{t}_e]_\times\,{}^{\mathbf{s}}\dot{\mathbf{R}}_e \\ & = & \left[-\boldsymbol{v} + [{}^{\mathbf{s}}\mathbf{t}_e]_\times\,\boldsymbol{\omega}\right]_\times\,{}^{\mathbf{s}}\mathbf{R}_e + [{}^{\mathbf{s}}\mathbf{t}_e]_\times\,{}^{\mathbf{s}}\mathbf{R}_e\,[\boldsymbol{\omega}]_\times \end{array} \tag{25}$$

from which we deduce:

$$\mathbf{A} = -[\boldsymbol{v}]_\times {}^{\mathbf{s}}\mathbf{R}_e \tag{26}$$

$$\mathbf{C} = \left[[{}^{\mathbf{s}}\mathbf{t}_e]_\times \boldsymbol{\omega}\right]_\times {}^{\mathbf{s}}\mathbf{R}_e + [{}^{\mathbf{s}}\mathbf{t}_e]_\times {}^{\mathbf{s}}\mathbf{R}_e [\boldsymbol{\omega}]_\times \tag{27}$$

If we denote $\mathbf{L}_{\mathbf{s}(\mathbf{s},\boldsymbol{\xi}_{\text{In}})} = [\mathbf{L}_{\boldsymbol{v}}\ \mathbf{L}_{\boldsymbol{\omega}}]$, substituting (24) in (21) yields:

$$\begin{aligned}
\dot{\boldsymbol{\varepsilon}}_{(\widetilde{\boldsymbol{\xi}_{\text{Ex}}})} &= [\mathbf{L}_{\boldsymbol{v}}\ \mathbf{L}_{\boldsymbol{\omega}}] \left[\begin{array}{c|c} \mathbf{B} & \mathbf{A}+\mathbf{C} \\ \hline \mathbf{0}_{3\times 3} & \mathbf{B} \end{array}\right] \begin{bmatrix} \boldsymbol{v}_e \\ \boldsymbol{\omega}_e \end{bmatrix} \\
&= [\mathbf{L}_{\boldsymbol{v}}\ \mathbf{L}_{\boldsymbol{\omega}}] \begin{bmatrix} \mathbf{B}\boldsymbol{v}_e + (\mathbf{A}+\mathbf{C})\boldsymbol{\omega}_e \\ \mathbf{B}\boldsymbol{\omega}_e \end{bmatrix} \\
&= \mathbf{L}_{\boldsymbol{v}}\mathbf{B}\boldsymbol{v}_e + \mathbf{L}_{\boldsymbol{v}}\mathbf{A}\boldsymbol{\omega}_e + \mathbf{L}_{\boldsymbol{v}}\mathbf{C}\boldsymbol{\omega}_e + \mathbf{L}_{\boldsymbol{\omega}}\mathbf{B}\boldsymbol{\omega}_e \\
&= \mathbf{K}_1\boldsymbol{v} + \mathbf{K}_2\boldsymbol{\omega}
\end{aligned} \tag{28}$$

$$\text{with} \begin{cases} \mathbf{K}_1\boldsymbol{v} = & \mathbf{L}_{\boldsymbol{v}}\mathbf{A}\boldsymbol{\omega}_e \\ \mathbf{K}_2\boldsymbol{\omega} = & \mathbf{L}_{\boldsymbol{v}}\mathbf{B}\boldsymbol{v}_e + \mathbf{L}_{\boldsymbol{v}}\mathbf{C}\boldsymbol{\omega}_e + \mathbf{L}_{\boldsymbol{\omega}}\mathbf{B}\boldsymbol{\omega}_e \end{cases}$$

Substituting (26) in (28) leads to:

$$\begin{aligned}
\mathbf{K}_1\boldsymbol{v} &= \mathbf{L}_{\boldsymbol{v}}\mathbf{A}\boldsymbol{\omega}_e = -\mathbf{L}_{\boldsymbol{v}} [\boldsymbol{v}]_\times {}^{\mathbf{s}}\mathbf{R}_e\boldsymbol{\omega}_e \\
&= \mathbf{L}_{\boldsymbol{v}} [{}^{\mathbf{s}}\mathbf{R}_e\boldsymbol{\omega}_e]_\times \boldsymbol{v}
\end{aligned}$$

from which we deduce:

$$\mathbf{K}_1 = \mathbf{L}_{\boldsymbol{v}} [{}^{\mathbf{s}}\mathbf{R}_e\boldsymbol{\omega}_e]_\times \tag{29}$$

In the same way, substituting (25) and (27) in (28) leads to:

$$\begin{aligned}
\mathbf{K}_2\boldsymbol{\omega} &= \mathbf{L}_{\boldsymbol{v}}\mathbf{B}\boldsymbol{v}_e + \mathbf{L}_{\boldsymbol{v}}\mathbf{C}\boldsymbol{\omega}_e + \mathbf{L}_{\boldsymbol{\omega}}\mathbf{B}\boldsymbol{\omega}_e \\
&= \mathbf{K}_{2a}\boldsymbol{\omega} + \mathbf{K}_{2b}\boldsymbol{\omega} + \mathbf{K}_{2c}\boldsymbol{\omega}
\end{aligned}$$

with

$$\begin{aligned}
\mathbf{L}_{\boldsymbol{v}}\mathbf{B}\boldsymbol{v}_e &= \mathbf{L}_{\boldsymbol{v}}{}^{\mathbf{s}}\mathbf{R}_e[\boldsymbol{\omega}]_\times \boldsymbol{v}_e \\
&= -\mathbf{L}_{\boldsymbol{v}}{}^{\mathbf{s}}\mathbf{R}_e [\boldsymbol{v}_e]_\times \boldsymbol{\omega} \\
\mathbf{L}_{\boldsymbol{v}}\mathbf{C}\boldsymbol{\omega}_e &= \mathbf{L}_{\boldsymbol{v}}\left(\left[[{}^{\mathbf{s}}\mathbf{t}_e]_\times\boldsymbol{\omega}\right]_\times {}^{\mathbf{s}}\mathbf{R}_e + [{}^{\mathbf{s}}\mathbf{t}_e]_\times {}^{\mathbf{s}}\mathbf{R}_e[\boldsymbol{\omega}]_\times\right)\boldsymbol{\omega}_e \\
&= \mathbf{L}_{\boldsymbol{v}}\left(\left[[{}^{\mathbf{s}}\mathbf{t}_e]_\times\boldsymbol{\omega}\right]_\times {}^{\mathbf{s}}\mathbf{R}_e\boldsymbol{\omega}_e - [{}^{\mathbf{s}}\mathbf{t}_e]_\times {}^{\mathbf{s}}\mathbf{R}_e[\boldsymbol{\omega}_e]_\times\boldsymbol{\omega}\right) \\
&= -\mathbf{L}_{\boldsymbol{v}}\left([{}^{\mathbf{s}}\mathbf{t}_e]_\times {}^{\mathbf{s}}\mathbf{R}_e[\boldsymbol{\omega}_e]_\times\boldsymbol{\omega} + \left({}^{\mathbf{s}}\mathbf{t}_e{}^\top {}^{\mathbf{s}}\mathbf{R}_e\boldsymbol{\omega}_e\right)\boldsymbol{\omega} - \left(\boldsymbol{\omega}^\top {}^{\mathbf{s}}\mathbf{R}_e\boldsymbol{\omega}_e\right){}^{\mathbf{s}}\mathbf{t}_e\right) \\
&= -\mathbf{L}_{\boldsymbol{v}}\left([{}^{\mathbf{s}}\mathbf{t}_e]_\times {}^{\mathbf{s}}\mathbf{R}_e[\boldsymbol{\omega}_e]_\times\boldsymbol{\omega} + \left({}^{\mathbf{s}}\mathbf{t}_e{}^\top {}^{\mathbf{s}}\mathbf{R}_e\boldsymbol{\omega}_e\right)\boldsymbol{\omega} - {}^{\mathbf{s}}\mathbf{t}_e\left({}^{\mathbf{s}}\mathbf{R}_e\boldsymbol{\omega}_e\right)^\top\boldsymbol{\omega}\right)\right) \\
&= -\mathbf{L}_{\boldsymbol{v}}\left([{}^{\mathbf{s}}\mathbf{t}_e]_\times {}^{\mathbf{s}}\mathbf{R}_e[\boldsymbol{\omega}_e]_\times\boldsymbol{\omega} + \left({}^{\mathbf{s}}\mathbf{t}_e{}^\top {}^{\mathbf{s}}\mathbf{R}_e\boldsymbol{\omega}_e\right)\boldsymbol{\omega} - \left({}^{\mathbf{s}}\mathbf{t}_e\left({}^{\mathbf{s}}\mathbf{R}_e\boldsymbol{\omega}_e\right)^\top\right)\boldsymbol{\omega}\right) \\
&= -\mathbf{L}_{\boldsymbol{v}}\left([{}^{\mathbf{s}}\mathbf{t}_e]_\times {}^{\mathbf{s}}\mathbf{R}_e [\boldsymbol{\omega}_e]_\times - {}^{\mathbf{s}}\mathbf{t}_e{}^\top {}^{\mathbf{s}}\mathbf{R}_e\boldsymbol{\omega}_e\mathbf{I}_3 + {}^{\mathbf{s}}\mathbf{t}_e({}^{\mathbf{s}}\mathbf{R}_e\boldsymbol{\omega}_e)^\top\right)\boldsymbol{\omega} \\
\mathbf{L}_{\boldsymbol{\omega}}\mathbf{B}\boldsymbol{\omega}_e &= \mathbf{L}_{\boldsymbol{\omega}}{}^{\mathbf{s}}\mathbf{R}_e [\boldsymbol{\omega}]_\times \boldsymbol{\omega}_e \\
&= -\mathbf{L}_{\boldsymbol{\omega}}{}^{\mathbf{s}}\mathbf{R}_e [\boldsymbol{\omega}_e]_\times \boldsymbol{\omega}
\end{aligned}$$

from which we deduce:

$$\begin{aligned}
\mathbf{K}_{2a} &= -\mathbf{L}_{\boldsymbol{v}}{}^{\mathbf{s}}\mathbf{R}_e [\boldsymbol{v}_e]_\times \\
\mathbf{K}_{2b} &= -\mathbf{L}_{\boldsymbol{v}}\left([{}^{\mathbf{s}}\mathbf{t}_e]_\times {}^{\mathbf{s}}\mathbf{R}_e [\boldsymbol{\omega}_e]_\times - {}^{\mathbf{s}}\mathbf{t}_e{}^\top {}^{\mathbf{s}}\mathbf{R}_e\boldsymbol{\omega}_e\mathbf{I}_3 + {}^{\mathbf{s}}\mathbf{t}_e({}^{\mathbf{s}}\mathbf{R}_e\boldsymbol{\omega}_e)^\top\right) \\
\mathbf{K}_{2c} &= -\mathbf{L}_{\boldsymbol{\omega}}{}^{\mathbf{s}}\mathbf{R}_e [\boldsymbol{\omega}_e]_\times \\
\mathbf{K}_2 &= \mathbf{K}_{2a} + \mathbf{K}_{2b} + \mathbf{K}_{2c}
\end{aligned} \tag{30}$$

From (28), (29) and (30) we deduce the interaction matrix:

$$\dot{\boldsymbol{\varepsilon}}_{(\widetilde{\boldsymbol{\xi}_{\text{Ex}}})} = [\mathbf{K}_1\ \mathbf{K}_2] \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{\omega} \end{bmatrix} = \mathbf{L}_{\boldsymbol{\varepsilon}_{(\widetilde{\boldsymbol{\xi}_{\text{Ex}}})}} \mathbf{v}$$

$$\text{with} \qquad \mathbf{L}_{\boldsymbol{\varepsilon}_{(\widetilde{\boldsymbol{\xi}_{\text{Ex}}})}} = [\mathbf{K}_1\ \mathbf{K}_2] \tag{31}$$

Assuming $\mathbf{L}_{\mathbf{s}(\mathbf{s},\boldsymbol{\xi}_{\text{In}})}$ is of full rank 6 (i.e., at least 6 independent sensor features are used), $\mathbf{K}_1$ and $\mathbf{K}_2$ coming from skew-symmetric matrices, they are both of rank 2, which makes $\mathbf{L}_{\boldsymbol{\varepsilon}_{(\widetilde{\boldsymbol{\xi}_{\text{Ex}}})}}$ of rank 4 and thus *rank deficient* as we have 6 parameters to estimate. Therefore, we must use at least two measurements to get a full rank error interaction matrix to be inversed:

$$\mathbf{v} = -\lambda \begin{bmatrix} \mathbf{L}_{\boldsymbol{\varepsilon}_1} \\ \vdots \\ \mathbf{L}_{\boldsymbol{\varepsilon}_h} \end{bmatrix}^+ \begin{bmatrix} \mathbf{L}_{\mathbf{s}1}\mathbf{W}_1\mathbf{v}_{e1} - \dot{\mathbf{s}}_1 \\ \vdots \\ \mathbf{L}_{\mathbf{s}h}\mathbf{W}_h\mathbf{v}_{eh} - \dot{\mathbf{s}}_h \end{bmatrix} \tag{32}$$

with $h \geq 2$ being the size of the history used for the extrinsic parameters estimation.

V. Simultaneous estimation

In order to achieve simultaneous intrinsic and extrinsic parameters estimation, the sensor feature error expressed in (16) becomes:

$$\dot{\mathbf{s}} = \mathbf{L}_{\mathbf{s}(\mathbf{s},\widetilde{\boldsymbol{\xi}_{\text{In}}})} {}^{\mathbf{s}}\mathbf{W}_{e(\widetilde{\boldsymbol{\xi}_{\text{Ex}}})}\mathbf{v}_e + \boldsymbol{\varepsilon}_{(\widetilde{\boldsymbol{\xi}_{\text{In}}},\widetilde{\boldsymbol{\xi}_{\text{Ex}}})}$$

From (8) and (18) we can deduce the error evolution wrt. the parameters being updated:

$$\dot{\boldsymbol{\varepsilon}}_{(\widetilde{\boldsymbol{\xi}_{\text{In}}},\widetilde{\boldsymbol{\xi}_{\text{Ex}}})} = \boldsymbol{J}_{(\mathbf{s},\widetilde{\boldsymbol{\xi}_{\text{In}}})}\dot{\boldsymbol{\xi}}_{\text{In}} + \mathbf{L}_{\boldsymbol{\varepsilon}}\mathbf{v}$$

where $\dot{\boldsymbol{\xi}}_{\text{In}}$ is the intrinsic parameters evolution, and $\mathbf{v}$ the velocity of the virtual end-effector frame corresponding to the current extrinsic parameters estimation.
If we denote $\Delta w = [\Delta\boldsymbol{\xi}_{\text{In}}\ \mathbf{v}]$ and impose an exponential decrease of the error, we obtain:

$$\Delta w = -\lambda [\boldsymbol{J}\ \mathbf{L}_{\boldsymbol{\varepsilon}}]^+ \boldsymbol{\varepsilon} \tag{33}$$

From (33) using the update rules (9) and (20) we can perform simultaneous intrinsic and extrinsic parameters estimation. In the case of a camera, the additional Z-depth parameters cannot be estimated in the same time as t_z due to an observability problem, therefore we have to measure the Z-depth somehow in order to perform simultaneous estimation.

VI. Application

We have chosen to perform a visual servoing (VS) task [4]. We first describe the considered task before presenting the obtained results. In addition, in order to validate the proposed approach, we have realized several simulations using ViSP software [11].

A. Vision-based Task

The objective of the task is to position the embedded camera with respect to a visual landmark. To this aim, we have applied the VS technique given in [6] that consists in regulating the following task function to zero:

$$\boldsymbol{e}_{\text{VS}} = \mathbf{C}(\mathbf{s} - \mathbf{s}^\star) \tag{34}$$

where $\mathbf{s}^\star$ represents the desired value of the visual features, while $\mathbf{C}$ is a full-rank combination matrix allowing to take into account more visual features than available DOFs [6]. A simple way to choose $\mathbf{C}$ is to consider the Moore-Penrose pseudo-inverse of an estimation of the interaction matrix: $\mathbf{C} = \hat{\mathbf{L}}_{\mathbf{s}}^+$ [4]. One often used estimation is the interaction matrix computed at the desired configuration: ${\mathbf{L}_{\mathbf{s}^\star}}^+$ with the known desired Z-depths. However, estimating the Z-depths allows using the current interaction matrix that improves the VS behavior. In the sequel, we use the current interaction

matrix while performing intrinsic calibration, and the desired interaction matrix while performing extrinsic calibration (Z-depths not estimated).

Using Eqs (4) and (34), and considering $\mathbf{v_s}$ as the input of the robot controller, we obtain the control law that ensures an exponential decoupled decrease of the error [4]:

$$\mathbf{v_s} = -\lambda_{\text{VS}} \hat{\mathbf{L}}_{\mathbf{s}}^{+} (\mathbf{s} - \mathbf{s}^{\star}) \tag{35}$$

The control law requires the knowledge of the intrinsic and extrinsic parameters of the camera: therefore, our goal is to estimate the parameters during the VS in order to perform the task with the best behaviour.

B. Simulation results

Simulations are done with the following parameters : $\boldsymbol{\xi}_{\text{In}} = (595, 607, 192, 144)$, $(t_x, t_y, t_z) = (0.5, -0.3, 0.1)m$ and $(\theta u_x, \theta u_y, \theta u_z) = (-30, 45, 60)deg$.

Fig.3 shows the estimation of $\boldsymbol{\xi}_{\text{In}}$ and Z_i-depths, from the robot velocity and the features variation measurement after the *first* iteration of the VS task. In this first estimation, the parameters and Z_i-depths do not converge perfectly towards the right value yet, the first guess for $Z_i(0)$ inducing a local minimum.

Fig.4 shows the same for $\boldsymbol{\xi}_{\text{EX}}$, this time the estimation is done after *two* iterations of the VS task (see (32)). Here, the estimation converges exactly towards the right values.

Fig.5 shows $\boldsymbol{\xi}_{\text{In}}$ and Z_i estimation obtained during the first 20 iterations of the VS. A new estimation is performed at each iteration of the VS since new measurements are available. The values estimated at the first iteration correspond to the achieved estimation in Fig.3. In order to get redundancy, we use up to 4 successive measurements of the VS in the optimization process. This calibration window is damped so that past Z estimated values have less impact on the current estimation: each past iteration has its weight divided by 2. After a few iterations, the Z-depth estimation delay does not impact the $\boldsymbol{\xi}_{\text{In}}$ calibration.

In the same way, Fig.6 shows the simulation results for simultaneous $\boldsymbol{\xi}_{\text{In}}$ and $\boldsymbol{\xi}_{\text{EX}}$ estimation during the first 20 iterations of the VS. This time, we assume the Z-depths are known. The parameters converge towards the right values in a few steps.

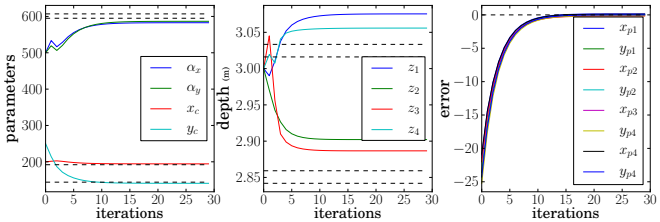

Fig. 3. Estimation of the intrinsic parameters. Evolution of $\boldsymbol{\xi}_{\text{In}}$, Z-depth and $\boldsymbol{\varepsilon}_{(\widetilde{\boldsymbol{\xi}_{\text{In}}})}$ components during the first minimization step.

C. Experimental results

Experiments have been carried on the IRISA 6-dof eye-in-hand robot.

The true parameters are: $\boldsymbol{\xi}_{\text{In}} = (1129, 1127, 313, 270)$, $(t_x, t_y, t_z) = (0.009, -0.23, -0.01)m$ and $(\theta u_x, \theta u_y, \theta u_z) = (-70.5, -68.8, -69.9)deg$. The complete VS task takes 300 iterations, but only the first 100 ones are presented on Fig.7 and 8

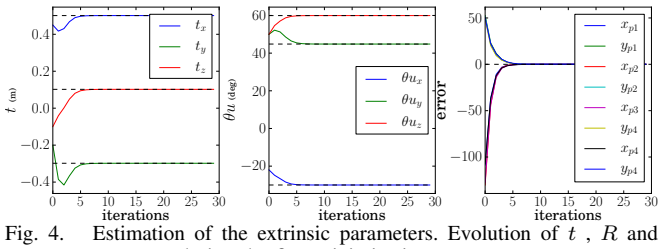

Fig. 4. Estimation of the extrinsic parameters. Evolution of t , R and $\boldsymbol{\varepsilon}_{(\widetilde{\boldsymbol{\xi}_{\text{EX}}})}$ components during the first minimization step.

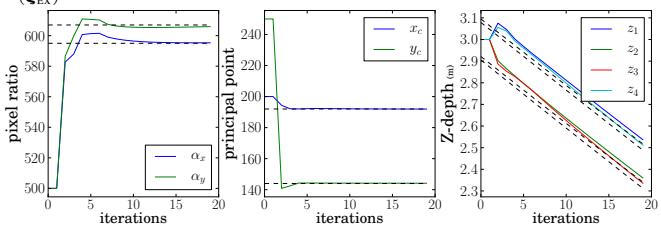

Fig. 5. Estimation of the intrinsic parameters. Evolution of the pixel ratio , principal point and Z-depth during the VS task (first 20 iterations)

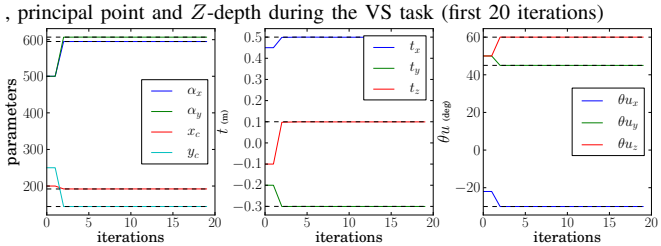

Fig. 6. Simultaneous estimation (known Z-depth). Evolution of $\boldsymbol{\xi}_{\text{In}}$, t and R during the VS task (first 20 iterations)

During the VS, the estimation is stopped when the amplitude of the camera velocity is less than a given threshold (that is near the convergence of the VS) since no more measurements are available when the camera is motionless. Z being no more estimated, the VS uses the desired interaction matrix afterwards.

Contrary to the simulations, here the parameters (but not Z) are updated at each iteration with the mean value between the past estimation and the latest achieved one, in order to filter the values of $\boldsymbol{\xi}_{\text{In}}$ and $\boldsymbol{\xi}_{\text{EX}}$.

For intrinsic calibration (see Fig.7), Z-depths are all initialized to $0.43m$ while the actual value is between $0.38m$ and $0.48m$.

Once again, intrinsic calibration can be performed with a slight delay due to the initial error and the estimation filter. As in simulation, the history window is damped so that past estimated values of Z have less impact.

For extrinsic calibration (see Fig.8), translation and rotation parameters are estimated in a few iterations. As in simulation, we assume Z is known. The results obtained are thus particularly accurate and stable.

The VS results are shown in Fig.9. Intrinsic calibration is performed during the task, with damped calibration window and mean-value parameters update. It is a well-known fact that VS is robust to calibration errors, that is why both VS converge towards the good value. However, estimating the Z-depths allows using the *current* interaction matrix, that leads to a better behavior of the feature error decrease. During this task, we can retrieve the real camera parameters $\boldsymbol{\xi}_{\text{In}}$ and Z-

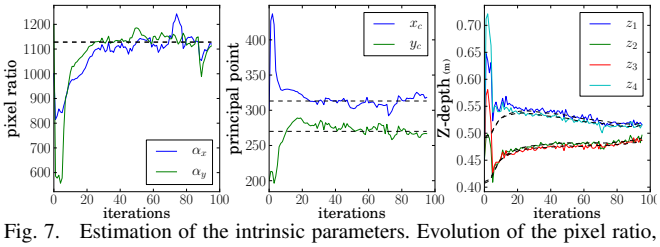

Fig. 7. Estimation of the intrinsic parameters. Evolution of the pixel ratio, principal point and Z_i-depths during the VS task (first 100 iterations)

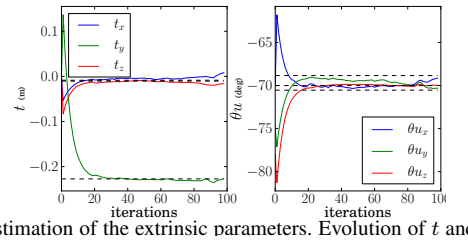

Fig. 8. Estimation of the extrinsic parameters. Evolution of t and R during the VS task

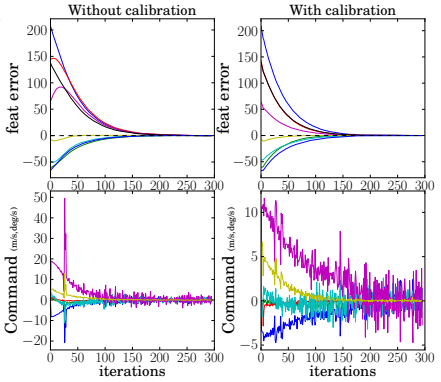

Fig. 9. Visual servoing results with and without calibration. Evolution of the components of the error (top) and the command (bottom). The calibration stops at the iteration 150

depths. However, as the VS converges the camera gets slower and the calibration can become unstable as it depends on the camera velocity. That is why under a certain velocity value, the on-line calibration is turned off, and the VS task uses the *desired* interaction matrix. In this experiment this occurs around the iteration 150. Since the system is very near from the convergence, it has no effect on its behaviour.

VII. Conclusion

A sensor calibration framework has been presented. It uses the sensor/robot motion and velocity measurements instead of usual position measurements. This framework allows retrieving the true intrinsic and extrinsic parameters of any sensor the interaction matrix can be expressed for.

Parameter estimation is expressed as a minimization problem, the jacobian of which can be expressed analytically from the interaction matrix and the frame transformation matrix. The framework has been detailed for a camera observing four points. The classically unknown Z-depths have been considered as additional intrinsic parameters. Simulations and experiments show good results for intrinsic estimation with unknown Z-depth and extrinsic parameters estimation with known Z-depth. Simultaneous intrinsic and extrinsic calibration can also be performed under the condition that Z-depth is known.

As a future work, we could use this framework to perform hand-eye calibration, which consists in estimating the pose between the tool and the camera. The case of varying intrinsic parameters (e.g. zooming camera) could also be studied. Finally, the framework could benefit from an observability analysis that would show the parameters that can be estimated at the same time, and highlight the singular motions preventing from having rank-sufficient matrices.

Acknowledgment

The authors acknowledge support from the ANR project SCUAV (Sensory Control for Unmanned Aerial Vehicles) of the French ministry of research.

References

[1] J. Borenstein and L. Feng, "Measurement and correction of systematic odometry errors in mobile robots," *IEEE Trans. Robot. Autom.*, vol. 12, no. 6, pp. 869–880, Dec. 1996.

[2] F. Chaumette, "Image moments: a general and useful set of features for visual servoing," *IEEE Trans. Robot.*, vol. 20, no. 4, pp. 713–723, Aug. 2004.

[3] F. Chaumette, S. Boukir, P. Bouthemy, and D. Juvin, "Structure from controlled motion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 5, pp. 492–504, May 1996.

[4] F. Chaumette and S. Hutchinson, "Visual servo control, part I: Basic approaches," *IEEE Robot. Autom. Mag.*, vol. 13, no. 4, pp. 82–90, Dec. 2006.

[5] A. De Luca, G. Oriolo, and P. Robuffo Giordano, "Feature depths observation for image-based visual servoing: Theory and experiments," *Int. J. of Robotics Research*, vol. 27, no. 10, pp. 1093–1116, Oct. 2008.

[6] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. Robot. Autom.*, vol. 8, no. 3, pp. 313–326, Jun. 1992.

[7] D. Folio and V. Cadenat, "Dealing with visual features loss during a vision-based task for a mobile robot," *International Journal of Optomechatronics*, vol. 2, pp. 185–204, 2008.

[8] E. Foxlin, "Generalized architecture for simultaneous localization, auto-calibration, and map-building," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 1, Lausanne, Switzerland, Oct. 2002, pp. 527–533.

[9] E. Hemayed, "A survey of camera self-calibration," in *IEEE Conf. on Advanced Video and Signal Based Surveillance.*, Jul. 2003, pp. 351–357.

[10] A. LaMarca, W. Brunette, D. Koizumi, M. Lease, S. B. Sigurdsson, K. Sikorski, D. Fox, and G. Borriello, "Making sensor networks practical with robots," in *1st Int. Conf. on Pervasive Computing (Pervasive'02)*, vol. 2414, London, UK, 2002, pp. 152–166.

[11] E. Marchand, F. Spindler, and F. Chaumette, "ViSP for visual servoing: a generic software platform with a wide class of robot control skills," *IEEE Robot. Autom. Mag.*, vol. 12, no. 4, pp. 40–52, Dec. 2005.

[12] A. Martinelli, J. Weingarten, and R. Siegwart, "Theoretical results on on-line sensor self-calibration," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Beijing, China, Oct. 2006, pp. 43–48.

[13] L. Matthies, T. Kanade, and R. Szeliski, "Kalman filter-based algorithms for estimating depth in image sequences," *Int. J. of Computer Vision*, vol. 3, no. 3, pp. 209–238, 1989.

[14] S.-J. Maybank and O. Faugeras, "A theory of self calibration of a moving camera," *Int. J. of Computer Vision*, vol. 8, no. 1, pp. 123–152, 1992.

[15] J. Oliensis, "Exact two-image structure from controlled motion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 12, pp. 1618–1633, Dec. 2002.

[16] S. Thrun, D. Fox, W. Burgard, and F. Dallaert, "Robust monte-carlo localization for mobile robots," *Artifial Intelligence*, vol. 128, no. 1-2, pp. 99–141, May 2001.