Nonlinear optimization methods for uncalibrated robot visual servoing

MIRJANA BONKOVIĆ
Department for Robotics and Intelligent Systems
University of Split
Ruđera Boškovića bb
CROATIA
mirjana.bonkovic@fesb.hr

Abstract: -This paper considers the uncalibrated, model free, robot visual servoing as a nonlinear optimization problem in which stochasticity is present due to a noise and there is no analytical form of the objective function. Such (robot) systems are suitable for unknown and unstructure envinronments due to minimal requirements related to calibration and robot kinematic's parameters. The numerical quasy-Newton methods offer a theoretical background for problem solving. However, additional attention has to be paid which assured stability and the robustness of the proposed method. In this paper we present the simulation results of various, well-known iterative numerical solution efficacy performing the dynamic target visual servoing. The results show that the generalized form of solutions perform better than the simple form adopted for a specific type of a goal functions.

Key-Words: - visual servoing, uncalibrated system, robot control, nonlinear optimization, generalized solutions

1 Introduction

Technical systems in general can be treated in a variety of ways. Nonlinear optimization procedures are one of the most general methods applicable on almost all type of real systems. In the case of visual servoing (VS), nonlinear optimization solutions make the minor part of all possible VS solutions. However, such type of control represents the most challenge one since it can be applied on the vision based system performing in an unstructured and unknown environment, where the real tasks usually have to be solved. Moreover, due to the characteristics of a VS system it is possible to visualize the interaction among the systems themselves: on one side, it is a technical system for vision based robot control, and, the opposite side, the system of well founded procedures for nonlinear optimization. Vision based system exposes different behavior under different nonlinear optimization procedures. At the same time, visual behavior of the technical system, proves the efficacy of the applied procedure or emphasize its drawbacks. The visual nature of a system can help to understand the method better – it simplifies the method behavior visualization. Consequently, the sense of mathematical parameters, constraints, defined preconditions and implications, get the physical meaning through the visual servoing performance which emerges the method as a paradigm for a specific problem solving.
In this paper, we present the VS problem as a typical nonlinear optimization problem and test the efficacy of a various numerical approaches in achieving the visual goals regarding the common conditions under which system performs such as noise, dynamic target tracking and static positioning.

The rest of paper consists as follows. In Section 2 the visual servoing problem in general is described brifly and the specific type of visual servoing problem which could be treated as nonlinear optimization problem, in more details. Section 3 described the appropriate algorithms which result with the specific visual servoing performances. In Section 4 we have compared the efficacy of the methods under the dynamic target tracking along the specific trajectory and static target positioning obtained through the simulations, whereas Section 5 concludes the paper.

2 The Visual Servo Control Problem Formulation

Visual servo control refers to the use of computer vision data to control the motion of the robot [1]. The main goal of the visual servoing is to move the robot tip (or mobile robot) to a certain pose with respect to the particular objects or features in images. Regarding the camera position, two types of visual servoing could be distinguished: fix-camera (or eye-on-hand) visual servoing in which vision data is acquired from stationary camera and nonstationary camera VS (eye in hand) during which the motion of the robot induces camera motion. Based on the error signal domain, two types of visual servoing system could be defined: image base visual servoing and position based visual servoing [2], [3]. The first one assumes that the error is defined in 3D (task space) coordinates, while IBVS is based on the error which is defined in terms of image features. The specification of an image-based visual servo task involves determining an appropriate error function e,

such that when the task is achieved, e=0 [3], where e is typically defined by

$$e(t) = s(m(t), a) - s^* \tag{1}$$

The parameter m in (1) is a set of image measurements, which is used to compute a set of k image features $s(m(t), a)$. Parameter a is a set of parameters that represent potential additional knowledge about the system. Vectos s^* contains the desired values of the features [1]. In this paper we are interested in robot visual control in a fixed camera configuration. Fig.1. shows the structure of the visual servo system used in this paper. Here, so called image-based visual servoing is considered, in which the error signal that is measured directly in the image (IBVS), is mapped to the robot actuators' command input.

The design of control scheme is based on velocity controller, in which the relationship between $\dot{s}$ (changes in visual appearance) and v (changes in control space) is given by

$$\dot{s} = L_S v \tag{2}$$

where, $L_s \in \Re^{k\times 6}$ is feature Jacobian.

Using (1) and (2) it is possible to obtain the realationship between robot velocity and the time variation of the error as:

$$\dot{e} = L_e v \tag{3}$$

considering $L_e = L_s$. The visual controller is constructed in order to determine the joint velocities v such as:

$$v = -\lambda L_e^{+} e \tag{4}$$

where L_e^{+}, λ , and e are the pseudoinverse of the Jacobian matrix L_e that relates joint coordinates with image features, control gain, and the error signal that is obtained by comparing the desired and current image feature parameters defined as (1), respectively.

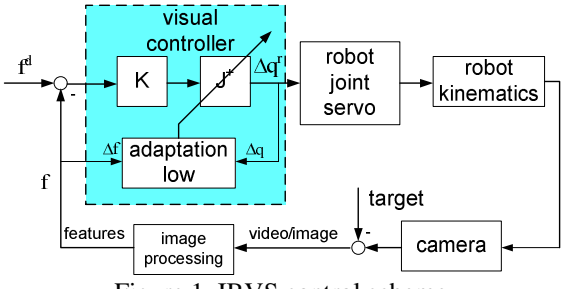

Figure 1. IBVS control scheme

In this paper we treat the visual servoing problem as a nonlinear optimization problem in which it could be formulated as a nonlinear least squares problem in which the goal function F is defined as:

$$F = \frac{1}{2} e(t)^T e(t) \tag{5}$$

where $e(t)$ is an appropriate error function expressed as (1).

One of the first and still today very efficient solutions was offered by Jägersand [5] in which he formulated the visual servoing problem as a nonlinear least squares problem solved by a quasi-Newton method using Broyden Jacobian estimation. Stability is ensured using thrust region method. The similar principles have also been applied for multiple camera model-based 3-D visual servoing [4].

If the target is moving, the system model has encountered the error not only as a function of robot poses but also of the pose of a moving objects. Consequently, [6] suggests use of dynamic quasi-Newton with recursive estimation scheme for Jacobian calculation. Recently, the generalized secant method has been proposed [7], [12], [13] based on the population of iterates which improves the servoing based on the acquired information from past iterates to calibrate at the best the model of a nonlinear function.

In this paper we present the efficasy of various numerical procedures through simulation results for uncalibrated vision-guided robotic control and compare the methods successfullness for the appropriate task solving.

3 Numerical Solutions for Visual Servoing Problem Solving

The visual servoing problem solution can be formulated as a resolutions of systems of nonlinear equations. We can characterize resolution of a nonlinear system of equations as finding $m \in \Re^n$ such that

$$F(m) = 0$$

where $F : \Re^n \to \Re^n$ is a continuously differentiable function.

Since Newton, this problem has received a tremendous amount of attention. Newton's method and its many variations are still intensively analyzed and used in practice. The philosophy of Newton like methods is to replace the nonlinear function F by a linear model, which approximates F in the neighbourhood of the current iterate. The original Newton method invokes

Taylor's theorem and uses the first derivative matrix (Jacobian) to construct the linear model. When the Jacobian is too expensive to evaluate, secant methods build the linear model based on the secant equation. Becouse secant methods exhibit a decent rate of convergence, they have been extensively analyzed in the literature. Each secant method derives a specific update formula which arbitrarily picks one linear model among them. The most common strategies are called "least – change updates" and select the linear model which minimizes the difference between two succesive models. Such methods are known as quasi-Newton methods [7]. Broyden [8] proposed the most successful class of quasi-Newton methods based on the secant equations, imposing the linear model M_{k+1} to exactly match the nonlinear function at iterates x_k and x_{k+1}, that is

$$\begin{aligned} M_{k+1}(x_k) &= F(x_k) \\ M_{k+1}(x_{k+1}) &= F(x_{k+1}) \end{aligned} \tag{6}$$

Subtracting these two equations and defining

$$\begin{aligned} y_k &= F(x_{k+1}) - F(x_k) \text{ and} \\ s_k &= x_{k+1} - x_k \end{aligned} \tag{7}$$

we obtain the classical secant equation:

$$L_{k+1} s_k = y_k \tag{8}$$

If the dimension *n* is strictly greater than 1, there are an infinite number of matrices L_{k+1} satisfying (8). The "least-change secant update", proposed by Broyden, could be desbribed as:

1 minimize $\|L_{k+1} - L_k\|$ (9)

2 use constraint expressed with (8)

which result with the following update formula

$$L_{k+1} = L_k + \lambda s_k^T \tag{10}$$

This method has been proved as successful for visual servoing if an additional technique has been used to improve the robustness and stability of the method. In the rest of this section the various methods have been described which appropriately supplement the quazy-Newton base for nonlinear system solving.

3.1 Broyden visual servoing model

This method has been similar to those described in the introductory part of this section. Pure Broyden has been sensitive to noise, so numerous authors suggest use of of modified Broyden, improved with the factor η [1], [14] which defines the update speed. Having this fact in mind, the Jacobian update has been calculated with Broyden estimation according to (12):

$$J_{k+1} = J_k + \eta \frac{(y_k - J_k s_k) s_k^T}{s_k^T s_k} \tag{12}$$

The influence of changing η has been the subject of our earlier works [12], while in this paper we reference the method for the efficacy comparison reasons.

3.2 Dynamic visual servoing model

Piepmeier suggests [6] using of dynamic quasi-Newton with recursive estimation scheme for Jacobian calculation. A new, dynamic visual servoing model has been proposed in which the qualifier "dynamic" refers to the presence of the error velocity term ($\frac{\partial f(x_{k+1})}{\partial t}$). Desired recursive estimation scheme that minimizes a cost function based on the change in the affine model, result with the Jacobian update equation:

$$\begin{aligned} J_{k+1} &= J_k + \frac{\left(y_k - J_k s_k - \frac{\partial f(x_{k+1})}{\partial t} s_t \right) s_k^T}{\lambda + s_k^T P_k s_k}, \\ P_{k+1} &= \frac{1}{\lambda}\left(P_k - \frac{P_k s_k s_k^T P_k}{\lambda + s_k^T P_k s_k} \right) \end{aligned} \tag{13}$$

where $y_k = f(x_{k+1}) - f(x_k)$, $s_k = x_{k+1} - x_k$, $s_t = t_{k+1} - t_k$ and $0 < \lambda \leq 1$ is a weighting parameter.

3.3 Population based uncalibrated visual servoing

Population-based generalization prefer to identify the linear model which is as close as possible to the nonlinear function in the least-squares sense. At each iteration, the finite population of iterates $x_0 .... x_{k+1}$ are maintained. The method also belongs to quasi-Newton framework, where B_{k+1} is computed as

$$L_{k+1} = \arg\min_J \left(\sum_{i=0}^{k} \left\| \omega_{k+1}^i F(x_i) - \omega_{k+1}^i M_{k+1}(x_i, J) \right\|_2^2 + \left\| \Gamma J - \Gamma L_{k+1}^0 \right\|_F^2 \right) \tag{14}$$

where L_{k+1} is defined by (3) and the $L_{k+1}^0 \in \Re^{n \times n}$ is an a priori approximation of L_{k+1}. The role of the second term is to overcome the under-determination of the least-square problem based on the first term and also to control the numerical stability of the method. The matrix contains weights associated with the arbitrary term L_{k+1}^0, and the weights $\omega_{k+1}^i \in \Re^+$ are associated with the previous iterates. Equation (14) can be rewritten in matrix form as follows:

$$J_{k+1} = \arg\min_{J} \left\| \left(J(S_{k+1} I_{n\times m}) \begin{pmatrix} \Omega & 0_{k\times n} \\ 0_{k\times k} & \Gamma \end{pmatrix} - \begin{pmatrix} Y_{k+1} & B^0_{k+1} \end{pmatrix} \begin{pmatrix} \Omega & 0 \\ 0 & \Gamma \end{pmatrix} \right) \right\|_F^2 \tag{15}$$

where $\Omega \in \Re^{k\times 1}$ is a diagonal matrix with weights ω^i_{k+1} on the diagonal for i=0,…,k. The normal equations of the least-squere problem lead to the following formula:

$$J_{k+1} = J^0_{k+1} + (Y_{k+1} - J^0_{k+1} S_{k+1}) \Omega^2 S^T_{k+1} (\Gamma^2 + S_{k+1} \Omega^2 S^T_{k+1})^{-1} \tag{16}$$

$$Y_{k+1} = (y_k, y_{k-1}, ..., y_0)$$

and

$$s_{k+1} = (s_k, s_{k-1}, ..., s_0).$$

The role of the a priori matrix J^0_{k+1} is to overcome the possible under-determination of problem (16). We have chosen $J^0_{k+1} = J_{k+1}$, which exhibits good properties, so (16) becomes an update formula, which local convergence has been proved in [7]. The weights ω^i_{k+1}, capture the relative importance of each iterate in the population, and the matrix Γ captures the importance of the arbitrary terms defined by J^0_{k+1} for the identification of the linear model. The weights have to be finite and Γ must be such that $\Gamma^2 + S_{k+1}\Omega^2 S^T_{k+1}$ is safely positive definite. To ensure this property we seek for a technique to guarantee both the problem of overcoming the under-determination, and numerical stability. Such problem can be solved in a variety of ways [7], [11], but we have found out that the most appropriate is to define it trough simulations as a small positive constant which guaranties positive definition of the term $\Gamma^2 + S_{k+1}\Omega^2 S^T_{k+1}$.

4 Simulations

4.1 The system

The simulated system is presented in Fig.1. It consists of components which characteristics are transformed from real experimental setup [12],[13]. During simulations the task has been performed using 2DOF planar parallel manipulator with four revolute joints and a fixed camera that can provide position information of the robot tip and the target in the robot workplace. The robot direct kinematics is given by the following equations,

$$x = L \begin{bmatrix} \cos(q_1) + \cos(q_2) \\ \sin(q_1) + \sin(q_2) \end{bmatrix} \tag{17}$$

where q_1, q_2 are the robot joint angles, and x is a vector of robot tip coordinates in the Cartesian world coordinate frame (Fig.2). L=0.4m is the length of the robot single link. Translation and rotation of the camera frame with respect to the robot world base frame is given by the RPY homogenous transformation matrix Rc (41). It is rotated around y-axis for 135°, and translated for 1.2, and 1.2 m in y and z direction respectively.

$$R_C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -0.707 & -0.707 & 1.2 \\ 0 & 0.707 & -0.707 & 1.2 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{18}$$

A block named “robot servo” in Fig. 1. represents the robot system dynamics which includes motor, current and velocity-loop dynamics for joints. It has been modeled with the first order open loop transfer function as:

$$G(s)=100/(s+100), \tag{19}$$

which means that the velocity-loop is very fast with respect to the sampling interval (T_{camera}). The input velocity error has been saturated according to robot specification with limit=0.5. Visual feedback gain has been set to K=5. The “robot servo” itself represents an open loop system, due the direct feedback from joints has been used as input in visual servo controller (Fig.1) for Jacobian update B_{k+1} calculation.

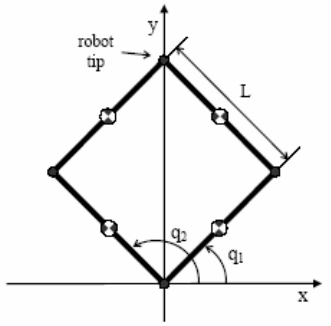

Fig.2. Planar 2DOF robot manipulator

4.2 Simulation results

In this paper, the image processing node generates the target point applied in the visual task definition within the image. When the robot tip reached the target, the target point was moved to another position in order to provide traveling of the robot tip through the whole robot work plane. The position of the target point determined corners of an "envelope" shape trajectory in the image plane.The projection of the target positions on the robot workplane is depicted by Fig.3a. The initial robot tip position is marked with “0” and the

corresponding robot joint angles have the following values: $q_1 = 15^\circ, q_2 = 165^\circ$. The initial target position is marked with "0", and the referent positions are marked "1", "2", "3" and "4". The target point positions were generated in the following order: "0"-"1"-"2"-"0"-"4"-"1"-"2"-"3"-"0". The control algorithm has been implemented in SIMULINK model using appropriate S function. For reference trajectory, marked with points "0"-"1"-"2"-"3"-"4" in Fig. 3., the rectangle has been chosen with the upper left corner (x0-X,y0-Y) and the down right corner (x0+X,y0), expressed in the image coordinates, where (x0,y0) is the target start point defined with robot angles $q_1 = 15^\circ, q_2 = 165^\circ$, X=125 pixels and Y=190 pixels respectively. A target start position has been the same as the robot tip start position and it has been moved during simulations with constant speed (measured in pixel/s). Camera refresh rate has been *T_camera=0.033 s.* Along the curves "1"-"2" and "3"-"4" the y component of the speed has been set to zero. The robot tip starts from the point where target is positioned and marked in Figure 2. as "0". It is worth to notice that all simulations have been performed under the geometrical noise, which is generated through truncation of image pixels value of the robot tip position, which is a normal procedure in IBVS. We have performed our simulations with target speed $v = 2/T_{cam}$ to obtain the dynamic visual servoing results in which the target has "the minimally changed" positions along the path for "ordinary" Broyden method with a constant Jacobian update speed $\eta = 0.2$, dynamic visual servoing and population based visual servoing. The results are presented in Fig.3. The first column of Fig.3a presents the reference (red line) and real robot end effector trajectory (blue line), whereas the second column shows the robot joints' speed and u, v feature tracking. It can be noticed that the Broyden method performs poorly (Fig.3ab), with big deviations from the reference trajectory (especially on the segment marked as "2"-"0"). The Broyden emerges nonstability in the end point, which can be clearly seen in Fig.3b., as oscillations of the robot joints at the time instances after 41 s. Contrary, the dynamic VS method as well as population method expose the stable behavior (Fig.3cd. and Fig.3ef.). Althoug the both methods result with the trajectories which tightly follows the referent one, the population method could be characterized as "the better one" due to shape of joint velocities curve (Fig.3.d.) which are smoother that the joint velocities curve belonging to dynamic visual servoing method (Fig.3f.). The same considerations have been confirmed even for static target positioning and for varying speed target tracking.

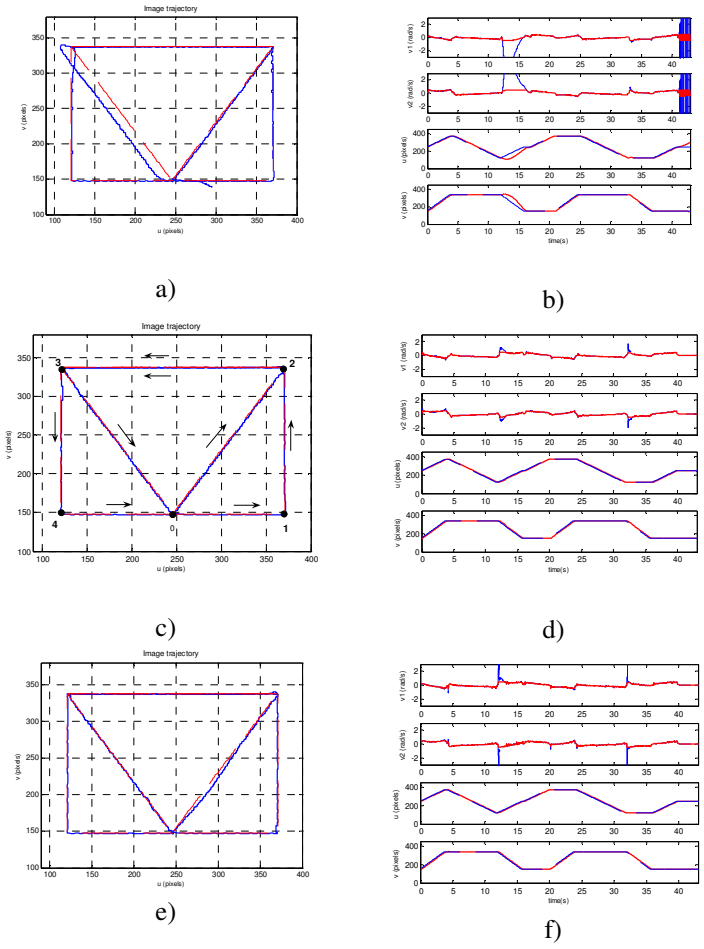

Figure 3. Simulation results: image contour (left column: reference trajectory-red line, achieved trajectory-blue line), joint velocities and u, v feature tracking (right column) for:
a), b) Population method
c), d) Broyden method
e), f) Dynamic visual servoing method

5 Conclusions

The uncalibrated image based visual servoing represents the challenge in the visual controller design due to numerous unknowns present in the system. Such systems have been usefull in an unstructured envinronment which we usually have in the real world. In this paper we have presented the various numerical approaches for problem solving which offers the "Broyden-like" update for the Jacobian. Although all methods have been declared as quasy-Newton based, they are not equaly succesfull in achieving the desired goals. Simulation shows that additional attention have to be paid in overwhelming the unwanted system characteristics. Generalized form of solutions performs better than the simple form adopted for a specific type of a goal functions. Dynamic visual servoing which represents

generalized form to the case of moving objents performs better than Broyden method with adopted Jacobian update speed, whereas population based visual servoing, which represents the generalized form of a solution which includes the history of iterates and calibration of the model during servoing too, performs the best.

6 Acknowledgement

This work was supported by the Ministry of Science and Technology of the Republic Croatia under project: AgISEco – Agent based systems for envinronment monitoring and protection (023-0232005-2003).

References:

[1] F. Chaumette, S.Hutchinson, "Visual Servo Control Part I & II", IEEE Robotics and Automation Magazine, December 2006.

[2] S. Hutchinson, G.D. Hager, P. Corke, "A Tutorial on Visual Servo Control", IEEE Trans. On Robotics and Automation, Vol. 12, No. 5, Oct 1996.

[3] P. I. Corke, "Visual control of robots, high performance visual servoing", John Wiley & Sons Inc., 1996.

[4] J. Stavitzky, D.Capson, "Multiple Camera Model Based Visual Servo", IEEE Trans. On Robotics and Automation, pp. 732-739, Vol. 16, No.6, Dec.2000.

[5] M. Jagersand, R. Nelson, "On-line Estimation of Visual-Motor Models using Active Vision", Proc. ARPA Image Understanding Workshop 1996.

[6] J. A. Piepmeier, G. V. McMurray, H. Lipkin "Uncalibrated Dynamic Visual Servoing", IEEE Trans. On Robotics and Automation, Vol. 20, No.1, pp. 143-147, February 2004.

[7] F. Crittin, M.Bierlaire, "A generalization of secant methods for solving nonlinear systems of equations", in Proc. 3rd Swiss Transport Research Conference, march 19-21, 2003.

[8] C. G. Broyden, "A class of methods for solving nonlinear simultaneous equations", Mathematics of Computation 19, pp. 577-593, 1965.

[9] P. Wolfe, "The secant method for solving nonlinear equations", Communications ACM, Vol.12: pp.12-13, 1959.

[10] Ortega, Rheinboldt, "Iterative solution of nonlinear equations in several variables", Academic Press, New York, 1970.

[11] M. Bierlaire, F. Crittin, "Solving Noisy, Large Scale Fixed-Point Problems and Systems of Nonlinear Equations", Transportation Science, Vol. 40, No. 1, February 2006, pp.44-63

[12] Bonković M., Hace A., Jezernik K., "A new method for uncalibrated visual servoing", Proc. of AMC, Istanbul, Turkey, pp.624-629., 2006.

[13] Bonković M., Hace A., Jezernik K., "Population based visual servoing", IEEE/ASME Tran. On Mechatronics, Vol.13, June 2008.

[14] Dodds G., Zatari A.* and Bischof R., "Uncalibrated Visual Servoing for Full Motion Dextrous Robot Systems with Tracking Cameras" (http://robotik.w3.rz.unibw-muenchen.de/PDF/UnViSer.pdf)