Improving vision-based control using efficient second-order minimization techniques

Ezio Malis

Improving vision-based control using efficient second-order minimization techniques

Ezio Malis
I.N.R.I.A. Sophia Antipolis
2004, route des Lucioles, France

***Abstract*— In this paper, several vision-based robot control methods are classified following an analogy with well known minimization methods. Comparing the rate of convergence between minimization algorithms helps us to understand the difference of performance of the control schemes. In particular, it is shown that standard vision-based control methods have in general low rates of convergence. Thus, the performance of vision-based control could be improved using schemes which perform like the Newton minimization algorithm that has a high convergence rate. Unfortunately, the Newton minimization method needs the computation of second derivatives that can be ill-conditioned causing convergence problems. In order to solve these problems, this paper proposes two new control schemes based on efficient second-order minimization techniques.**

I. Introduction

Vision-based control is a very flexible method for positioning a robot using visual feedback. A positioning task is generally formulated as an output regulation problem using a teaching-by-showing technique [2]. The robot is positioned when current image features reach their corresponding reference values. The control problem can also be formulated as a nonlinear least squares minimization problem. Thus, vision-based robot control methods can be classified following an analogy with minimization methods. The classification will be limited to methods which use cost function derivatives. In order to avoid non-smooth robot motion, methods needing multiple evaluations of the cost function, like the simplex method [3], will not be considered here. Indeed, the evaluation of the cost function is equivalent to moving the robot to a given position. Comparing the rate of convergence between minimization algorithms, help us to understand the difference of performance of the control schemes [1]. In this paper, it will be shown that standard vision-based control methods [2] correspond to minimization methods that have a low convergence rate. Thus, the performance of vision-based control can be improved using schemes performing like the Newton minimization algorithm which has higher convergence rate. Unfortunately, the Newton minimization method requires the computation of second derivatives that can be ill-conditioned causing convergence problems. For these reasons, two new control schemes with high convergence rates are proposed in the paper. The two control schemes are computationally efficient since they need only first derivatives. The new control methods can be used with any visual servoing scheme. As an example, they are applied to image-based visual servoing for solving the well known advance/retreat problem [4].

II. Theoretical Background

Let $\mathbf{x} \in \mathbb{R}^6$ be a vector containing the global coordinates of the end-effector frame in an open subset of $\mathbb{R}^3 \times SO(3)$. Let $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{v})$ be the kinematic equation linking the control input $\mathbf{v}$ to the derivative of the state vector. Consider now the task of positioning the robot end-effector frame using some (n×1) output vector $\mathbf{s}(\mathbf{x})$.

A. Formulating the positioning as a control problem

Moving the robot to a reference position $\mathbf{x}_1$ starting from an initial position $\mathbf{x}_2$ can be formulated as an *output regulation* problem. The control problem consists in finding a feedback law $\mathbf{v}$ such that the output $\mathbf{s}(\mathbf{x})$ reaches a desired output $\mathbf{s}(\mathbf{x}_1)$.

B. Formulating the positioning as a minimization problem

The output regulation problem can also be viewed as a Nonlinear Least Squares (NLS) minimization. The problem can be formulated in two different ways. The first is to move the current frame towards the reference frame solving the problem P_1 starting from $\mathbf{x}_2$:

$$\min_{\mathbf{x}} f_1(\mathbf{x}) = \frac{1}{2}(\mathbf{s}(\mathbf{x}) - \mathbf{s}(\mathbf{x}_1))^\top (\mathbf{s}(\mathbf{x}) - \mathbf{s}(\mathbf{x}_1)) \qquad (1)$$

The second is to "virtually" move the reference frame towards the current frame solving the problem P_2 starting from $\mathbf{x}_1$:

$$\min_{\mathbf{x}} f_2(\mathbf{x}) = \frac{1}{2}(\mathbf{s}(\mathbf{x}) - \mathbf{s}(\mathbf{x}_2))^\top (\mathbf{s}(\mathbf{x}) - \mathbf{s}(\mathbf{x}_2)) \qquad (2)$$

The reference only "virtually" moves since the current position is updated. The difference between the two problems will become clear in the following section. Several minimization methods can be used to solve the problems. The performance of a minimization method can be measured by its asymptotic convergence rate r (for a rigorous definition see [6]). Without going into further details, the bigger is r the faster the algorithm will converge to the solution. Consider for example a cost function $f(\mathbf{x})$ quadratic in $\mathbf{x}$. Then, an algorithm with "quadratic convergence" ($r = 2$) will find the exact solution in only one step. On the other hand, an algorithm with "linear convergence" ($r = 1$) will need an infinite number of iterations. Finally, an algorithm with "super-linear convergence" will converge after a finite number of steps. Note that, in the case of visual servoing, a high convergence rate implies a camera trajectory closer to the geodesic in $\mathbb{R}^3 \times SO(3)$ (i.e. a straight line for the translation and a rotation around the axis of rotation). Indeed, being able to go in one step to the solution means we know exactly the displacement of the camera and we obtain better 3D trajectories.

III. Analogy between control and minimization

In this section, several vision-based robot control methods are classified following an analogy with well known minimization methods. For sake of simplicity, only the first step of the minimization is considered. Once the current position updated, the step is repeated until convergence.

A. Steepest Descent (Jacobian Transpose)

The Steepest Descent minimization method (SDM) is based on the first-order Taylor series of the cost function. For the problem P_1, consider first-order Taylor series of the cost function $f_1(\mathbf{x})$ about $\mathbf{x}_2$, evaluated at $\mathbf{x}_1$:

$$f_1(\mathbf{x}_1) \approx f_1(\mathbf{x}_2) + \left.\frac{\partial f_1(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}_2} \Delta\mathbf{x}$$

where $\Delta\mathbf{x} = \mathbf{x}_1 - \mathbf{x}_2$. For P_2, consider the first-order Taylor series of $f_2(\mathbf{x})$ about $\mathbf{x}_1$, evaluated at $\mathbf{x}_2$:

$$f_2(\mathbf{x}_2) \approx f_2(\mathbf{x}_1) - \left.\frac{\partial f_2(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}_1} \Delta\mathbf{x}$$

Obviously, the gradients of the two functions are different. Indeed, after setting $\mathbf{J}(\mathbf{x}) = \frac{\partial \mathbf{s}(\mathbf{x})}{\partial \mathbf{x}}$ we obtain:

$$\begin{aligned} \left.\frac{\partial f_1(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}_2} &= +\Delta\mathbf{s}^\top \mathbf{J}(\mathbf{x}_2) \\ \left.\frac{\partial f_2(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}_1} &= -\Delta\mathbf{s}^\top \mathbf{J}(\mathbf{x}_1) \end{aligned}$$

where $\Delta\mathbf{s} = \mathbf{s}(\mathbf{x}_2) - \mathbf{s}(\mathbf{x}_1)$. The strategy of the SDM is to move in the opposite direction to the gradient. Thus, the displacements $\Delta\mathbf{x}$ to solve P_1 and P_2 are respectively:

$$\Delta\mathbf{x} = -\lambda \left.\frac{\partial f_1(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}_2}^\top = -\lambda\, \mathbf{J}^\top(\mathbf{x}_2)\, \Delta\mathbf{s} \tag{3}$$

$$\Delta\mathbf{x} = +\lambda \left.\frac{\partial f_2(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}_1}^\top = -\lambda\, \mathbf{J}^\top(\mathbf{x}_1)\, \Delta\mathbf{s} \tag{4}$$

The positive gain λ tunes the amplitude of the displacement. For P_2, the Jacobian is constant since it is computed at the reference position $\mathbf{x}_1$ while for P_1 the Jacobian is varying since it is computed at the current position $\mathbf{x}_2$. In robot control theory the SDM corresponds to the Jacobian Transpose control method (JTC). For example, the constant JTC has been used in [7] while the varying JTC has been used in [8]. The SDM has a slow (linear) convergence rate.

B. Newton

The Newton minimization method (NM) is based on the second-order Taylor series of the cost function. For the problem P_1, consider the second-order Taylor series of the cost function $f_1(\mathbf{x})$ about $\mathbf{x}_2$, evaluated at $\mathbf{x}_1$:

$$f_1(\mathbf{x}_1) \approx f_1(\mathbf{x}_2) + \left.\frac{\partial f_1(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}_2} \Delta\mathbf{x} + \frac{1}{2}\Delta\mathbf{x}^\top \left.\frac{\partial^2 f_1(\mathbf{x})}{\partial \mathbf{x}^2}\right|_{\mathbf{x}_2} \Delta\mathbf{x}$$

For P_2, consider the second-order Taylor series of the cost function $f_2(\mathbf{x})$ about $\mathbf{x}_1$, evaluated at $\mathbf{x}_2$:

$$f_2(\mathbf{x}_2) \approx f_2(\mathbf{x}_1) - \left.\frac{\partial f_2(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}_1} \Delta\mathbf{x} + \frac{1}{2}\Delta\mathbf{x}^\top \left.\frac{\partial^2 f_2(\mathbf{x})}{\partial \mathbf{x}^2}\right|_{\mathbf{x}_1} \Delta\mathbf{x}$$

where the Hessian matrices of the cost functions $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ are respectively:

$$\left.\frac{\partial^2 f_1(\mathbf{x})}{\partial \mathbf{x}^2}\right|_{\mathbf{x}_2} = \mathbf{J}^\top(\mathbf{x}_2)\mathbf{J}(\mathbf{x}_2) + \sum_{k=0}^{n} \mathbf{H}_k(\mathbf{x}_2)\, \Delta s_k \tag{5}$$

$$\left.\frac{\partial^2 f_2(\mathbf{x})}{\partial \mathbf{x}^2}\right|_{\mathbf{x}_1} = \mathbf{J}^\top(\mathbf{x}_1)\mathbf{J}(\mathbf{x}_1) + \sum_{k=0}^{n} \mathbf{H}_k(\mathbf{x}_1)\, \Delta s_k \tag{6}$$

$\mathbf{H}_k(\mathbf{x})$ being the Hessian matrix of function $s_k(\mathbf{x})$. The solutions for P_1 and P_2 are respectively:

$$\begin{aligned} \Delta\mathbf{x} &\approx -\lambda \left(\left.\frac{\partial^2 f_1(\mathbf{x})}{\partial \mathbf{x}^2}\right|_{\mathbf{x}_2} \right)^{-1} \left.\frac{\partial f_1(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}_2}^\top \qquad (7) \\ &= -\lambda \left(\mathbf{J}^\top(\mathbf{x}_2)\mathbf{J}(\mathbf{x}_2) + \sum_{k=0}^{n} \mathbf{H}_k(\mathbf{x}_2)\, \Delta s_k \right)^{-1} \mathbf{J}^\top(\mathbf{x}_2)\Delta\mathbf{s} \\ \Delta\mathbf{x} &\approx +\lambda \left(\left.\frac{\partial^2 f_2(\mathbf{x})}{\partial \mathbf{x}^2}\right|_{\mathbf{x}_1} \right)^{-1} \left.\frac{\partial f_2(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}_1}^\top \qquad (8) \\ &= -\lambda \left(\mathbf{J}^\top(\mathbf{x}_1)\mathbf{J}(\mathbf{x}_1) + \sum_{k=0}^{n} \mathbf{H}_k(\mathbf{x}_1)\, \Delta s_k \right)^{-1} \mathbf{J}^\top(\mathbf{x}_1)\Delta\mathbf{s} \end{aligned}$$

In the NM, the direction of descent of the SDM, defined by the gradient, is modified by the inverse of the Hessian. The convergence rate of NM is quadratic. Thus, if the function to minimize is concave quadratic, the NM method will attain the minimum in a single step. The minimum for the approximated problem is obtained for $\lambda = 1$. If the original cost function is not quadratic, the minimum for the approximated problem does not correspond to the minimum of the original cost function. Thus, it can be preferable to modulate the amplitude of the displacement by setting λ to a different value. The NM has a much faster convergence rate than SDM but may not converge when the Hessian is negative definite. Another drawback of the NM method can be the computational burden for the computation of the Hessian.

C. Approximated Newton

Approximated Newton methods are based on the modification of the direction of descent defined by the gradient with an approximation of the Hessian of the function. To overcome NM convergence problems, the approximation of the Hessian is chosen to be always definite positive. The price to pay is a slower convergence rate depending on the accuracy of the approximation.

1) Gauss-Newton (Jacobian pseudo-inverse):

The Gauss-Newton minimization method (GNM) takes into consideration the special structure of the NLS minimization. It is based on the first-order Taylor series of the vector function $\mathbf{s}(\mathbf{x})$. For the P_1 and P_2 problems we have respectively:

$$\Delta\mathbf{s} \approx -\mathbf{J}(\mathbf{x}_2)\Delta\mathbf{x} \tag{9}$$

$$\Delta\mathbf{s} \approx -\mathbf{J}(\mathbf{x}_1)\Delta\mathbf{x} \tag{10}$$

Let the Jacobian have full rank. The displacement can be computed using the pseudo-inverse of the Jacobian matrix

$\mathbf{J}^+ = (\mathbf{J}^\top\mathbf{J})^{-1}\mathbf{J}^\top$. For this reason, the GNM method is the equivalent to the Jacobian Pseudo-inverse control method (JPC). We obtain for the problems P_1 and P_2 respectively:

$$\Delta\mathbf{x} \approx -\lambda(\mathbf{J}^\top(\mathbf{x}_2)\mathbf{J}(\mathbf{x}_2))^{-1}\,\mathbf{J}^\top(\mathbf{x}_2)\,\Delta\mathbf{s} = -\lambda\mathbf{J}^+(\mathbf{x}_2)\,\Delta\mathbf{s} \quad (11)$$

$$\Delta\mathbf{x} \approx -\lambda(\mathbf{J}^\top(\mathbf{x}_1)\mathbf{J}(\mathbf{x}_1))^{-1}\,\mathbf{J}^\top(\mathbf{x}_1)\,\Delta\mathbf{s} = -\lambda\mathbf{J}^+(\mathbf{x}_1)\,\Delta\mathbf{s} \quad (12)$$

It must be noticed that the positive symmetric matrix $\mathbf{J}^\top(\mathbf{x})\mathbf{J}(\mathbf{x})$ is an approximation of the Hessian matrix. The GNM method achieves quadratic convergence when the vector $\mathbf{s}(\mathbf{x})$ is linear on $\mathbf{x}$ (i.e. the cost function is quadratic), otherwise, the convergence is only linear. Both constant and varying JPC methods have been used in [9].

2) Levenberg-Marquardt (Damped Least Squares):

The Levenberg-Marquardt minimization method (LMM) can be thought of as a method which allows to smoothly pass from Gauss-Newton method to the Steepest Descent method. The SDM will be used far from the minimum, but when we get closer to it we will use the GNM. The LMM is based on the following approximations:

$$\Delta\mathbf{x} \approx -\lambda(\mathbf{J}(\mathbf{x}_1)^\top\mathbf{J}(\mathbf{x}_1) + \gamma\,\mathbf{D})^{-1}\mathbf{J}(\mathbf{x}_1)^\top\,\Delta\mathbf{s} \quad (13)$$

$$\Delta\mathbf{x} \approx -\lambda(\mathbf{J}(\mathbf{x}_2)^\top\mathbf{J}(\mathbf{x}_2) + \gamma\,\mathbf{D})^{-1}\mathbf{J}(\mathbf{x}_2)^\top\,\Delta\mathbf{s} \quad (14)$$

Several choices for the diagonal matrix $\mathbf{D}$ are possible. The simplest one is to set the matrix equal to the identity (Levenberg [10]), while a more sophisticated one is to set the matrix equal to the entries on the diagonal of $\mathbf{J}^\top\mathbf{J}$ (Marquardt [11]). For a large γ the method approaches the Steepest Descent method, while for a small γ the method approaches the Gauss-Newton method. In robotics, the LMM method is called Damped Least Squares control method (DLSC) [12] [13]. Since the LMM method is a blending of methods with linear convergence it cannot achieve quadratic convergence. However, the DLSC has been used in robotics since it can help avoiding problems with singularities (the matrix $\mathbf{J}^\top\mathbf{J}$ is not invertible if the Jacobian is not full rank) and it gives a numerically stable method.

3) Quasi-Newton:

The goal of the Quasi-Newton minimization methods (QNM) is to build a sequence of symmetric positive definite matrices $\mathbf{A}_k$ such that the sequence asymptotically converges to the true Hessian $\frac{\partial^2 f(\mathbf{x})}{\partial\mathbf{x}^2}$. The approximated Hessians are used in equations (7) and (8) instead of the true ones. Usually, we choose $\mathbf{A}_0 = \mathbf{I}$ so that the first step of a quasi-Newton method coincides with a Steepest Descent step. For a NLS problem, we can also choose $\mathbf{A}_0 = \mathbf{J}^\top\mathbf{J}$ so that the first step of a quasi-Newton method coincides with a Gauss-Newton step. Well known algorithms used to build the sequence are the Davidon-Fletcher-Powell algorithm and the Broyden-Fletcher-Goldfarb-Shanno algorithm [6]. Both algorithms converge to the true Hessian in 6 steps if $f(\mathbf{x})$ is a quadratic form. It means we need at least 6 iterations of the algorithm before going in "one step" to the optimum (like the Newton minimization method). In fact, the algorithm is at most super-linearly convergent. The QNM control has been applied to visual servoing in [14].

IV. Efficient second-order control methods

The SDM, GNM and LMM methods have low convergence rates. The QNM method converges quadratically only asymptotically, while the NM method converges only if the Hessian is definite positive. The aim of this paper is to propose new control methods that achieve high convergence rates (at least quadratic) and avoid convergence problems of NM. Similarly to the GNM, I take into consideration the special structure of the NLS minimization. On the other hand, instead of the first-order Taylor series, the control laws proposed in this paper are based on the second-order Taylor series of $\mathbf{s}(\mathbf{x})$:

$$\Delta\mathbf{s} = -\mathbf{J}(\mathbf{x}_1)\Delta\mathbf{x} + \frac{1}{2}\mathbf{M}(\mathbf{x}_1,\Delta\mathbf{x})\Delta\mathbf{x} + O_{s2}(\Delta\mathbf{x}^3) \quad (15)$$

$$\Delta\mathbf{s} = -\mathbf{J}(\mathbf{x}_2)\Delta\mathbf{x} - \frac{1}{2}\mathbf{M}(\mathbf{x}_2,\Delta\mathbf{x})\Delta\mathbf{x} + O_{s1}(\Delta\mathbf{x}^3) \quad (16)$$

where O_{s1} and O_{s2} are the reminders, $\mathbf{M}(\mathbf{x}_1,\Delta\mathbf{x})$ and $\mathbf{M}(\mathbf{x}_2,\Delta\mathbf{x})$ are matrices containing all the n Hessian matrices of the (n×1) vector function $\mathbf{s}(\mathbf{x})$:

$$\mathbf{M}(\mathbf{x}_1,\Delta\mathbf{x}) = (\Delta\mathbf{x}^\top\mathbf{H}_1(\mathbf{x}_1), \Delta\mathbf{x}^\top\mathbf{H}_2(\mathbf{x}_1), \cdots, \Delta\mathbf{x}^\top\mathbf{H}_n(\mathbf{x}_1))$$

$$\mathbf{M}(\mathbf{x}_2,\Delta\mathbf{x}) = (\Delta\mathbf{x}^\top\mathbf{H}_1(\mathbf{x}_2), \Delta\mathbf{x}^\top\mathbf{H}_2(\mathbf{x}_2), \cdots, \Delta\mathbf{x}^\top\mathbf{H}_n(\mathbf{x}_2))$$

Starting from these equations it is possible to design two efficient second-order control methods that will be called respectively Mean of Jacobian Pseudo-inverses (MJP) and Pseudo-inverse of the mean of the Jacobians (PMJ).

A. Mean of Jacobian Pseudo-inverses

Multiplying both sides of equation (15) by $\mathbf{J}^+(\mathbf{x}_1)$ and both sides of equation (16) by $\mathbf{J}^+(\mathbf{x}_2)$ we obtain:

$$\Delta\mathbf{x} = -\mathbf{J}^+(\mathbf{x}_1)\Delta\mathbf{s} + \frac{1}{2}\mathbf{J}^+(\mathbf{x}_1)\mathbf{M}(\mathbf{x}_1,\Delta\mathbf{x})\Delta\mathbf{x} + O'_{s2}(\Delta\mathbf{x}^3) \quad (17)$$

$$\Delta\mathbf{x} = -\mathbf{J}^+(\mathbf{x}_2)\Delta\mathbf{s} - \frac{1}{2}\mathbf{J}^+(\mathbf{x}_2)\mathbf{M}(\mathbf{x}_2,\Delta\mathbf{x})\Delta\mathbf{x} + O'_{s1}(\Delta\mathbf{x}^3) \quad (18)$$

Let the matrix $\mathbf{J}^+(\mathbf{y})\mathbf{M}(\mathbf{y},\Delta\mathbf{x})$ be a function of $\mathbf{y}$. Consider its first-order Taylor series about $\mathbf{x}_1$, evaluated at $\mathbf{x}_2$:

$$\mathbf{J}^+(\mathbf{x}_2)\mathbf{M}(\mathbf{x}_2,\Delta\mathbf{x}) = \mathbf{J}^+(\mathbf{x}_1)\mathbf{M}(\mathbf{x}_1,\Delta\mathbf{x}) + O_{J^+}(\Delta\mathbf{x}^2) \quad (19)$$

where the reminder O_{J^+} is quadratic in $\Delta\mathbf{x}$. Computing the mean of equations (18) and (17), and plugging equation (19) into the mean, we obtain:

$$\Delta\mathbf{x} \approx -\frac{1}{2}(\mathbf{J}^+(\mathbf{x}_2) + \mathbf{J}^+(\mathbf{x}_1))\,\Delta\mathbf{s} + O_{MJP}(\Delta\mathbf{x}^3) \quad (20)$$

where:

$$O_{MJP}(\Delta\mathbf{x}^3) = O'_{s1}(\Delta\mathbf{x}^3) + O'_{s2}(\Delta\mathbf{x}^3) + O_{J^+}(\Delta\mathbf{x}^2)\Delta\mathbf{x}$$

is the total remainder which is cubic in $\Delta\mathbf{x}$. In conclusion, the mean of first-order approximations of the displacement, given in equations (11) and (12), is a second-order approximation of the displacement:

$$\Delta\mathbf{x} \approx -\lambda\frac{1}{2}(\mathbf{J}^+(\mathbf{x}_1) + \mathbf{J}^+(\mathbf{x}_2))\,\Delta\mathbf{s} \quad (21)$$

B. Pseudo-inverse of the mean of the Jacobians

We only consider problem P_1 since the same result can be obtained for P_2. Consider the second-order Taylor series of the Jacobian $\mathbf{J}(\mathbf{x})$ about $\mathbf{x}_2$ and evaluated at $\mathbf{x}_1$:

$$\mathbf{J}(\mathbf{x}_1) = \mathbf{J}(\mathbf{x}_2) + \mathbf{M}(\mathbf{x}_2, \Delta\mathbf{x}) + O_J(\Delta\mathbf{x}^2) \tag{22}$$

where O_J is the remainder. This formula provides an estimation to the second-order of matrix $\mathbf{M}(\mathbf{x}_2, \Delta\mathbf{x})$:

$$\mathbf{M}(\mathbf{x}_2, \Delta\mathbf{x}) = \mathbf{J}(\mathbf{x}_1) - \mathbf{J}(\mathbf{x}_2) - O_J(\Delta\mathbf{x}^2) \tag{23}$$

Plugging this equation into equation (18) we obtain:

$$\Delta\mathbf{s} = -\frac{1}{2}(\mathbf{J}(\mathbf{x}_1) + \mathbf{J}(\mathbf{x}_2))\Delta\mathbf{x} + O_{PMJ}(\Delta\mathbf{x}^3) \tag{24}$$

where $O_{PMJ}(\Delta\mathbf{x}^3) = O_{s2}(\Delta\mathbf{x}^3) - O_J(\Delta\mathbf{x}^2)\Delta\mathbf{x}$ is the total reminder which is cubic in $\Delta\mathbf{x}$. As a consequence, a second-order approximation of $\mathbf{s}(\mathbf{x})$ is again obtained using only first derivatives:

$$\Delta\mathbf{s} \approx -\frac{1}{2}(\mathbf{J}(\mathbf{x}_1) + \mathbf{J}(\mathbf{x}_2))\Delta\mathbf{x} \tag{25}$$

The displacement can be obtained by computing the pseudo-inverse of the mean of the Jacobians:

$$\Delta\mathbf{x} \approx -2\lambda\,(\mathbf{J}(\mathbf{x}_1) + \mathbf{J}(\mathbf{x}_2))^{+}\,\Delta\mathbf{s} \tag{26}$$

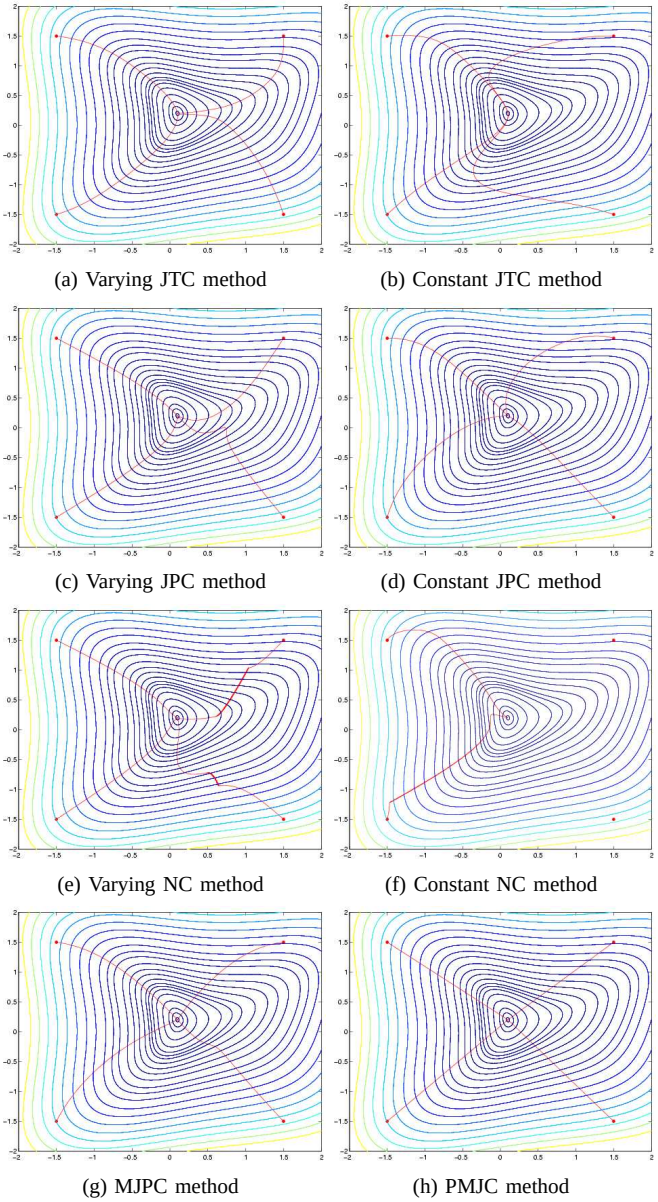
Note that for both second-oder methods the optimal minimization gain λ is 1. A similar control law with rough approximations of the Jacobians has been recently used in [15], but without any theoretical justification.

C. Differences between the two methods

Both methods compute a second-order approximation of the displacement $\Delta\mathbf{x}$. Therefore, despite they only use first derivatives, they have at least quadratic convergence. In conclusion, both methods will perform better than standard methods. However, there is a little difference between the two methods. When the vector $\mathbf{s}(\mathbf{x})$ is quadratic in $\mathbf{x}$ then $O_{s1} = O_{s2} = O_J = 0$. Thus, the reminder $O_{PMJ} = 0$ and equation (26) is exact. This means that the PMJ method has indeed a "quartic" convergence rate. On the other hand, even if $\mathbf{s}(\mathbf{x})$ is quadratic in $\mathbf{x}$ the reminder $O_{J^+} \neq 0$. When $\mathbf{s}(\mathbf{x})$ is not quadratic in $\mathbf{x}$ and $\Delta\mathbf{x}$ is big the approximations are not valid any more and the behavior of the two methods is unknown. Continuing research will give an insight into their differences.

V. Comparison between control methods

The convergence rate of minimization methods help us to understand the performance of control methods. Indeed, high convergence rates are achieved when the displacement is correctly estimated. For control methods, the direction of displacement is more important since the closed-loop compensates for errors on amplitude. The comparison between minimization/control methods is made here considering a simple example. Suppose a (4×1) vector function $\mathbf{s}(\mathbf{x})$ is quadratic in the (2×1) state vector $\mathbf{x} = (x, y)$. The NLS cost function is thus *quartic* in $\mathbf{x}$. Suppose the state to reach is $\mathbf{x}_1 = (0.1, 0.2)$. The simulation is repeated 4 times with different starting points: $\mathbf{x}_2 \in \{(\pm 1.5, \pm 1.5)\}$. Suppose we can measure $\mathbf{J}(\mathbf{x}_1)$ and $\mathbf{J}(\mathbf{x}_2)$ without knowing neither $\mathbf{x}_1$ nor $\mathbf{x}_2$. The results for 8 different minimization methods are given in Figure 1. The contours represent isolines (i.e. the cost function has the same value for each point of the contour) while the lines represent the paths for each starting point. Obviously, the ideal path (i.e. the shortest one) would be a straight line. Figure 1(a) shows that the varying JTC moves always in a direction perpendicular to the isolines (i.e. the steepest descent direction). For this reason, it has a slow convergence rate and can not reach the minimum following a straight line. The paths for the constant JTC method are even longer (see the length of the paths in Figure 1(b)) since the Jacobian is computed at the reference position. As expected, the constant JPC (Figure 1(d)) and the varying JPC (Figure 1(c)) performs better than the constant JTC and the varying JTC respectively. Ill conditioned and indefinite Hessian matrix cause the oscillations, observed in Figure 1(e), of the varying NC method.



(a) Varying JTC method (b) Constant JTC method

(c) Varying JPC method (d) Constant JPC method

(e) Varying NC method (f) Constant NC method

(g) MJPC method (h) PMJC method

Fig. 1. Comparing the behavior of 8 different control methods.

In two cases the constant NC (see Figure 1(f)) did not converge proving that the NC performs correctly only in a neighborhood of the solution. The MJPC method gives better results than previous methods (see the paths in Figure 1(g)). Finally, the PMJC method obviously gives the best solution since the paths in Figure 1(h) are straight lines from the starting points. Indeed, when the function is exactly quartic we can correctly estimate the displacement (in only one step) and thus the correct descent direction regardless on the shape of the isolines. Even if we do not take a full length stem (i.e. $\lambda < 1$), the trajectory in the state space will be optimal.

VI. Application to image-based visual servoing

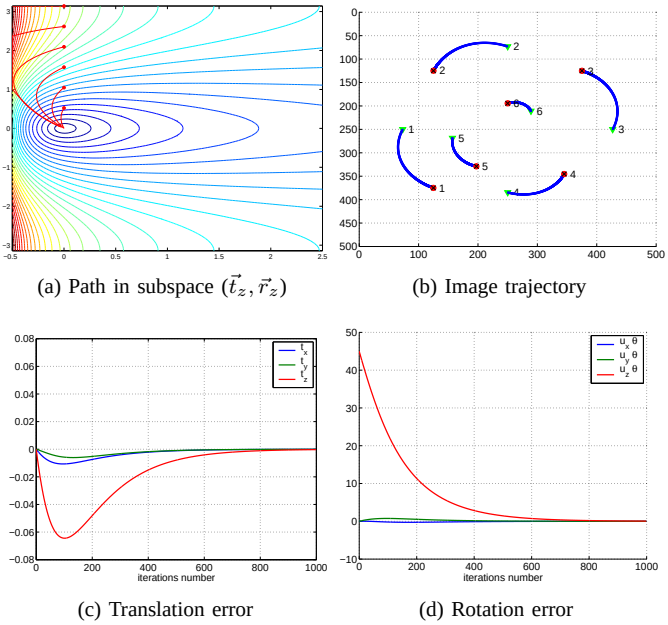
Consider the task of positioning an eye-in-hand camera with respect to a target composed of 6 feature points. In order to apply the new control scheme to visual servoing we must be able to compute the Jacobians at the reference and the current positions. Let the displacement of the camera be approximated by $\Delta\mathbf{x} \approx \mathbf{v}\,dt$, where $\mathbf{v}$ is the camera velocity. Thus, the Jacobian $\mathbf{J}(\mathbf{x}_1)$ coincides with the interaction matrix computed at the reference position and the Jacobian $\mathbf{J}(\mathbf{x}_2)$ coincides with the interaction matrix computed at the current position. Let us suppose that the depths of the points are correctly estimated. In order to show the improvement of the new control laws over standard control laws I consider here the very well known retreat/advance problem [4]. The retreat/advance problem of image-based visual servoing appears when coordinates of points are used in $\mathbf{s}$ and when the camera rotation is around the $\overrightarrow{z}$ axis. Standard control laws do not estimate accurately enough the direction of displacement of the camera inducing a backward motion (the retreat problem) or a forward motion (the advance problem). A degenerated case (known as the Chaumette Conundrum [5]) appears when the camera rotation reaches $\pm\pi$. The simulation is repeated for 6 different starting points $r_z \in \left\{\frac{\pi}{6}, \frac{\pi}{3}, \frac{\pi}{2}, \frac{2\pi}{3}, \frac{5\pi}{6}, \pi\right\}$, $t_z = 0 \;\forall\; r_z$.

A. Standard control laws

The results are obtained with the JPC methods. Similar retreat/advance behavior is obtained with other methods.

1) Constant Jacobian (advance problem):

In the first set of simulations, the Jacobian is constant and computed at the reference position. Figure 2(a) shows the the isolines of the cost function projected into the subspace ($\vec{t}_z$, $\vec{r}_z$) and the paths for each starting point. Since the initial movement is a pure rotation, the ideal path should be a straight line perpendicular to the $\vec{t}_z$ axis (i.e. $t_z = 0$). On the contrary, we observe that as r_z reach $\pm\pi$ the translational motion becomes bigger since the isolines become perpendicular to the $\vec{t}_z$ axis (i.e. the steepest descent direction is along $\vec{t}_z$). Figure 2 shows also the detailed results for a rotation of $\pi/4$. The camera moves towards the target (see the translation error initially increasing in Figure 2(c)) and it correctly rotates at the same time (see the rotation error decreasing in Figure 2(d)). Due to this undesired advancement some features can go out of the camera field of view when the camera comes too close to the target.



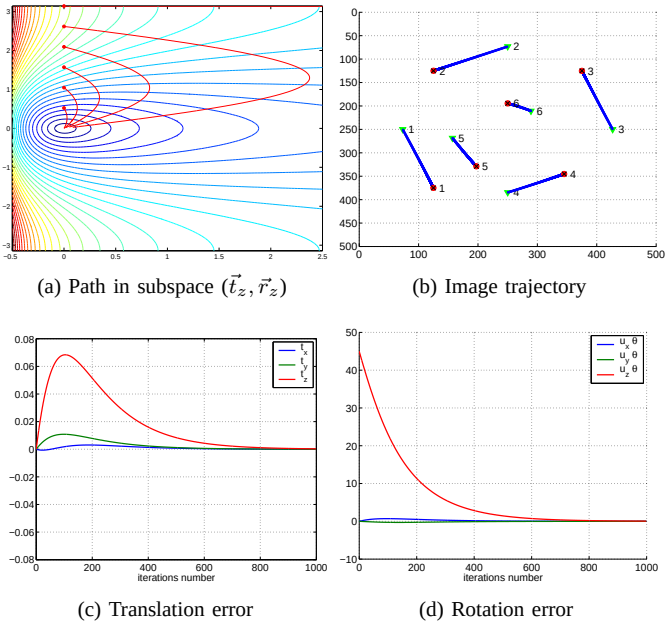
(a) Path in subspace ($\vec{t}_z, \vec{r}_z$) (b) Image trajectory

(c) Translation error (d) Rotation error

Fig. 2. The advance problem with the constant JPC. The paths for different initial rotations and the details for an initial rotation $r_z = \pi/4$.

2) Varying Jacobian (retreat problem):

In the second set of simulations, the Jacobian is updated at each iteration. The results of the simulations are plotted in Figure 3(a). As the initial rotation reaches π, the translation becomes bigger and bigger. Due to this undesired retreat the robot might reach the limit of its workspace. Figure 3(c) shows the translation induced when the initial rotation is $r_z = \pi/4$. The camera moves backwards and it correctly rotates at the same time (see the rotation error in Figure 3(d)). For both constant and varying JPC methods the approximation of the Hessian is not good enough to modify the direction of the steepest descent.



(a) Path in subspace ($\vec{t}_z, \vec{r}_z$) (b) Image trajectory

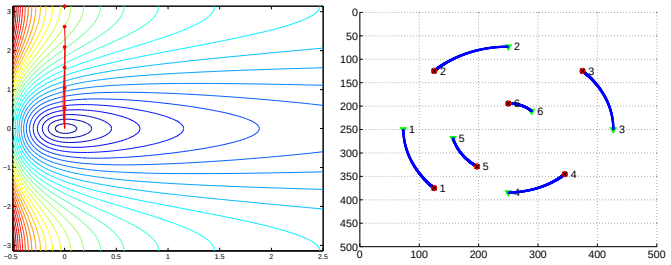
(c) Translation error (d) Rotation error

Fig. 3. The retreat problem with the varying JPC. The paths for different initial rotations and the details for an initial rotation $r_z = \pi/4$.

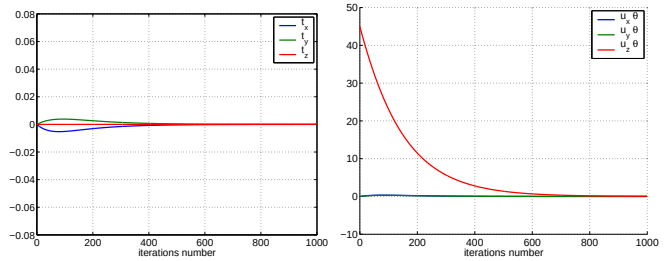
B. *Efficient second-order control laws*

For the proposed control laws the Chaumette Conundrum corresponds to an unstable local minimum for both methods. In that case, there are two symmetric possible solutions (turning clockwise or anti-clockwise) and the problem can be solved by choosing one of the solutions.

1) Mean of the Jacobian Pseudo-inverses: Using the MJPC method we obtain better results even for very large camera displacements. Except for $r_z = \pi$, Figure 4(a) shows that the paths are very close to straight lines parallel to the $\vec{r_z}$ axis. The direction of rotation is correctly estimated (see Figure 4(d)). Despite a small translation error still visible in Figure 4(c), the method gives better results than the previous ones.



(a) Path in subspace $(\vec{t_z}, \vec{r_z})$ (b) Image trajectory



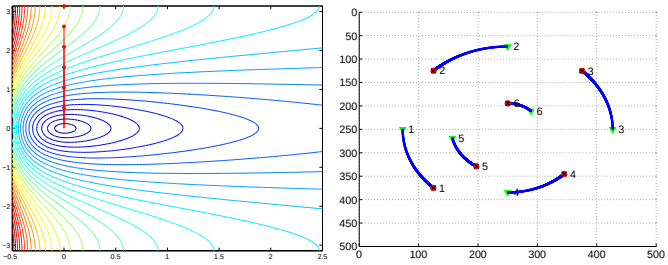
(c) Translation error (d) Rotation error

Fig. 4. Results using the MJPC method. The paths for different initial rotations and the details for an initial rotation $r_z = \pi/4$.

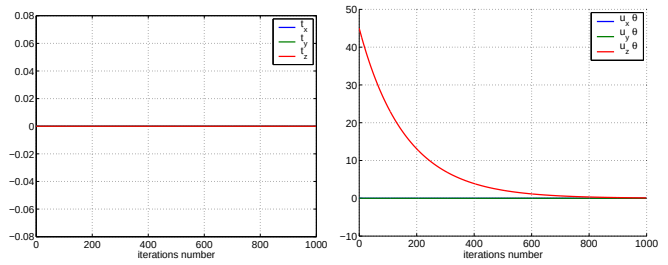
2) Pseudo-inverse of the mean of the Jacobians: Using the PMJC method, the camera motion is now a pure rotation. Indeed, the paths plotted in Figure 5(a) are straight lines proving that the PMJC method provide the best result since no additional motion is induced in the $\vec{t_z}$ direction (see Figure 5(c)). This is a simple example of the improvement that can been obtained over standard control laws. Indeed, even using a rough approximation of the Jacobians as in [15] a lager convergence domain and better 3D camera trajectory have been experimentally observed for generic camera displacements.

VII. Conclusion

In this paper, two new control laws with high convergence rates have been proposed. The control laws can be used for improving the performance of any vision-based robot control scheme. As an example, it has been shown that the new control methods allow to solve the well known camera advance/retreat problem of image-based visual servoing. Since they provide a better estimation of the camera displacement (to second order), the new control methods are expected to have a larger stability domain. Future work will be devoted to experiments on a real robot and to the stability analysis.



(a) Path in subspace $(\vec{t_z}, \vec{r_z})$ (b) Image trajectory



(c) Translation error (d) Rotation error

Fig. 5. Results using the PMJC method. The paths for different initial rotations and the details for an initial rotation $r_z = \pi/4$.

References

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

[2] S. Hutchinson, G. Hager, P. Corke, "A tutorial on visual servo control," *IEEE Trans. on Robotics and Automation*, vol. 12, no. 5, pp. 651-670, 1996.

[3] K. Miura, J. Gangloff, and M. Mathelin, "Robust and uncalibrated visual servoing without Jacobian using a simplex method," in *IEEE I.R.O.S.*, 2002, pp. 311–316.

[4] F. Chaumette, "Potential problems of stability and convergence in image-based and position-based visual servoing," in *The confluence of vision and control*, vol. 237 of *LNCIS Series*, pp. 66–78. Springer Verlag, 1998.

[5] P. Corke and S. Hutchinson, "A new partitioned approach to image-based visual servo control," *IEEE Trans. on Robotics and Automation*, vol. 14, no. 4, pp. 507–515, August 2001.

[6] J.E. Dennis and Schnabel R. B., *Numerical methods for unconstrained optimization and nonlinear equations*, Class. in app. math. SIAM, 1983.

[7] K. Hashimoto and H. Kimura, "LQ optimal and nonlinear approaches to visual servoing," in *Visual servoing*, vol. 7 of *World Scientific Series in Rob. and Aut. Syst.*, pp. 165–198. World Scientifi c Press, 1993.

[8] E. Malis, "Hybrid vision-based robot control robust to large calibration errors on both intrinsic and extrinsic camera parameters," in *European Control Conference*, 2001, pp. 2898–2903.

[9] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. on Robotics and Automation*, vol. 8, no. 3, pp. 313-326, July 1992.

[10] K. Levenberg, "A method for the solution of certain problems in least squares.," *Quart. Appl. Math.*, vol. 2, pp. 164–168, 1944.

[11] D.W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *SIAM J. of App. Math.*, vol. 11, pp. 431-441, 1963.

[12] C. W. Wampler, "Manipulator inverse kinematic solution based on vector formulations and damped least squares methods," *IEEE Trans. on System Man and Cybernetics*, vol. 16, no. 1, pp. 93–101, 1986.

[13] Y. Nakamura and H. Hanafusa, "Inverse kinematics solutions with singularity robustness for robot manipulator control," *Trans. ASME J. of Dynamic System, Measures and Control*, vol. 108, pp. 163–171, 1986.

[14] J. A. Piepmeier, G. V. McMurray, and H. Lipkin, "A dynamic quasi-newton method for uncalibrated visual servoing," in *IEEE Int. Conf. on Robotics and Automation*, May 1999, vol. 2, pp. 1595–1600.

[15] O. Tahri and F. Chaumette, "Application of moment invariants to visual servoing," in *IEEE Int. Conf. on Robotics and Automation*, 2003, vol. 3, pp. 4276-4281.