Visual servoing with linearized observer

Koichi Hashimoto
Okayama University

Toshiro Noritsugu
Okayama University

Proceedings of the 1999 IEEE
International Conference on Robotics & Automation
Detroit, Michigan • May 1999

Visual Servoing with Linearized Observer

Koichi Hashimoto and Toshiro Noritsugu
Department of Systems Engineering,
Okayama University
koichi@sys.okayama-u.ac.jp

Abstract

One of the most important problem in feature-based visual servoing is the slow sampling rate and the delay of the camera that can make the closed loop system oscillative or unstable easily. In this paper, a linearized observer that estimates the object velocity and updates the visual information with the joint sampling rate is proposed. Stability of the observer-based control system and effectiveness of the observer are verified by experiments on a PUMA 560 robot.

1 Introduction

Vision sensor provides rich information on the object and environment, however the sampling rate is usually very slow (e.g., 30Hz) compared with that of mechanical systems (e.g., 1000Hz). Thus, it is natural to compose the control system with two feedback loops having different sampling rates. A block diagram of a feature-based visual servo system is shown in Figure 1. Visual sensor is incorporated in the feedback loop and joint servo loop lies inside the vision loop. Since the inner loop is 30 times faster than the outer loop, the reference command v for the joint servo system should be interpolated so as to avoid saturation of the inner loop driver due to large step change of the reference.

Not only interpolating the sensor readings, tracking performance will be improved if the object motion dynamics is predicted. Since the CCD camera and image processing hardware has sampling/computation delay, feedforward type controller is much more effective than pure feedback for achieving high performance visual servoing. There are many publications that try to incorporate feedforward structure into the visual servo controller. Most of them assume zero-mean Gaussian object acceleration, which is required to apply Kalman filter, α-β-γ filter or AR model [9, 1, 6]. However this assumption is not proper for many tracking applications.

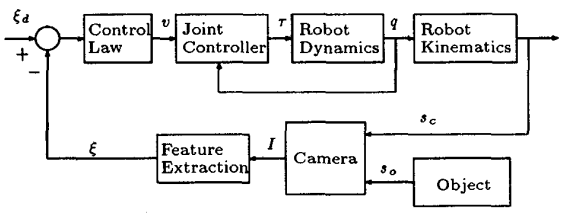

Figure 1: Feature-based Visual Tracking

On the other hand, if a model for object motion is available, a Luenberger type observer can be adopted. Ghosh et al. [2] propose observers for estimating the target velocity . Rizzi and Koditchek [7] study a window position predictor for object tracking. The authors use an observer for estimating the object velocity and propose a nonlinear controller with the observer [4]. On the basis of the object motion model, the observer is formulated as a nonlinear adaptive identification problem and unknown parameters such as position, direction, velocity, center of circle and so on are estimated. Thus it can be used for a large class of motions including constant velocity, constant acceleration and cyclic motions.

This paper considers an compensation scheme for the vision sampling delay by estimating the target velocity. A model describing the object motion is introduced and a nonlinear observer is presented. The effectiveness of the observer-based method was evaluated by simulations and experiments on a two link planar direct drive robot [4]. Since the computational burden of the controller was too large it was not suitable for robots that have more than three degrees-of-freedom. In this paper, a linearized version of the observer is presented by neglecting the nonlinear dynamics of the robot. Experiments with PUMA 560 are carried out to show the validity of the linearized feedforward type controller.

0-7803-5180-0-5/99 $10.00 © 1999 IEEE

2 Mathematical Formulation

Robot Model Assume that the robot has m (≤ 6) joints and the camera is mounted on the robot hand. Let q be the ($m \times 1$) joint angle vector and s_c be the (6×1) vector of camera position and orientation, then the **kinematic model** is given by $s_c = \phi(q)$.

Object Motion Model Suppose that the object has m_o (≤ 6) degrees of freedom and its generalized coordinates is p ($m_o \times 1$). Then the object position and orientation is given by $s_o = \psi(p)$ (6×1). Assume that the object velocity is generated by ℓ ($\leq m_o$) dimensional parameter vector θ^* such that

$$\dot{p} = W(p)\theta^* \tag{1}$$

is satisfied, where $W(p)$ is an $m_o \times \ell$ matrix function of p. The vector θ^* and the equation (1) are called the **velocity parameter** and the **object motion model**, respectively. This motion model is simple but it can model fairly large class of autonomous motions including straight, circular, oval and "figure 8" motions. Similar model was studied and extended to the two stage estimator in [3].

Camera Model The object image is generated by the perspective projection of the relative position between the camera and the object. Let ${}^cR_w$ be the (6×6) coordinate transformation matrix from the world coordinates to the camera coordinates. Then the relative position of the object r is defined by

$$r = [X\ Y\ Z\ \alpha\ \beta\ \gamma]^T = {}^cR_w(s_o - s_c), \tag{2}$$

where X, Y, Z represent the object position in the camera coordinate system and α, β, γ are the object orientation parameters of any kind. Let the vector $[x\ y]^T$ be the coordinates of a feature point in the image plane. Then, the camera model is defined by

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{f}{Z}\begin{bmatrix} X \\ Y \end{bmatrix}, \tag{3}$$

where f is the focal length of the lens. If there are n feature points on an object, then the object image can be represented by a $2n$ dimensional feature vector $\xi = [\xi_1^T \ \cdots \ \xi_n^T]^T$, which is a function of the relative position between the object and the camera r. This relation is called the **camera model** and is expressed by the mapping ι defined by

$$\xi = \iota(r), \quad \xi_d = \iota(r_d), \tag{4}$$

where r_d is the desired relative position between the camera and the object and ξ_d is the desired feature vector.

Jacobians Differentiation of the camera model (4) yields

$$\dot{\xi} = J\dot{q} + L\dot{p}, \quad J = \frac{\partial \iota}{\partial r}\frac{\partial r}{\partial q}, \quad L = \frac{\partial \iota}{\partial r}\frac{\partial r}{\partial p}. \tag{5}$$

The matrices J ($2n \times m$) and L ($2n \times m_o$) are called **image Jacobian** and **motion Jacobian**, respectively. It is straight forward to see that

$$J_f = \frac{\partial \iota}{\partial r} = \begin{bmatrix} J_f^{(1)} \\ \vdots \\ J_f^{(n)} \end{bmatrix} \tag{6}$$

where

$$J_f^{(i)} = \begin{bmatrix} -\frac{f}{Z_i} & 0 & \frac{x_i}{Z_i} & \frac{x_i y_i}{f} & -\frac{x_i^2+f^2}{f} & y_i \\ 0 & -\frac{f}{Z_i} & \frac{y_i}{Z_i} & \frac{y_i^2+f^2}{f} & -\frac{x_i y_i}{f} & -x_i \end{bmatrix}. \tag{7}$$

Then we have

$$J = J_f\,{}^cR_w\frac{\partial \phi}{\partial q}, \qquad L = -J_f\,{}^cR_w\frac{\partial \psi}{\partial p}. \tag{8}$$

Consider a feature point that lies on the optical axis of the camera. When the point moves along the optical axis or when the camera rotates around the optical axis, the object image do not change. Thus this point feature is not useful to control the camera position in the Z axis and rotation around the Z axis. In general, if the Jacobians are not full rank, there exist such direction(s) of motion that do not change the image features and the closed loop system becomes internally unstable [5]. Thus, we assume rank$J = m$ and rank$L = m_o$. To satisfy the condition on J, $2n \geq m$ is necessary. Also if four feature points on a object plane is available, then the condition is satisfied for practically all object/camera configuration. The relation between the feature/camera configuration and visual servo performance are studied in [5].

3 Nonlinear Control Law

Controlled Variable Let the nominal point be r_d that satisfies $\xi_d = \iota(r_d)$. Define a matrix $B = J(q^*, p^*)$ as the image Jacobian at the nominal position of the object p^* and the nominal configuration of the robot q^* that satisfy $s_c(q^*) - s_o(p^*) = r_d$. Then the controlled variable z is defined by

$$z = B^T(\xi - \xi_d). \tag{9}$$

The objective $\xi \to \xi_d$ is guaranteed when $z \to 0$ provided that $B^T J$ is positive definite [4].

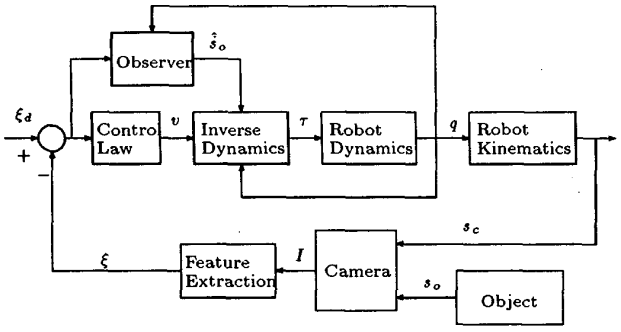

Figure 2: Observer-based Controller

Controller A strictly linearizing controller is derived by calculating the second derivative of z:

$$\ddot{z} = B^T J M^{-1}(\tau - h) + \lambda + N\theta^* + \Phi\kappa(\theta^*). \quad (10)$$

Please refer to [4] for the definition of λ, N, Φ, κ. Since $B^T J$ is invertible for p and q in the neighborhood of p^* and q^*, the actuator torque

$$\tau = M(B^T J)^{-1}(v - \lambda - N\theta^* - \Phi\kappa(\theta^*)) + h \quad (11)$$

with new input $v = -K_1 z - K_2 \dot{z}$ linearizes the dynamics to yield $\ddot{z} + K_2\dot{z} + K_1 z = 0$, which is exponentially stable.

Observer The control law (11) requires unknown variables $\dot{z}$ and θ^*. Thus an observer is used to estimate them. Let the estimates of the parameter θ^* and controlled variable z be $\hat{\theta}$ and $\hat{z}$, respectively, and consider the following estimator [8]

$$\begin{aligned} \dot{\hat{z}} &= B^T J\dot{q} + B^T LW\hat{\theta} + H(\hat{z} - z) \\ \dot{\hat{\theta}} &= -W^T L^T BP(\hat{z} - z), \end{aligned} \quad (12)$$

where H is any stable matrix and Q is any positive definite matrix. While P is selected to satisfy

$$H^T P + PH = -Q, \quad Q > 0. \quad (13)$$

Let the estimation error be $e = [(z - \hat{z})^T \ (\theta^* - \hat{\theta})^T]^T$, then the estimator (12) makes the equilibrium point $e = 0$ asymptotically stable.

Observer-based Controller On the basis of the estimated velocity of the feature vector $\hat{z}$ and the estimated object motion parameter $\hat{\theta}$, consider the following controller:

$$\begin{aligned} \tau &= M(B^T J)^{-1}(v - \lambda - N\hat{\theta} - \Phi\kappa(\hat{\theta})) + h, \\ v &= -K_1 z - K_2 B^T(J\dot{q} + LW\hat{\theta}). \end{aligned} \quad (14)$$

Defining $\bar{\kappa} = \kappa(\theta^*) - \kappa(\hat{\theta})$ and substituting (14) into (10) yields

$$\ddot{z} = -K_1 z - K_2(\dot{z} - B^T LW\bar{\theta}) + N\bar{\theta} + \Phi\bar{\kappa}. \quad (15)$$

Thus we obtain the following closed loop dynamics

$$\dot{x} = \bar{A}x + \bar{N}\bar{\theta} + \bar{\Phi}\bar{\kappa}, \quad (16)$$

where $x, \bar{A}, \bar{N}$ and $\bar{\Phi}$ are defined by

$$\begin{aligned} x &= \begin{bmatrix} z \\ \dot{z} \end{bmatrix}, \ \bar{A} = \begin{bmatrix} 0 & I \\ -K_1 & -K_2 \end{bmatrix}, \\ \bar{N} &= \begin{bmatrix} 0 \\ N + K_2 B^T LW \end{bmatrix}, \ \bar{\Phi} = \begin{bmatrix} 0 \\ \Phi \end{bmatrix}. \end{aligned} \quad (17)$$

Then, the estimator (12) and the controller (14) make the equilibrium point $(x, e) = 0$ asymptotically stable [4]. The block diagram in Figure 2 shows the structure of the object motion estimation and robot motion control that are carried out in parallel.

4 Linearized Controller

The controller derived in the previous section is rigorous and useful for theoretical analysis. However, the computational burden of the inverse dynamics becomes huge for robots that have many degrees of freedom ($M, J, L, W, \lambda, N, \kappa$ are all state dependent). On the other hand, most general purpose industrial robots are velocity control type and the joints contain high ratio gears, thus the terms of acceleration, centrifugal force and Colioris force can be neglected. Then the controller can be linearized without having serious performance deterioration and the computational burden can be reduced considerably.

The linearized dynamics of the visual servo system (9) at $(q, p) = (q^*, p^*)$ is given by

$$\dot{z} = B^T B\dot{q} + B^T L^* W\theta^* \quad (18)$$

where $L^* = L(q^*, p^*)$. Then the following control law

$$\dot{q} = (B^T B)^{-1}(v - B^T L^* W\theta^*) \quad (19)$$

linearizes the dynamics from v to z as $\dot{z} = v$. Thus with a positive definite gain matrix K, the feedback law $v = -Kz$ makes the equilibrium point $z = 0$ exponentially stable.

For the linearized system (18), linearized observer similar to (12) can be used:

$$\begin{aligned} \dot{\hat{z}} &= B^T B\dot{q} + B^T L^* W\hat{\theta} + H(\hat{z} - z), \\ \dot{\hat{\theta}} &= -W^T L^{*T} BP(\hat{z} - z). \end{aligned} \quad (20)$$

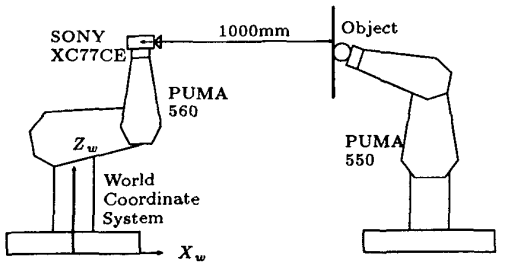

Figure 3: Robot Configuration and Object Position

Note that, at the rate of joint servo, the observer can be drived and the image feature vector z can be updated. Thus the large vision delay is canceled by using $\hat{z}$ instead of z. Since B and L^* are constant matrices, the computational burden is fairly reduced. Only W may depend on the object position p. Some examples of the linearized controller with the PUMA robot will be described in the following section.

5 Experiments

5.1 Linear Motion

Experimental Setup A board with five feature points is attached to a PUMA 550. The X_w-Y_w-Z_w world coordinate system is at the base of the PUMA 560 which holds the camera. A nominal camera position is in front of the the marks and the distance is about 1000 [mm]. The nominal positions of the object and the camera are shown in Figure 3.

Object Motion In this experiment, the object moves up and down; that is, the object motion is translational in the Z_w direction. Thus, $\dot{s}_o = [0\ 0\ v_z^*\ 0\ 0\ 0]^T$, where v_z^* is the object velocity in the vertical direction Z_w. Since the object motion is one dimensional, we can choose the generalized coordinate as the object height in the world coordinate system, i.e., $p = Z_o$. Then we have $\dot{p} = v_z^*$ and the parameterization (1) is given by $W = 1$ and $\theta = v_z$. Since $\partial s_o/\partial p = [0\ 0\ 1\ 0\ 0\ 0]^T$, the matrix L for the observer becomes as follows:

$$L = J_f {}^cR_w [0\ \ 0\ \ 1\ \ 0\ \ 0\ \ 0]^T \qquad (21)$$

The observer estimates v_z^*, that is, the object velocity in the vertical direction. At $t = 10$ [sec] the object starts to move with velocity -20 [mm/sec], that is, 20 [mm/sec] in the downward ($-Z_w$ direction), and stops at $t = 15$; after 10 seconds of pause, it moves upward with velocity 10 [mm/sec] and stops at $t = 35$.

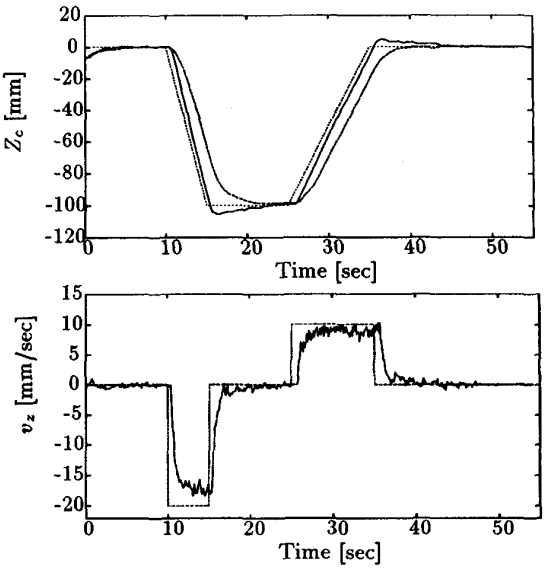

Figure 4: Step Response
(a) —: with Observer, - -: without Observer, · · ·: Reference (b) —: Estimated, - -: True

Experiment The experimental results are shown in Figures 4 (a) and (b). In Figure 4 (a), the vertical axis shows the position of the camera in the world coordinate system. Only the data in Z_w direction is shown but note that all 6 DOF are controlled by visual servo. The performance difference due to observer is found only in Z_w direction. The solid and broken lines are the results with and without the observer, respectively. The dotted line is the reference trajectory of the camera. The control law with the observer is given by (20) and (??), and the control law without observer is

$$\dot{q} = -(B^T B)^{-1} K z. \qquad (22)$$

The matrix K is the gain, which is the same as the observer-based controller. By using the observer the tracking speed is improved and the tracking error is reduced considerably. As shown in the nonlinear case [4], overshoot is a shortcoming of the observer-based scheme. Figure 4 (b) shows the estimated velocity of the object. The broken line shows the true value and the solid line is the estimated value. The observer estimates the object velocity fairly accurately.

5.2 Tracking a Mini Robot

Object Motion The camera tracks a Khepera robot as it moves on the floor. The motion is cir-

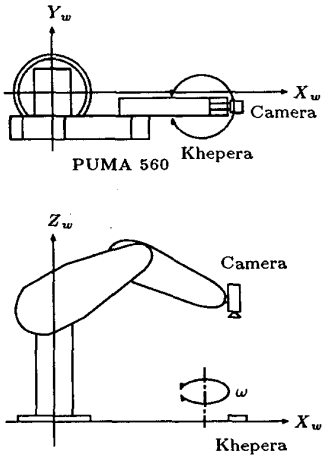

Figure 5: Experimental Setup

cular with radius 97 [mm]. We assume that the center of circle is known. Two cases with velocities 0.9 [rad/sec] and 1.8 [rad/sec] are examined. The experimental setup is depicted in Figure 5. The feature is the center of the Khepera in the image. The camera tracks the object in a plane parallel to the floor, that is, the orientation and the height of the camera are kept constant.

Let the object position be $[X_o\ Y_o\ 0]^T$. Orientation is not significant because the object is considered as a point. Then the generalized coordinates of the object becomes $p = [X_o\ Y_o]^T$. Using $\partial s_o/\partial p = [1\ 1\ 0\ 0\ 0\ 0]^T$ yields

$$L = J_f {}^cR_w [1\ \ 1\ \ 0\ \ 0\ \ 0\ \ 0]^T. \tag{23}$$

The observer estimates the rotational velocity of the object.

The Khepera robot starts to move counterclockwise at $t = 5$ [sec] with rotational velocity $\omega = 0.9$ [rad/sec]; changes its velocity to $\omega = 1.8$ at $t = 20$; and stops at $t = 35$. After 5 seconds of rest, Khepera starts to move again clockwise with $\omega = -0.9$; changes the velocity to $\omega = -1.8$ at $t = 54$; and stops at $t = 69$.

Control Law In this experiment, the controlled degree of freedom is two, i.e., the X and Y coordinates of the camera position in the world coordinate system. Thus to simplify the controller the controlled variable z is set to ξ. In other words, we choose $B = I$. Also, since the optical axis of the camera is aligned to the Z_w axis which is orthogonal to the floor, the image

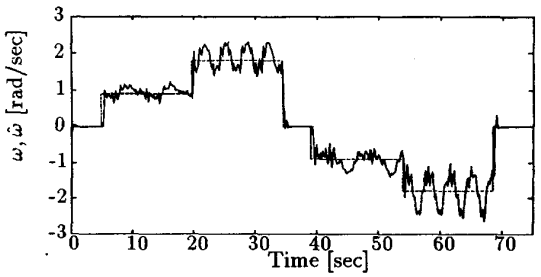

Figure 6: Estimated Rotational Velocity ($\hat{\omega}$)

Jacobian linearized at the reference position can be simplified to

$$J = -\frac{f}{Z_r} {}^cR_w \frac{\partial \phi}{\partial q}, \tag{24}$$

where Z_r is the camera height at the reference position.

Experiment Figure 6 shows the estimated value of ω. The solid line is the estimated velocity and the dashed line is the true velocity. The observer estimates the velocity fairly accurately but oscillations are found. The oscillations are due to the calibration error of the rotation center. Also, the arm configuration becomes almost singular (the arm is almost stretched out) when the object is at the farthest position. Thus the joint control accuracy is not very good around the farthest point. This singularity problem is another reason for oscillation.

Figures 7 (a) and (b) show the error in the image plane for x and y directions, respectively. The solid line is the result with the observer and the broken line is the result without the observer. Note that the error is reduced for both direction by using the observer.

Figures 8 (a) and (b) show the feedforward term $-LW\hat{\theta}$ and the feedback term $-K\hat{\xi}$ of the observer-based controller. The feedforward input is larger than the feedback input and the magnitude of the feedforward input changes with the object velocity. Thus the object tracking is mainly driven by the feedforward input and the feedback is used to correct the tracking error.

6 Conclusions

The vision sensor includes delay in its structure and the sampling rate is usually very slow. Thus, increasing the feedback gain yields oscillations and feedforward type controllers are effective. We have intro-

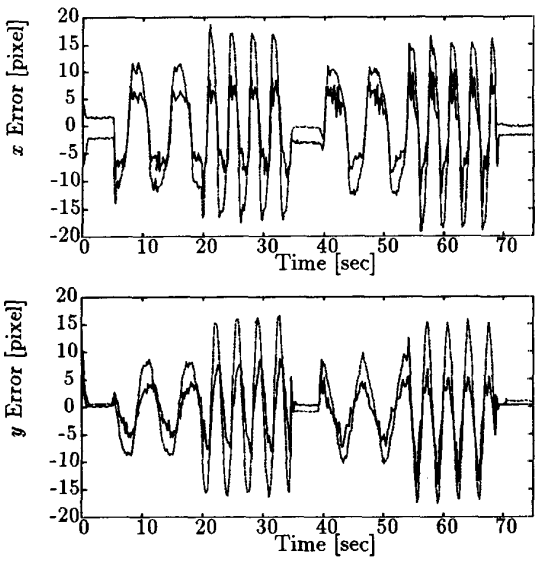

Figure 7: Controlled Error
—: With Observer, - -: Without Observer

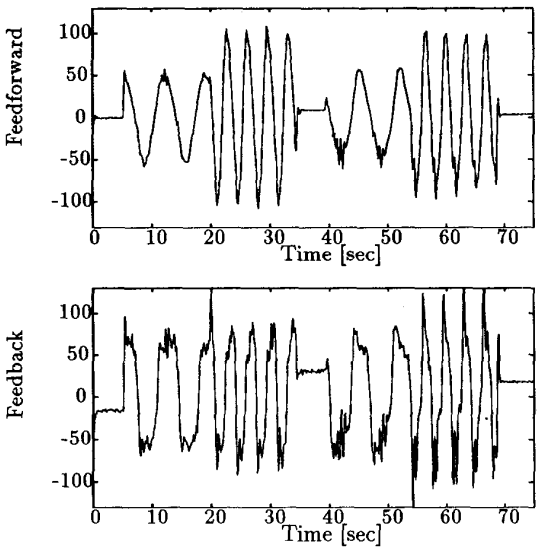

Figure 8: Control Input
(a) Feedforward ($-LW\hat{\theta}$) (b) Feedback ($-K\hat{\xi}$)

duced a visual feedback controller with a velocity feedforward that compensates the vision delay and generates the inter-sample information. Stability of the observer-based control system is presented in a nonlinear form and a linearized version is derived to reduce the computational complexity. Experimental results with PUMA 560 have shown the stable and accurate performance of the observer-based visual servo controller. The results emphasizes the usefulness of object velocity feedforward in the application of vision-based control.

References

[1] P. K. Allen, A. Timcenko, B. Yoshimi, and P. Michelman. Automated tracking and grasping of a moving object with a robotic hand-eye system. *IEEE Trans. Robotics and Automation*, 9(2):152–165, 1993.

[2] B. K. Ghosh and E. P. Loucks. A realization theory for perspective systems with application to parameter estimation problems in machine vision. *IEEE Trans. on Automatic Control*, 41(12):1706–1722, 1996.

[3] B. K. Ghosh, E. P. Loucks, and M. Jankovic. Multistage nonlinear estimation with application to image based parameter estimation. In *13th IFAC World Congress, Vol.F*, pages 447–452, San Francisco, 1996.

[4] K. Hashimoto and H. Kimura. Visual servoing with nonlinear observer. In *IEEE Int. Conf. Robotics and Automation*, pages 484–489, Nagoya, Japan, 1995.

[5] K. Hashimoto and T. Noritsugu. Performance and sensitivity in visual servoing. In *IEEE Int. Conf. Robotics and Automation*, pages 2321–2326, Leuven, Belgium, 1998.

[6] A. J. Koivo and N. Houshangi. Real-time vision feedback for servoing robotic manipulator with self-tuning controller. *IEEE Trans. Systems, Man, and Cybernetics*, 21(1):134–142, 1991.

[7] A. A. Rizzi and D. E. Koditchek. An active visual estimator for dexterous manipulation. *Trans. on Robotics and Automation*, 12(5):697–713, 1996.

[8] A. Teel, R. Kadiyala, P. Kokotovic, and S. Sastry. Indirect techniques for adaptive input-output linearization of non-linear systems. *Int. J. Control*, 53(1):193–222, 1991.

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