A Comparative Study between Analytic and Estimated Image Jacobian by Using a Stereoscopic System of Cameras

L. Pari, J.M. Sebastián, A. Traslosheros, and L. Angel

Abstract—This paper describes a comparative study of performance between the estimated image Jacobian that come from taking into account the epipolar geometry in a system of two cameras, and the well known analytic image Jacobian that is utilized for most applications in visual servoing. Image Based Visual Servoing architecture is used for controlling a 3 DOF articular system using two cameras in eye to hand configuration. Tests in static and dynamic cases were carried out, and showed that the performance of estimated Jacobian by using the properties of the epipolar geometry is such as good and robust against noise as the analytic Jacobian. This fact is considered as an advantage because the estimated Jacobian does not need laborious previous work prior to control task in contrast to the analytic Jacobian does.

I. INTRODUCTION

VISUAL servoing consists in the use of visual information given by visual sensors (i.e. cameras) to control a robotic system. This kind of control turns out to be very useful in many applications because it allows us to know which objects are present in the scene with high accuracy, as well as their position, orientation and velocity. It makes possible to use robots in new domains where the workspace is not known a priori.

Among the existing classifications of visual servoing [4] [11] [12], one of the most known is the way that visual information is used to define the signal error to control the system [2]: Position Based Visual Servoing (PBVS) and the Image Based Visual Servoing (IBVS). In PBVS features are extracted from the image and used to reconstruct the 3D position of the target, whereas in IBVS the task is defined in the image plane directly through image features. In the latter a matrix is defined called the Image Jacobian, which linearly relates changes in image features and changes in Cartesian coordinates or changes in joints (in this case, it is called full-visual-motor Jacobian [1] [5] [12]).

Analytic Image Jacobian is widely used by the most researchers in visual servoing and, it is well known that it requires a thorough knowledge of the involved systems: calibration of the joint system, kinematic calibration of the vision system, and 3D reconstruction of features positions; they all are common sources of possible errors. In past papers [13] [16] [17] we presented a method to estimate

L. Pari, J. M. Sebastián and A. Traslosheros, are with the Centro de Automática y Robótica, Universidad Politécnica de Madrid, C/ José Gutiérrez Abascal, 2, 28006 Madrid, Spain. (e-mail: {lpari, jsebas, atraslosheros}@etsii.upm.es).

L. Angel is with the Facultad de Ingeniería Electrónica, Universidad Pontificia Bolivariana, Km. 7 Via de Piedecuesta, Bucaramanga, Colombia, (e-mail: langel@etsii.upm.es).

image Jacobian by integrating epipolar geometry of the system [3], and compared with other existing algorithms from the literature by means of tests that consider static [13] and dynamic [16] cases.

In this paper, the performance of two methods of estimating the image Jacobian is compared with that one which calculates the image Jacobian analytically. Methods that consider the estimation of the image Jacobian are represented by both the method that incorporates the fundamental matrix and the recursive least square method, the first method takes into account properties of the epipolar geometry in the system of two cameras [16] and the second method considers properties of recursive techniques [14] which gave good results among others in past work [17].

The present work realizes a comparative study of the mentioned methods and shows that the system with low values of error in calibration (camera and kinematic) is advantageous against the estimation of the Jacobian. But it is showed that when levels of error increase, the use of the estimated Jacobian is widely justified, this is the main contribution of the paper. Tests were carried out with an articular system of 3 d.o.f., and due to the fact that the articular system comprises revolute and prismatic joints, we consider that conclusions obtained in the present study are valid, and can be extended to a system of more d.o.f.

This paper is organized as follows: after the present introduction, section II details the terminology and theoretical concepts used in the paper. Sections III and IV put forward the estimated and analytic Jacobian respectively. Section V describes the control law, whereas section VI describes the workspace applied, tests, and results. Finally section VII reflects our conclusion and future work.

II. IMAGE JACOBIAN

We assume that a robot or positioning system is observed from one or various fixed views. Let $\mathbf{r} = \begin{bmatrix} r_1 & r_2 & \cdots & r_p \end{bmatrix}^T$ be the p-dimensional vector that represents the position of the end effector in a Cartesian coordinate system. Let $\mathbf{q} = \begin{bmatrix} q_1 & q_2 & \cdots & q_n \end{bmatrix}^T$ be the n-dimensional vector that represents the joint position of the robot. Let $\mathbf{s} = \begin{bmatrix} s_1 & s_2 & \cdots & s_m \end{bmatrix}^T$ be the m-dimensional vector that represents the image features (for example the coordinates of a point in one or both images).

The relation between joint velocity of the robot $\dot{\mathbf{q}} = \begin{bmatrix} \dot{q}_1 & \dot{q}_2 & \cdots & \dot{q}_n \end{bmatrix}^T$ and its corresponding velocity in

task space, $\dot{\mathbf{r}} = \begin{bmatrix} \dot{r}_1 & \dot{r}_2 & \cdots & \dot{r}_p \end{bmatrix}^T$, is captured in terms of the robot Jacobian, $\mathbf{J}_{rq}$, as $\dot{\mathbf{r}} = \mathbf{J}_{rq}\dot{\mathbf{q}}$. The relation between feature velocities $\dot{\mathbf{s}} = \begin{bmatrix} \dot{s}_1 & \dot{s}_2 & \cdots & \dot{s}_m \end{bmatrix}^T$ and task space velocities is given by $\dot{\mathbf{s}} = \mathbf{J}_{sr}\dot{\mathbf{r}}_c$, if the chosen feature is a point $\mathbf{s} = (u, v)^T$ in the image, and the Cartesian coordinates of the camera are used, $\mathbf{J}_{sr}$ is given by:

$$\mathbf{J}_{sr} = \begin{bmatrix} f/Z & 0 & -u/Z & -uv/f & (f^2+u^2)/f & -v \\ 0 & f/Z & -v/Z & -(f^2+u^2)/f & uv/f & u \end{bmatrix} \quad (1)$$

where u, v represent the central image coordinates, f is the focal distance, Z is the space coordinate of the point in camera coordinates, and $\dot{\mathbf{r}}_c = \begin{bmatrix} T_x & T_y & T_z & w_x & w_y & w_z \end{bmatrix}^T$ represents the translational and rotational speed of the point in camera coordinates also. Generally, $\mathbf{J}_{sr}$ is named as the interaction matrix.

The velocity of the image features can be directly related to joint velocities in terms of a composite Jacobian named the full-visual-motor Jacobian [5] [18]:

$$\dot{\mathbf{s}} = \mathbf{J}_{sq}\dot{\mathbf{q}} = \begin{bmatrix} \dfrac{\partial s_1}{\partial q_1} & \cdots & \dfrac{\partial s_1}{\partial q_n} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial s_m}{\partial q_1} & \cdots & \dfrac{\partial s_m}{\partial qn} \end{bmatrix} \dot{\mathbf{q}} \quad ; \quad \text{where } \mathbf{J}_{sq} = \mathbf{J}_{sr}\mathbf{J}_{rq} \quad (2)$$

Analytic Jacobian comes from (1) whereas estimated Jacobian from (2). It must be emphasized that to obtain the analytic Jacobian, there must be considered: the intrinsic parameters of the camera calibration (focal distance, image center coordinates), the 3D reconstruction of the point or an approximation (Z coordinate), the kinematic calibration of the camera (relation between camera coordinates and joint space origin), and the kinematic calibration of the robot. Most of previous works on visual servoing assume that the system structure and the system parameters are known, or the parameters can be identified in an off-line process. In contrast to estimate image Jacobian dynamically based on changes in both features and joints only. Hereafter, the full-visual-motor Jacobian $\mathbf{J}_{sq}$ will be simply named as the Jacobian $\mathbf{J}$.

It is well known that to calculate the analytic Jacobian is necessary to know the depth of the 3D point, the more precise method to calculate the depth (among others, or to consider a depth constant equal to the desired position) is to realize a 3D reconstruction of the point by using two cameras, thus in the present work a system of two cameras is utilized.

A. Multiple View Jacobian

When several views are used, whether the interaction matrix or the full visual-motor Jacobian can be defined as the concatenation of the partial Jacobian for each view [1] [2] [16]:

$$\mathbf{J} = \begin{bmatrix} \mathbf{J'} \\ \mathbf{J''} \end{bmatrix} \quad (3)$$

where $\mathbf{J'}$ and $\mathbf{J''}$ are Jacobians of the first and the second image respectively.

It's worth noticing that both Jacobians share the same joint increments, although visual features are managed independently. In previous work [16], we carried out experiments comparing the results obtained using one of the cameras and those obtained using two cameras: our results showed that using two cameras instead of one improved the behaviour of the all methods we tested. In many applications, improvement on the performance justifies the possible disadvantages: increased equipment cost or calculation time

III. ESTIMATED JACOBIAN

Literature [5] [14] [15] [16] [18] gathers diverse methods to estimate the image Jacobian, Two of them are described in this section, the first method was designed by the authors and is based on incorporating the epipolar constraint of the system [16], the second method is based on the recursive updating of the Jacobian considering a covariance matrix with a forgetting factor [14], due to its good results in past works [13] [16] [17] this method is included into the tests and is described below. It is worth also mentioning the Kalman method where the system is modelled by its state variables which are updated using Kalman filter equations [15], the Broyden method that recursively update the Jacobian by using the last movement and the previous Jacobian [5]. Recently, it was presented two new methods of estimating the global visual-motor Jacobian [7], the first one is a K-nearest neighbourhood regressor over Jacobian that uses previously estimated local models, the second method stores previous movements and computes an estimate of the Jacobian by solving a local least squares problem.

The all mentioned methods considerate a discrete counterpart model of equation (2) defined by the equations:

$$\Delta \mathbf{s_k} = \mathbf{J_k}\Delta \mathbf{q_k} \qquad \text{or} \qquad \Delta \mathbf{s_k} = \mathbf{J}\Delta \mathbf{q_k} \quad (4)$$

where $\mathbf{J}$ is assumed to be a constant.

A. Adding the Epipolar Constraint

Epipolar constraint (5) is taken into account in calculating the image Jacobian (2) (4). If the considered visual features are centroids of points, and if we note a point in the first camera by ('), and in the second camera by ("), the projection of a 3D point on both images must satisfy the epipolar restriction equation:

$$\tilde{\mathbf{s}}''^T_k \mathbf{F} \tilde{\mathbf{s}}'_k = 0 \quad \text{and} \quad \tilde{\mathbf{s}}''^T_{k-1} \mathbf{F} \tilde{\mathbf{s}}'_{k-1} = 0 \quad (5)$$

for moments k or $k-1$ respectively, where features are expressed in projective notation ($\sim$), and $\mathbf{F}$ is a 3x3 matrix known as the fundamental matrix. Its knowledge is known as weak or projective calibration. A more detailed description can be found in [8] and [10].

On the other hand, features at moments *k* and *k-1* for each camera, are given by:

$$\tilde{\mathbf{s}}'_k = \tilde{\mathbf{s}}'_{k-1} + \tilde{\mathbf{J}}' \Delta\mathbf{q}_k \quad ; \quad \tilde{\mathbf{s}}''_k = \tilde{\mathbf{s}}''_{k-1} + \tilde{\mathbf{J}}'' \Delta\mathbf{q}_k \qquad (6)$$

where $\Delta\mathbf{q}_k = \mathbf{q}_k - \mathbf{q}_{k-1}$, and $\tilde{\mathbf{J}}', \tilde{\mathbf{J}}''$ contain the variables to be solved, which are elements of image Jacobian for each camera and have the form (system of *n* DOF):

$$\tilde{\mathbf{J}}' = \begin{bmatrix} \mathbf{J}' \\ \mathbf{0}_{1_x n} \end{bmatrix} \quad ; \quad \tilde{\mathbf{J}}'' = \begin{bmatrix} \mathbf{J}'' \\ \mathbf{0}_{1_x n} \end{bmatrix} \qquad (7)$$

to do dimensionally correct equation (6), where $\mathbf{0}_{1,n}$ is a row vector of dimension *n*.

Substituting (6) in the second equation of (5), ordering terms and considering the first equation of (5), we have the following non-linear equation for $\tilde{\mathbf{J}}'$ and $\tilde{\mathbf{J}}''$ [13] [16]:

$$\Delta\mathbf{q}_k^T \tilde{\mathbf{J}}''^T \mathbf{F} \, \tilde{\mathbf{J}}' \Delta\mathbf{q}_k - \Delta\mathbf{q}_k^T \tilde{\mathbf{J}}''^T \mathbf{F} \tilde{\mathbf{s}}'_k - \tilde{\mathbf{s}}''^T_k \, \mathbf{F} \, \tilde{\mathbf{J}}' \Delta\mathbf{q}_k = 0 \ (8)$$

Equation (8) and linear equations (4) for each camera form a set of equations solved at every move applying Levenberg-Marquadt optimisation. The non-linear system is initialized with a Jacobian obtained by a linear method [18] and converging after a few iterations. To obtain the enough number of equations to solve the equations system, the last realized moves method [18] was applied as well as a reliability factor [16]. We must remark that a normalization of input data [9] (image points and articular increments) is carried out before calculation of the Jacobian in order to homogenize importance of each equation.

B. Recursive Least Squares (RLS) Methodt

In this method the Jacobian is estimated recursively by a least squares algorithm [1] [14], its equations are:

$$\mathbf{J}_k = \mathbf{J}_{k-1} + \frac{\left(-\Delta\mathbf{e}_k + \dfrac{\partial \mathbf{e}_k}{\partial t}\Delta t - \mathbf{J}_{k-1}\Delta\mathbf{q}_k \right)\Delta\mathbf{q}_k^T \mathbf{P}_{k-1}}{\lambda + \Delta\mathbf{q}_k^T \, \mathbf{P}_{k-1}\Delta\mathbf{q}_k} \qquad (9)$$

where $\mathbf{e}_k = \mathbf{s}_k^* - \mathbf{s}_k$ is the image features error, and $\mathbf{s}_k^*$ the desired features, and

$$\mathbf{P}_k = \frac{1}{\lambda}\left(\mathbf{P}_{k-1} - \frac{\mathbf{P}_{k-1}\Delta\mathbf{q}_k \Delta\mathbf{q}_k^T \mathbf{P}_{k-1}}{\lambda + \Delta\mathbf{q}_k^T \mathbf{P}_{k-1}\Delta\mathbf{q}_k} \right) \qquad (10)$$

is the covariance matrix. The behaviour of this method depends on the parameter λ, which varies in a range from 0 to 1, and ponders previous movements. λ settles a compromise between the information provided by old data from previous moves and new data, possibly corrupted by noise. In the presence of moderate noise, values of λ close to 0.9 are often used

IV. ANALYTIC IMAGE JACOBIAN

The analytic image Jacobian is given by [4]:

$$\mathbf{J}_A = \mathbf{J}_{sr}\mathbf{W}_{cr}\mathbf{N}\mathbf{J}_{rq} \qquad (11)$$

where $\mathbf{J}_{rq}$ is the robot Jacobian, $\mathbf{N}$ is a matrix that contains the direct kinematic of the robot:

$$\mathbf{N} = \begin{bmatrix} \mathbf{I}_3 & [\mathbf{t}_r]_x \\ \mathbf{0} & \mathbf{I}_3 \end{bmatrix} \qquad (12)$$

where $[\mathbf{t}_r]_x$ is the skew symmetric matrix of the position of the last link w.r.t. the base of the robot, $\mathbf{I}_3$ is the identity matrix of order 3. $\mathbf{W}_{cr}$ is the relationship between kinematic screws of the camera and the robot given by:

$$\mathbf{W}_{cr} = \begin{bmatrix} \mathbf{R}_{cr} & [\mathbf{t}_{cr}]_x\mathbf{R}_{cr} \\ \mathbf{0}_3 & \mathbf{R}_{cr} \end{bmatrix} \qquad (13)$$

being $\mathbf{R}_{cr}$ and $\mathbf{t}_{cr}$ rotation and translation between camera and the base of the robot, $[\mathbf{t}_{cr}]_x$ is the skew-symmetric matrix of $\mathbf{t}_{cr}$, and $\mathbf{J}_{sr}$ is the interaction matrix given by (1). According to section *IIA*, both $\mathbf{J}_{sr}$ and $\mathbf{W}_{cr}$ are different for each camera even though they both share $\mathbf{N}$ and $\mathbf{J}_{rq}$, thus there is a different analytic Jacobian for each camera, and the control of the system is realized by using both Jacobians together.

As can be seen in (11), analytic Jacobian depends on several parameters and transformations that become it very dependent on their accuracy, some of them are constants as the calibration of the cameras, and the camera-robot transform, others must be determined on line as the direct kinematic and the robot Jacobian.

V. CONTROL LAW

The task function $\mathbf{e}_k$ to be regulated to zero is given by:

$$\mathbf{e}_k = (\mathbf{s}^* - \mathbf{s}_k) \qquad (14)$$

where $\mathbf{s}^*$ and $\mathbf{s}_k$ are vectors of desired and current features at moment k respectively.

A proportional control law based on the pseudoinverse of the Jacobian [2] [4] was utilized to control the system, where the exponential decay of $\mathbf{e}_k$ to 0 is obtained by imposing $\dot{\mathbf{e}}_k = -\lambda\mathbf{e}_k$ (being λ is a positive scalar that regulates the convergence rate), the corresponding control law for the static (positioning) case is:

$$\mathbf{q}_{k+1} = \mathbf{q}_k + \lambda\mathbf{J}^+ \mathbf{e}_k \qquad (15)$$

being $\mathbf{J}^+ = (\mathbf{J}^T\mathbf{J})^{-1}\mathbf{J}^T$ the pseudoinverse of the Jacobian, since there are more features than number of DOF. It is proved that (15) has local asymptotic stability [2].

For the dynamic (tracking) case, a predictive term is added to (15) based on the last and the penultimate reference [14]:

$$\mathbf{q}_{k+1} = \mathbf{q}_k + \mathbf{J}^+ \left(\mathbf{s}_k^* - \mathbf{s}_k + \mathbf{s}_k^* - \mathbf{s}_{k-1}^* \right) \qquad (16)$$

In control laws (15) and (16), the estimated Jacobian and analytic Jacobian are used alternatively.

VI. EXPERIMENTS

In this section we describe our experimental equipment and results.

A. Experimental Setup

The system used in the experiments consists of:

1) A joint system: composed of a high precision positioning device and its controller, model Newport MM3000 (see Fig. 1). The system has 3 DOF: one prismatic and two revolute joints, and their theoretical precisions are of a thousandth of a millimeter and a thousandth of a degree. The visual control object, is made out of five black points on a white background, the projection of which on the image will be the control features, and has been attached to the last link of the joint system

2) An image acquisition and processing system: composed by two CV-M50 analogic cameras and a Matrox Meteor II-MC image acquisition board, which allows simultaneous acquisition from both cameras. The cameras, fixed in the working environment, are separated by about 700 millimeters, their both axes converge towards the joint system, and they are separated from it by about 1200 millimeters. Visual features are detected with sub-pixel precision, and due to the simplicity of the image, the error is estimated to be of less than 0.2 pixels. Tests are realized by using 2, 3, 4, or 5 points
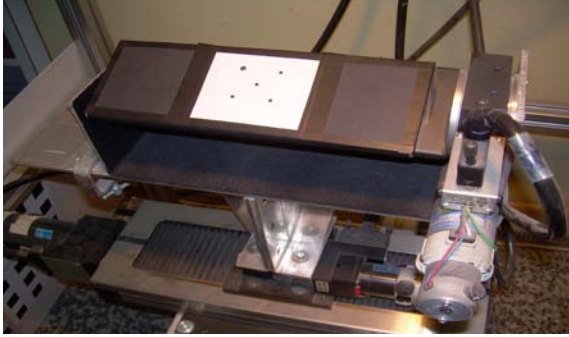

Fig. 1. Experimental setup.

B. Control Objective

We have contrasted the performance of the analytic Jacobian (ANLTC) method with two methods to estimate the image Jacobian: by integrating epipolar restriction (FUNDMAT) and by using recursive last square algorithm (RLS). Experiments consist of static (positioning) and dynamic (tracking) tasks using the respective control law, where the number of points for controlling the system is taken into account. Likewise tests were carried out without added noise and with added noise (Gaussian noise with $\sigma = 0.5$ pixel) in detecting features. Image features are centroids of projected points (Fig. 1). Visual features must be reachable and the visual object must be visible from both views. Due to the fact that the joint system has only 3 DOF, and to ensure coherence, we have obtained visual features for all desired positions previously from a teach-by-showing technique [12] where the joint system is moved to a desired

position and its corresponding image features are recorded. In this way, values of desired features ($\mathbf{q}*$) are obtained, which are utilized not to control the system but only in the evaluation of the control.

1) Static case: starting from an initial position, the system has to achieve desired features consecutively (Fig.2). A trajectory will be generated in both image planes and in the joint space. If the error (Euclidean distance) between current and desired features is less than 0.6 pixels, it is meant that desired features have been reached. A high number of positions (up to 50) obtained randomly in the whole workspace are linked in order to obtain more representative results

2) Dynamic case: The system has to follow some image features belonging to a curve trajectory set in advance into the workspace (Fig. 3) built from random parameters. Similar to static case, in order to obtain more representative results, a high number of trajectories to be tracked are generated into the workspace.

C. Evaluation Indices

In order to evaluate the performance of the methods, we consider an index to be measured from image plane Index 0, and two indices to be measured from joint space Index 1s, and Index 1d. Index 0 is utilized either for static or for dynamic case, Index 1s for static case and Index 1d for dynamic case. It is worth remarking that low values of indices mean better performance. Indices are defined as follows:

1) Index 0: Sum of Euclidean distances between desired and current visual features. It is weighted by number of considered points to control the system, number of cameras, and number of desired positions:

$$ind\_0 = \frac{1}{CTPU} \sum_{i=1}^{T} \sum_{j=1}^{P} \sum_{k=1}^{u} \left(\begin{matrix} \left\| \mathbf{s}*'(i,j) - \mathbf{s'}_k(i,j) \right\| + \\ + \left\| \mathbf{s}*''(i,j) - \mathbf{s''}_k(i,j) \right\| \end{matrix} \right) \quad (17)$$

$\mathbf{s'}_k(i,j)$, $\mathbf{s}*'_k(i,j)$: current and desired features in the first image at moment *k,* point number *j,* and trajectory *i*.
$\mathbf{s''}_k(i,j)$, $\mathbf{s}*''_k(i,j)$: current and desired features in the second image at moment *k,* point number *j,* and trajectory *i*.
u: number of movements to reach desired features (its value depends on the performance of the method)
U: maximum number of movements to reach desired feature (U=30)
P: number of considered points (P=2,3,4,5).
T: number of desired positions (T=50).
C: number of cameras (C=2).

2) Index 1s: Sum of Euclidean distances in joint space for all of the movements performed. It is weighted by number of desired positions:

$$ind\_1s = \frac{1}{T} \sum_{i=1}^{T} \sum_{k=1}^{u} \left\| \mathbf{q}_k(i) - \mathbf{q}_{k-1}(i) \right\| \quad (18)$$

$q_k(i)$, $q_{k-1}(i)$:positions at moment *k* and *k-1* respectively to reach desired position *i*. Joint values are expressed by either a thousandth of a millimeter or a thousandth of a degree. It is worth remarking that desired joint position is not utilized to

generate the algorithm of Jacobian estimation, but it is only utilized to calculate the effectiveness of the algorithm by means of this index.

3) Index 1d: Sum of Euclidean distances between desired and current joint positions. It is weighted by number of desired positions:

$$ind\_1d = \frac{1}{T}\sum_{i=1}^{T}\left\|\mathbf{q}^*(i) - \mathbf{q}(i)\right\| \qquad (19)$$

$\mathbf{q}^*(i)$, $\mathbf{q}(i)$: desired and current joint positions respectively

D. Results

Numerical results are resumed in Table I and Table II.

TABLE I
INDICES 0 AND 1S, WITHOUT AND WITH ADDED NOISE. STATIC CASE

IN DEX	ALGO RITHM	CONDITIONS							
		WITHOUT ADDED NOISE				WITH ADD. NOISE $\sigma = 0.5$			
		CONSIDERED POINTS				CONSIDERED POINTS			
		2	3	4	5	2	3	4	5
0	FUNDMAT	170.4	171.5	169.7	168.6	175.7	176.5	176.7	177.9
	RLS	174.9	171.8	172.3	171.0	179.7	181.1	179.2	177.3
	ANLTC	165.6	165.6	165.6	166.3	170.7	169.9	170.4	171.6
1s*	FUNDMAT	32.39	32.46	32.34	32.05	34.91	35.74	34.36	35.07
	RLS	32.98	32.24	32.24	32.34	34.69	36.66	35.64	35.52
	ANLTC	31.72	31.77	31.75	31.68	33.81	32.93	33.73	33.62

* divided by 1000.

TABLE II
INDICES 0 AND 1D, WITHOUT AND WITH ADDED NOISE. DYNAMIC CASE

IN DEX	ALGO RITHM	CONDITIONS							
		WITHOUT ADDED NOISE				WITH ADD. NOISE $\sigma = 0.5$			
		CONSIDERED POINTS				CONSIDERED POINTS			
		2	3	4	5	2	3	4	5
0	FUNDMAT	15.14	15.95	16.17	15.98	15.36	16.24	16.45	16.27
	RLS	15.58	16.02	16.35	16.15	15.80	17.12	17.57	17.95
	ANLTC	15.00	15.89	16.10	15.92	15.30	16.20	16.44	16.33
1d*	FUNDMAT	0.30	0.39	0.31	0.22	0.45	0.56	0.43	0.34
	RLS	0.52	0.33	0.52	0.38	1.66	1.70	2.41	3.22
	ANLTC	0.29	0.31	0.29	0.24	0.32	0.30	0.27	0.27

* divided by 1000.

Table I and Table II show that ANLTC and FUNDMAT methods work quite similar especially for dynamic case, in general both methods work quite well. RLS shows the least good performance. Similar graphs for the case without added noise can be also drawn. Fig. 2 and Fig. 3 show the articular evolution for FUNDMAT method with added noise, static and dynamic case respectively. Red circles represent desired positions to be reached, and the evolution of the joint system is drawn in blue lines

1) Perturbation of transforms: It is well-known that analytic Jacobian has a strong dependency of parameters that have to be calculated both on line and off line prior to the control task (see equation (11)). In this sense, in order to evaluate this dependence, the analytic Jacobian has been calculated based on:

- Degrading the depth Z of the 3D point to be detected in about 2%, up to 5% (Table III and Table IV). A constant depth equal to the desired position which is a common solution in visual servoing [2] [6].

- Degrading $\mathbf{R}_{cr}$ and $\mathbf{t}_{cr}$ of the camera-robot transform (equation (13)) in about 2% up to 4% (Table V and

Table VI). Where a random perturbation of both the elements of $\mathbf{t}_{cr}$ and Euler angles of $\mathbf{R}_{cr}$ was applied.
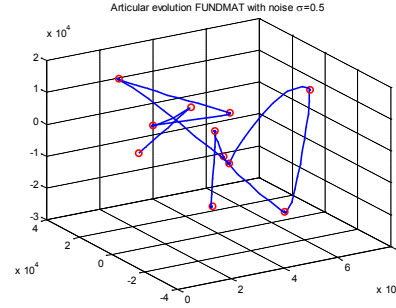

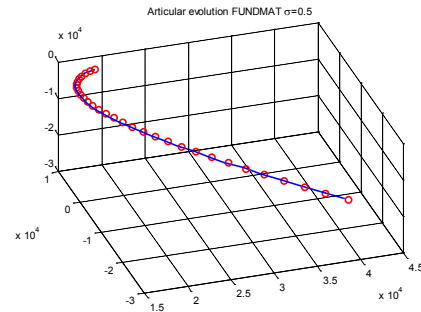Fig. 4. Articular evolution for ten desired position, static case with added noise (FUNDMAT)

Fig. 5. Articular evolution for dynamic case with noise (FUNDMAT)

In this way, tests were also carried out with this degraded analytic Jacobian. It was observed that increasing these levels of degradation the system is no longer controlled.

Table III and Table V show results for static case, whereas Table IV and Table VI for dynamic case. The notation is as follows: ANLTC Z2%, ANLTC Z3%, ANLTC Z4%, ANLTC Z5%, degradation of the depth of the detected 3D point in 2%, 3%, 4%, 5% respectively, for the calculation of the analytic Jacobian, and ANLTC Zd, calculation of the analytic Jacobian with depth constant and equal to the depth of the desired 3D point, ANLTC W2%, ANLTC W3%, and ANLTC W4%, perturbation in 2%, 3%, and 4% of the camera-robot transformation.

TABLE III
INDICES 0 AND 1S, WITHOUT AND WITH ADDED NOISE. STATIC CASE.
PERTURBATION OF DEPTH OF THE DETECED 3D POINT

IN DE X	ANLTC ALGO- RITHM	CONDITIONS							
		WITHOUT ADDED NOISE				WITH ADD. NOISE $\sigma = 0.5$			
		CONSIDERED POINTS				CONSIDERED POINTS			
		2	3	4	5	2	3	4	5
0	Z2%	204.4	183.5	186.7	182.5	211.0	189.3	192.1	190.9
	Z3%	219.7	192.4	201.9	194.6	223.7	197.7	204.7	198.2
	Z4%	239.4	212.9	225.7	215.7	243.5	217.5	228.0	219.6
	Z5%	259.2	233.5	249.5	236.9	263.3	237.4	251.4	241.0
	Zd	292.8	175.7	188.4	179.8	296.2	179.8	191.9	182.7
1s*	Z2%	35.51	36.23	35.92	36.34	42.21	40.01	38.73	38.11
	Z3%	37.47	37.50	38.20	37.45	43.02	41.38	40.41	39.82
	Z4%	37.66	38.59	39.00	38.68	43.20	42.09	41.22	40.81
	Z5%	37.85	39.69	39.81	39.92	43.38	42.81	42.04	41.80
	Zd	46.94	35.85	37.80	36.56	50.23	39.24	40.79	39.68

* divided by 1000.

TABLE IV
INDICES 0 AND 1D, WITHOUT AND WITH ADDED NOISE. DYNAMIC CASE.
PERTURBATION OF DEPTH OF THE DETECED 3D POINT

INDEX	ANLTC ALGO-RITHM	CONDITIONS							
		WITHOUT ADDED NOISE				WITH ADD. NOISE $\sigma=0.5$			
		CONSIDERED POINTS				CONSIDERED POINTS			
		2	3	4	5	2	3	4	5
0	Z2%	17.12	17.77	18.12	17.95	17.02	17.33	17.59	17.41
	Z3%	17.74	18.51	18.89	18.64	17.82	18.52	18.92	18.66
	Z4%	18.24	19.18	19.56	19.41	18.28	19.19	19.55	19.44
	Z5%	18.74	19.85	20.23	20.19	18.75	19.87	20.19	20.22
	Zd	18.00	17.40	18.14	17.58	18.03	17.44	18.16	17.63
Id*	Z2%	2.22	2.67	2.48	2.73	4.03	2.34	2.69	2.53
	Z3%	4.65	3.33	3.71	2.92	4.74	3.33	3.71	2.93
	Z4%	5.01	3.89	4.27	3.49	5.05	3.89	4.24	3.49
	Z5%	5.38	4.46	4.83	4.06	5.37	4.45	4.78	4.05
	Zd	6.12	2.70	3.39	2.39	6.15	2.71	3.39	2.39

* divided by 1000.

TABLE V
INDICES 0 AND 1S, WITHOUT AND WITH ADDED NOISE. STATIC CASE.
PERTURBATION OF CAMERA-ROBOT TRANSFORM

INDEX	ANLTC ALGO-RITHM	CONDITIONS							
		WITHOUT ADDED NOISE				WITH ADD. NOISE $\sigma=0.5$			
		CONSIDERED POINTS				CONSIDERED POINTS			
		2	3	4	5	2	3	4	5
0	W2%	184.4	175.7	177.9	172.8	174.6	165.4	166.3	166.2
	W3%	226.9	180.3	186.5	179.7	228.4	185.0	189.8	183.3
	W4%	323.2	213.4	237.6	215.6	325.9	215.0	238.6	219.2
Is*	W2%	39.85	34.08	34.64	33.66	41.26	35.66	36.29	35.96
	W3%	45.05	36.43	38.35	36.09	45.90	38.10	39.41	37.51
	W4%	44.97	38.96	43.04	39.03	45.10	39.55	43.58	39.78

* divided by 1000.

TABLE VI
INDICES 0 AND 1S, WITHOUT AND WITH ADDED NOISE. DYNAMIC CASE.
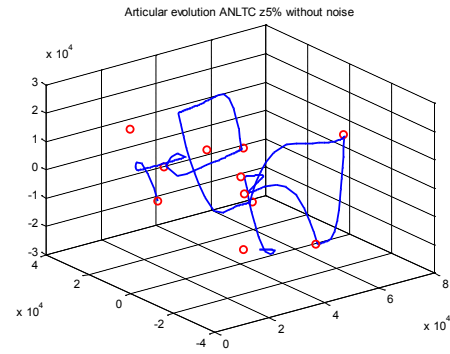PERTURBATION OF CAMERA-ROBOT TRANSFORM

| INDEX	ANLTC ALGO-RITHM	CONDITIONS							
		WITHOUT ADDED NOISE				WITH ADD. NOISE $\sigma=0.5$			
		CONSIDERED POINTS				CONSIDERED POINTS			
		2	3	4	5	2	3	4	5
0	W2%	16.52	17.01	17.29	17.05	16.61	17.14	17.34	17.12
	W3%	17.9	18.11	18.55	18.27	17.97	18.17	18.63	18.27
	W4%	18.78	19.19	20.09	19.59	18.80	19.21	20.09	19.62
Id*	W2%	3.78	2.68	2.53	2.03	3.74	2.65	2.53	2.03
	W3%	6.55	4.66	4.66	3.81	6.56	4.66	4.69	3.85
	W4%	8.97	6.95	7.60	6.22	8.95	6.99	7.59	6.22

* divided by 1000.

Tables III-VI show that with these levels of perturbations either depth Z or camera-robot transform, the behaviour of ANLTC is also degraded. Both static and dynamic cases show the same tendency. Furthermore, these figures show that RLS has less good behaviour than ANLTC and FUNDMAT.

Results with added noise are quite similar, that means that these kinds of perturbation affect to the analytic Jacobian much more than noise.

Fig. 4 and Fig. 5b show the articular evolution without noise of some degraded analytic Jacobian. Articular evolution with added noise static case for RLS is also shown in Fig. 5a
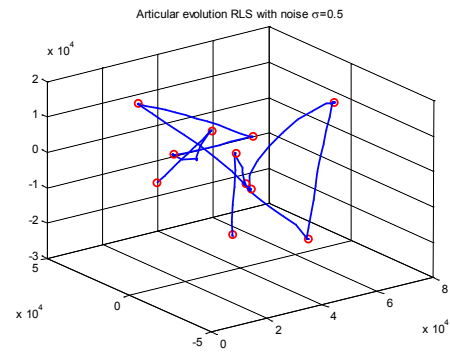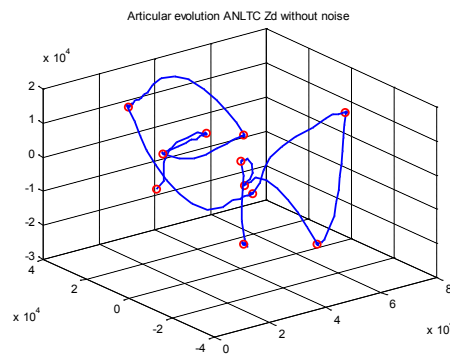

a)
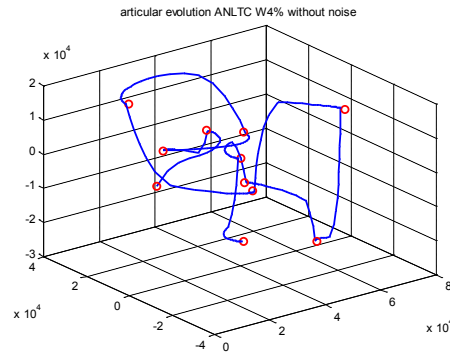
b)

Fig. 4. Evolution for ten desired positions, static case without noise: a)ANLTC with degradation of depth Z up to 5%, b) ANLTC with perturbation of camera-robot transform up to 4%.

a)

b)

Fig. 5. Evolution for ten desired positions, static case: a)RLS with added noise, b) ANLTC Zd without noise

Fig.4a and Fig.4b show an important degradation of the trajectory for ANLTC Z5% and ANLTC W4% respectively even noise was not added. Trajectory of RLS (Fig 5a) with added noise has not the same good behaviour as ANLTC and FUNDMAT (Fig.2). Trajectory of ANLTC Zd (Fig. 5b) has no good performance even without noise. Joint evolution for dynamic case is drawn in Fig. 6, which show that evolution in blue does not follows red circles trajectory even noise is not added.

Articular evolution ANLTC z5% without added noise

a)

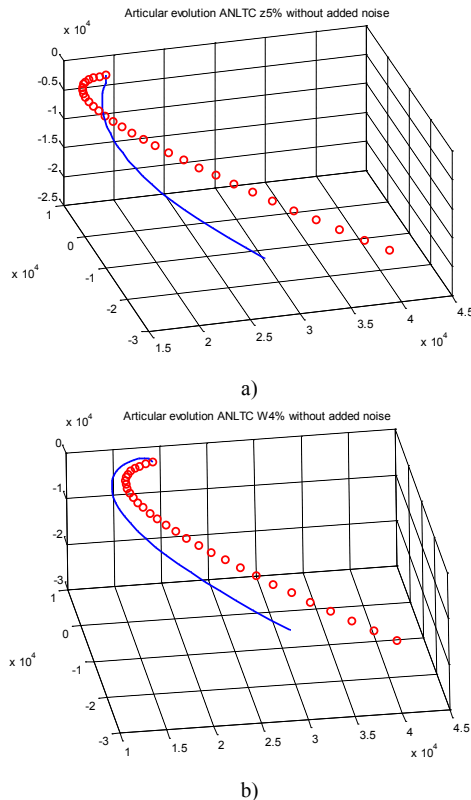Articular evolution ANLTC W4% without added noise

b)

Fig. 6. Articular evolution without added noise for dynamic case: a)ANLTC with perturbation of depth Z 5%, b) ANLTC with perturbation of camera-robot transform 4%.

2) Incrementing level of added noise: Additional tests were carried out by incrementing the level of added noise up to σ =1. The results showed that in all cases for te performance of the analytic Jacobian, the level of degradation are more important than the levels of added noise, whereas at these same levels of noise, the performance of the estimated Jacobian FUNDMAT is not affected.

VII. CONCLUSION

A performance comparison has been carried out between a control task that comes from the estimation of the image Jacobian (represented by both FUNDMAT that integrates the fundamental matrix and RLS the recursive least square method) and that one that comes from a calculation (ANLTC, analytic Jacobian method). Tests showed that in absence of noise, behavior of these three methods work quite similar. But in tests with added noise, FUNDMAT and ANLTC showed to be very robust. Analytic Jacobian works well, but at the expense of doing laborious previous work to the control task, it needs the camera calibration, 3D reconstruction of the point, the camera-robot transformation, and the robot Jacobian. Tests with perturbation of either camera-robot transformation or depth of the detected 3D point showed that performance of the analytic Jacobian is degraded even low values of perturbation.

Considering the good performance of the method that estimates the image Jacobian by integrating the fundamental matrix, and the fact that this method does not need previous work to the control task in contrast to the analytic Jacobian does, it is considered as an advantage. This fact becomes it to be appropriate for tasks in unknown or changing environments because the knowledge of the fundamental matrix is not an objection, and its calculation has been proven to be much more simple, robust and reliable. The future work is the analysis of the system stability with the control law generated from the estimated Jacobian..

REFERENCES

[1] .Asada, T. Tanaka, K. Hosoda, "Adaptive Binocular Visual Servoing for Independently Moving Target Tracking", Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'00), pp. 2076-2081, 2000.
[2] F. Chaumette, S. Hutchinson, "Visual Servo Control Part I: Basic Approaches", IEEE Robotics & Automation Magazine, vol.14, issue 1, pp. 109-118, 2006.
[3] F. Chaumette, S. Hutchinson, "Visual Servo Control Part II: Advanced Approaches", Robotics & Automation Magazine, IEEE, vol.13, issue 4, pp. 82-90, 2007.
[4] P. Corke, "Visual Control of Robot Manipulators – A Review", K. Hashimoto Ed., Visual Servoing, World Scientific, pp. 1-32,1994.
[5] Z. Deng, M. Jägersand, "Evaluation of Model Independent Image-Based Visual Servoing", Canadian Conference on Computer and Robot Vision, pp. 138-144,2004.
[6] B. Espiau, F. Chaumette, P. Rives, "A New Approach to Visual Servoing in Robotics", IEEE Transactions on Robotics and Automation, 8(3), pp. 313-326, 1992.
[7] A. Farahmand, A. Shademan, and M. Jagersand, "Global Visual-Motor Estimation Visual Servoing", Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1969-1974, 2007.
[8] O. Faugeras, Q.T. Luong, The Geometry of Multiple Images, The Massachusetts Institute of Technology Press, 2001.
[9] R.I. Hartley, "In Defence of the Eight-points Algorithm", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, Nº 6, 1997.
[10] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision. 2nd Ed., Cambridge University Press, 2003.
[11] S.A. Hutchinson, G.D. Hager, P.I. Corke, "A tutorial on visual servo control, IEEE Trans. Robotics and Automation, 12 pp. 651-670, 1996.
[12] D. Kragic, H.I. Christensen, "Survey on visual servoing for manipulation", Technical Report ISRN KTH/NA/P-02/01-Sen, CVAP259, 2002.
[13] L. Pari, J.M. Sebastián, C. González, L. Ángel, "Image Based Visual Servoing: A New Method for the Estimation of the Image Jacobian in Dynamic Environments", Lecture Notes in Computer Science, Vol 4142 pp. 850-861, 2006.

[14] J.A. Piepmeier, G.V. McMurray, H. Lipkin, "Uncalibrated Dynamic Visual Servoing", IEEE Transactions on Robotics and Automation, 20 (1) pp. 143-147, 2004.

[15] J. Qian, J. Su, "Online estimation of Image Jacobian Matrix by Kalman-Bucy filter for uncalibrated Stereo Vision Feedback", International Conference on Robotics & Automation (ICRA'02), pp. 562-567, 2002.

[16] J.M. Sebastián, L. Pari, C. González, L. Ángel, "A New Method for the Estimation of the Image Jacobian for the Control of an Uncalibrated Joint System", Lecture Notes in Computer Science, Vol. 3522, pp. 631-638, 2005.

[17] J.M. Sebastian, L. Pari, L. Angel, A. Traslosheros, "Uncalibrated Visual Servoing Using the Fundamental Matrix", Robotics and Autonomous System 57 (1), pp. 1-10, 2009.

[18] H. Sutanto, R. Sharma, V. Varma, "The role of exploratory movement in visual servoing without calibration", Robotics and Autonomous Systems 23, pp. 153-169, 1998.