Proceedings of the 37th IEEE
Conference on Decision & Control
Tampa, Florida USA • December 1998

Control Issues in Active Vision: Open Problems and Some Answers

Mario Sznaier [1]
Department of Electrical Engineering
The Pennsylvania State University
University Park, PA 16802
msznaier@frodo.ee.psu.edu

Octavia I. Camps [2]
Department of Electrical Engineering
Department of Computer Science and Engineering
The Pennsylvania State University
University Park, PA 16802
camps@whale.ee.psu.edu

Abstract

Recent hardware developments have rendered controlled active vision a viable option for a broad range of problems, spanning applications as diverse as Intelligent Vehicle Highway Systems, robotic-assisted surgery, 3D reconstruction, inspection, vision assisted grasping, MEMS microassembly and automated spacecraft docking. However, realizing this potential requires having a framework for synthesizing robust active vision systems, capable of moving beyond carefully controlled environments. In addition, in order to fully exploit the capabilities of newly available hardware, the control and computer vision aspects of the problem must be addressed jointly. In this paper we illustrate with a simple example the control-related issues involved in active vision and we show how some very recently developed control and computer vision techniques can be brought to bear on the problem. These results also point out new research directions and possible extensions of currently available techniques.

1 Introduction

Visual servoing systems, i.e. systems incorporating vision as an integral part of the loop appeared as far back as late 1970's [12]. Even though these earlier systems were relatively slow and had limited scope [6], it was already recognized at this time that the relatively large latency time involved posed a threat to closed-loop stability. Since then, recent advances in hardware have opened the possibility of having systems with performance comparable to that of the human occulomotor system [7]. However, realizing this potential requires control laws capable of fully exploiting these new hardware capabilities. Moreover, earlier attempts to improve performance using simple controllers such as PID resulted in closed loop systems with poor stability properties and performance far worse than expected from the sampling rates employed [5]. This prompted a significant increase in the amount of research being carried out, both from computer vision (attempting to reduce the delay) and from control standpoints. An excellent survey of the state-of-the art as of 1996 and a comprehensive literature review up to then can be found in [14].

Earlier systems dealt with the stability issue by detuning the controller, at the expense of performance, until stability was experimentally accomplished. Latter approaches combine PID controllers with some prediction (either using Smith predictors or Kalman filters) to explicitly address the delay [6, 3]. However, these predictors can tolerate only small amounts of model uncertainty [18]. Thus, this technique works well only if the optical parameters of the camera and the depth can be precisely estimated. Moreover, the combination PID controller/predictor must be tuned using a trial and error process without guarantee of optimality. As noted in [5], due to the existence of modelling errors this entailed considerable experimentation to obtain a good compromise between stability and performance. Performance can be improved by using a two-degrees of freedom (2-DOF) controller [5], where the feedback controller (in this case a PI) stabilizes the loop and the feedforward controller (a predictor) improves tracking. However, while the use of a 2-DOF structure

[1] supported in part by NSF grants CDA-9529460 and ECS-9625920
[2] supported in part by NSF grants CDA-9529460 and IRI-9712598

yields better set–point tracking [17], it can improve neither robustness (related only to the feedback controller) nor disturbance rejection (in this case unanticipated target maneuvers).

Alternatively, the use of optimal control based LQG controllers has been proposed [21, 11, 10]. As indicated in [10] these controllers have the potential to minimize the effect of noise (provided that the features to be tracked are properly selected). However, optimizing performance requires tuning the weighting matrices by trial and error until satisfactory performance is achieved. Moreover, there are no a–priori guarantees of the robustness of the resulting system to time–delays or modelling errors. This has been experimentally corroborated by using a slightly defocused camera, where the mismatch between the actual and nominal focal distance causes the controller to fail [21].

This difficulty can be overcome to a certain extent by using a self–tuning controller that attempts to estimate the parameters in real–time, modifying the gains accordingly. Experimental results [20] show that this approach can tolerate errors ranging from 5% to 25%. However, these bounds are obtained a–posteriori, experimentally, with no attempt to design a controller optimizing the maximum tolerable error before instability occurs.

The issue of rendering the closed–loop system insensitive to calibration errors has been recently addressed in [8]. The main idea is to use time–varying proportional feedback (either based on the image Jacobian [8] or epipolar geometry [15]) to render the system asymptotically stable and robust to calibration errors, while exploiting the special form of the dynamics (containing an integrator) to guarantee (asymptotically) perfect tracking. When combined with a set of tools for fast visual tracking (XVision, [9]) this approach has been successfully used for robot motion control [8] and domain–independent navigation [15]. However, while the control algorithm is simple and easily implementable, the gain matrices must be empirically tuned to achieve good performance. An additional difficulty is that implementing the control action requires computing the image Jacobian, which is a nonlinear function of the distance from the camera to the feature. Finally, although this line of research addresses both the computer vision and control aspects of the controlled active vision problem, it uses a "separation principle" type approach where these aspects are treated, to a very large extent, independently.

It is worth stressing that the fact that dynamic control effects coupled with the presence of uncertainty are the factors limiting performance in visual closed–loop systems has long been recognized. Indeed, the motivation behind the different approaches mentioned above is to obtain a compromise between these factors leading to acceptable performance. However, only very recently there have been attempts to explicitly address some of these tradeoffs in a systematic way, by recasting the problem in an optimization form. For instance [22] proposed to optimize the size of the fovea in order to maximize the range of target velocities that can be successfully tracked by recasting the problem into an ℓ^1 optimization form. However, neither model uncertainty nor measurement errors were taken into account.

In this paper we illustrate with a simple example the control–related issues involved in active vision and we show how some very recently developed control and computer vision techniques can be brought to bear on the problem. These results also point out new research directions and possible extensions of currently available techniques.

2 Illustrative Example

The control–related issues involved in active vision can be illustrated by considering a simple model consisting of a moving target with a single feature[1] located at a point P, and a moving pinhole camera with a frame $\{R_s\}$ attached to it [20]. Let (X_s, Y_s, Z_s) and (x, y) denote the coordinates of P with respect to $\{R_s\}$ and of its projection P_i on the image plane, respectively. Assuming a focal distance f for the lens and taking an ideal perspective projection yields:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{f}{Z_s} \begin{bmatrix} X_s \\ Y_s \end{bmatrix} \tag{1}$$

Assuming that the camera moves with translational velocity $T \doteq (T_x, T_y, T_z)$ and angu-

[1] This simple model displays all the pathologies of the problem, without the added complexity resulting from additional features. Additional features can be accommodated by simply stacking the image Jacobians, although care must be exercised to guarantee controllability of the resulting augmented system [10].

lar velocity $\theta \doteq (\theta_x, \theta_y, \theta_z)$, the velocity of the point P with respect to $\{R_s\}$ is given by $\frac{dP}{dt} = -T - \theta \times P$. Finally, taking time derivatives of (1) yields the following optical flow equations:

$$\begin{aligned} \dot{x} &= x\frac{T_z}{Z_s} - \frac{fT_x}{Z_s} + \frac{xy}{f}\theta_x - (f + \frac{x^2}{f})\theta_y + y\theta_z \\ \dot{y} &= y\frac{T_z}{Z_s} - \frac{fT_y}{Z_s} + (f + \frac{y^2}{f})\theta_x - \frac{xy}{f}\theta_y - x\theta_z \end{aligned} \quad (2)$$

Using (2) and assuming a sampling time τ, we obtain a state–space model relating the predicted next position of the feature in the image to its present position, θ and T:

$$x_f(k+1) = A_f x_f(k) + B_f u(k) + E_f v(k) \quad (3)$$

where the state $x_f \doteq (x, y)^T \in R^2$ represents the position of the feature, $v \doteq (v_x, v_y)^T \in R^2$ represents an exogenous disturbance, $u \doteq (T_x, T_y, T_z, \theta_x, \theta_y, \theta_z)^T \in R^6$ represents the control input, and where the matrices are given by:

$$\begin{aligned} A_f &= I_{2\times 2},\ E_f = \frac{\tau f}{Z_s} I_{2\times 2}, \\ B_f &= \tau \begin{bmatrix} \frac{-f}{Z_s} & 0 & \frac{x}{Z_s} & \frac{xy}{f} & -\frac{f^2+x^2}{f} & y \\ 0 & \frac{-f}{Z_s} & \frac{y}{Z_s} & \frac{f^2+y^2}{f} & -\frac{xy}{f} & -x \end{bmatrix} \end{aligned} \quad (4)$$

Finally, assuming that the measurement available to the controller is the position of the feature corrupted by measurements errors (for instance due to noise and blurring) the output equation is given by:

$$\xi_f(k) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} w_x \\ w_y \end{pmatrix} \quad (5)$$

where $\xi_f \doteq (\xi_x \quad \xi_y)^T$ represents the measured position of the feature in the image plane and $\omega = (w_x \quad w_y)^T$ represents noise. Using (3)–(5), the controlled active vision problem can be recast into the following optimization problem:

Problem 1 *Find a control law $u(k) = f(\xi_f, k)$ such that the resulting closed loop system is asymptotically stable and $||e(t)|| \doteq ||x_f - x_{ref}||$ is minimized; where $||.||$ represents a norm suitable for the intended application and where x_{ref} represents the desired position for the target feature.*

In the sequel we consider the problem of smooth pursuit, where $v(k)$ models an unknown (but bounded) target velocity and the goal is to keep the image of the target as close as possible to the origin of coordinates in the image plane. Thus here $x_{ref} = 0$. In this context, the minimization of $||e(t)||$ arises naturally both from cost and stability considerations, since it is related to the size of the region where the target is expected to be at any given time. Clearly, if a multiresolution strategy is to be implemented, this size determines the size of the required fovea [22]. In addition, the larger this region is, the more computationally expensive it becomes to find the target in each frame, resulting in larger time–delays, which in turn can compromise stability [5].

In order to further simplify the example, assume that the camera is constrained only to pan and tilt motions and that the depth Z_s remains constant. Under these assumptions u and B_f in (3) further reduce to:

$$\begin{aligned} u &\doteq (\theta_x, \theta_y)^T \in R^2 \\ B_f &= \tau \begin{bmatrix} \frac{xy}{f} & -\frac{f^2+x^2}{f} \\ \frac{f^2+y^2}{f} & -\frac{xy}{f} \end{bmatrix} \end{aligned} \quad (6)$$

Finally, if the change in the coordinates of the target feature is small, equation (3) can be linearized around the present position, yielding an (approximately) equivalent Linear Time Invariant problem, that can be solved using a number of techniques either classical (PID [21, 5], pole–placement [21, 5]) or modern (Linear Quadratic Gaussian Optimal[11, 21, 20]).

Figure 1a shows the response of the nonlinear model to a step velocity profile for the target achieved by an LQG controller. Here we have assumed that $f = 75mm$, $Z_s = 10$m, a sampling rate of $10Hz$ and a measurement noise level of 5%, values consistent with a setup using a Bisight binocular head and a dedicated pipelined processor for the image processing. As shown in the figure, the closed–loop system appears to have good performance, with a settling time of 0.5 seconds and no overshoot. Indeed, it can be shown that in the absence of noise this controller achieves perfect tracking. However, experimental results [21] show that actual performance can be far worse than expected based on simulation results, even in cases where the actual parameters of the system closely match the values used for synthesizing the controller. To a large extent this is due to the fact that the simple model (3) does not take into account the computational time required by the image processing algorithms to locate the object in each

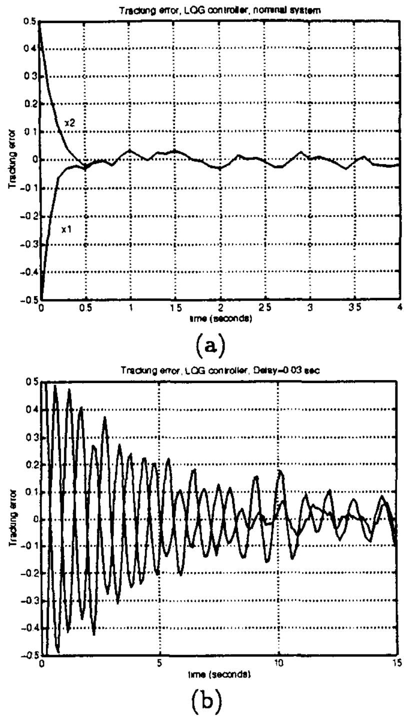

Figure 1: Tracking error for an LQG controller: (a) no delay (b) 0.03 sec. delay

frame. This computational time can be modelled as time delay T_d (proportional to the image size) appearing at the controller input. In active vision applications the value of this delay can be a substantial fraction of the sampling time, leading to degraded performance or even instability. This effect is illustrated in Figure 1b showing the effect of a 0.03 sec delay (consistent with image processing done using a pipelined MV250 board). As shown there this delay results in substantial oscillations and an 30^{th}–fold increase in the settling time. Moreover, increasing the delay to 0.05 seconds renders the system unstable. In principle, these delays can be handled by using a Smith Predictor [6, 3]. However, this technique cannot handle modelling errors [21], arising in this case from variations in the optical parameters, changes in the depth Z_s, linearization and unmodelled head dynamics.

This example highlights several features that make the problem challenging from a control stand-point, even when assuming that computer–vision related issues have already been addressed. In particular the problem includes:

1. **Presence of uncertain, potentially substantial time–delays** arising from the amount of time required by the image processing to locate features in each frame. This delay depends on the size of the image and the initial "guess" for the location of the feature.
2. **Uncertain and time–varying parameters.** Uncertain parameters include for instance the optical parameters of the camera (unless it is carefully calibrated). In addition, the depth Z_s (entering the dynamics in a nonlinear way) changes as the object moves.
3. **Conflicting performance specifications.** For instance the control action should be small in order to avoid actuator saturation limits. At the same time, the tracking error should stay small, since it is directly related to the size of the region where the target is expected to be at any time. The larger this region, the more time–consuming the image processing is resulting in larger time–delays which in turn can compromise stability. Additional specifications include good rejection of noise having different characterizations and adequate settling time.
4. **Nonlinearities,** since the dynamic equations (3)- (4) are nonlinear in the states x, y and the parameter Z_s

2.1 A possible solution using multiobjective robust control tools

The difficulties mentioned above can be overcome by modelling (albeit in a conservative fashion) the time delay as multiplicative dynamic uncertainty. Robust control techniques such as $\mathcal{H}_\infty$ can then be brought to bear on the problem, yielding a controller guaranteed to stabilize the closed–loop for all possible values of the time delay $T_{min} \leq T_d \leq T_{max}$. Moreover, by expanding the uncertainty description to include variations in the optical parameters, a range of values for Z_s, and the modelling error incurred by approximating the nonlinear model (3) by its linearization, it is possible to obtain a controller guaranteeing acceptable performance for a broader range of operating conditions at the price of using a higher order controller and more involved synthesis process.

Figure 2 shows the responses achieved by a controller designed using μ–synthesis [18] corresponding to $T_d = 0$ (nominal) and 0.05 seconds.

By experimenting with this problem we have found that there is a severe tradeoff between ro-

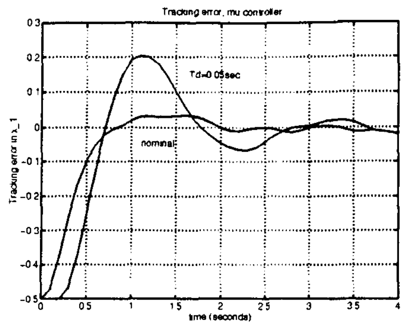

Figure 2: Tracking error for the μ–controller

bustness, time–delay tolerance, control action, settling time and controller complexity. Finally, an additional important tradeoff can be made apparent by examining the origin reachable sets (i.e. the region in the image that can be reached by the target), assuming that the target speed is unknown but bounded (for simplicity, and without loss of generality we have taken this bound here to be $1m/s$). Clearly, the smaller this region is, the less costly the image processing. Figure 3 shows the reachable regions corresponding to a controller synthesized using μ–synthesis, an LQG controller and a controller specifically designed to minimize this region (using a $\star$–norm approach [4]). Once more, the tradeoff between robustness, control action and performance (measured now in terms of the size of the reachable set) becomes apparent.

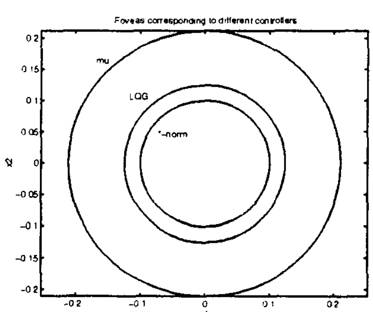

Figure 3: Foveas for different controllers

In this simpler case μ–synthesis yielded a controller achieving adequate performance. However, the problem was shifted from designing a controller to finding appropriate weighting functions, a process that requires both considerably design skills and multiple trial and error iterations. Alternatively, these conflicting performance specifications can be analytically traded–off by using multiobjective robust control tools[23, 24, 25]. To this effect the (linearized) plant is recast into the form shown in Figure 4.

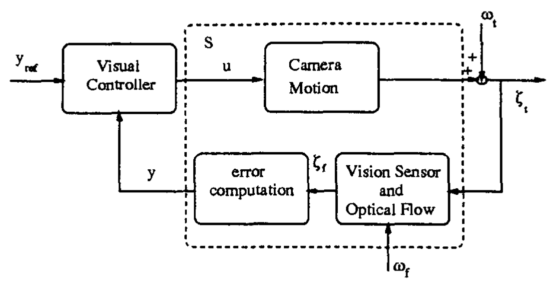

Figure 4: Visual Servoing as a Multiobjective Control Problem

In this context, the visual servoing problem can be stated in the following form:

Problem 2 *Find an internally stabilizing controller $u(z) = K(z)y(z)$ such that the following performance specifications are satisfied:*

1. $||T_{\zeta_t \omega_t}(z)||_* \leq \gamma_t$; *where* $||.||_*$ *indicates a suitable norm such as* $||.||_1$ *or* $||.||_2$;
2. $||T_{\zeta_f w_f}(z)||_{\mathcal{H}_\infty} \leq \gamma_f$;
3. $||W(z)T_{\zeta_t \omega_t}(z)||_{\mathcal{H}_\infty} \leq 1$; *where $W(z)$ is a suitable weighting function.*

and solved using the techniques proposed in [25]. Here, the first specification is related to both the maximum admissible tracking error and the size of the required fovea[2], the second is related to (bounded energy) noise rejection and the third is related to robustness against unmodelled dynamics and time–delays.

3 Improving Performance

As illustrated in the last section, robust control tools recently developed have the potential to successfully address the challenging features of visual servoing problems, by embedding them into an uncertainty structure. However, robustness against this "uncertainty" is achieved at the cost of perhaps substantial performance degradation, since the resulting robust controller tries to guarantee performance

[2]the choice of norms depends on whether it is desired to minimize the *RMS* or *peak* value of the error.

against situations that never arise. For instance, the robust controller used in the example not only guarantees stability against time–delays of up to 0.05 seconds but against *any* model uncertainty $\Delta(s)$ such that $\|W(s)\Delta(s)\|_\infty \leq 1$, where $W(s) = \frac{0.11s+1}{0.05s+1}$. A similar situation arises when linearization errors are handled in this way. Finally, multiobjective robust controllers may have high order[25], leading to implementation difficulties. In this section, we briefly discuss some of the potential solutions to these drawbacks.

1. **Controller complexity:** can be addressed by using model reduction, but this step is far from trivial due to the multiple performance specifications involved. An alternative is to use Linear Matrix Inequalities to optimize upper bounds of the performance[4]. This leads to low complexity controllers at the prize of some performance degradation.

2. **Nonconservative handling of time varying parameters:** Equations (3)–(4) depend nonlinearly in the depth Z_s. Since Z_s^{-1} multiplies the control action, changes in Z_s effectively act as a varying gain in the loop and, if large enough, can render the system unstable. If the changes in Z_s are relatively small, they can be accommodated by modelling the depth as a nominal value Z_o, subject to uncertainty. One can design a linear robust controller capable of accommodating these changes, at the expense of performance. Experimenting with the simple model used in section 2 we found out that this approach works well for small changes in Z_s, but entails substantial performance loss if the variations in the depth are not small. This issue can be addressed by measuring Z_s in real time, recasting the problem into a Linear Parameter Varying (LPV) form and exploiting very recently developed tools dealing with robust performance [27] and time delays [26] in this context. However, an additional issue that needs to be resolved is the fact that the usual LPV formalism assumes that the value of the time–varying parameter is *instantly* available to the controller, while in visual servoing problems there is a delay until Z_s can be computed from stereo information.

3. **Nonconservative handling of nonlinearities:** The nonlinearities in equations (3) can be handled by using the nonlinear controllers originally proposed for linear systems in [2], latter expanded to nonlinear systems in [16]. The basic idea of the method exploits a geometrical interpretation of the peak–to–peak (ℓ^1) norm to recast the synthesis problem as the problem of minimizing the size of the largest set contained in a given region that can be rendered invariant by an appropriate control action (in the context of visual servoing this amounts to optimizing the size of the fovea). However, the method needs to be expanded to handle multiple performance specifications and time delays.

4. **Reducing the Time Delay** The time required to estimate the target position can be substantially reduced by matching the appearance of local features[13]. The appearance of a feature is given by the intensity values in a window around the feature, and can be modeled *offline* using a set of training images, compactly stored as manifolds in relatively low dimensional spaces obtained by a Karhunen-Loeve (K-L) reduction. This approach has the advantage that it processes only a limited area of the image (the reachable set of the features being tracked, which is minimized by the controller) and thus is computational efficient as well as robust to occlusion. Additionally, data noise models can be propagated through this K-L reduction to provide the controller tight sensor error bounds.

4 Conclusions

Recent hardware developments have opened up the possibility of applying active vision techniques to a broad range of real–world problems, such as Intelligent Vehicle Highway Systems, robotic–assisted surgery, 3D reconstruction, inspection, vision assisted grasping, MEMS microassembly and automated spacecraft docking. A salient feature common to all these applications is that using a feedback structure incorporating the visual information in the loop (as opposed to open loop control) offers the possibility of achieving acceptable performance even in the presence of process modelling errors, noise and measurement noise, stemming for instance from poorly calibrated cameras, blurring or only partially determined feature correspondences between images.

However, as noted in a recent workshop [1] involving both control and computer vision researchers, while there seems to be a consensus in these communities about the implicit power of visual control, actually realizing this potential requires controllers capable of accommodating, in addition to uncertainty, the substantial time delays and nonlinearities typical of visual servoing problems.

In this paper we use a very simple example to illustrate the pathologies present in these problems and we indicate how recently developed robust control and computer vision techniques can be brought to bear on the problem. The paper concludes by analyzing different ways to reduce conservatism and by pointing out directions to extend currently available formalisms (such as multiobjective control and LPV systems) to make them better suited for applications involving computer vision in the feedback loop.

References

[1] In *Block Island Workshop on Vision and Control.* Block Island, Rhode Island, June 1997. Sponsored by the Army Research Office, the National Science Foundation, the Air Force Office of Scientific Research, and the Yale University Faculty of Engineering.

[2] F. Blanchini and M. Sznaier. Persistent disturbance rejection via static state feedback. *IEEE Trans. Autom. Contr.*, 40(6):1127–1131, 1995.

[3] C. Brown, D. Coombs, and J. Soong. Real time smooth pursuit tracking. In A. Blake and A Yuille, editors, *Artificial Intelligence*, pages 123–136. MIT Press, 1993.

[4] J. Bu and M. Sznaier. A linear matrix inequality approach to synthesizing low order mixed $\ell^1/\mathcal{H}_p$ controllers. In *1997 American Control Conf.*, pages 515–519, 1997.

[5] P. I. Corke and M. C. Good. Dynamic effects in visual closed–loop systems. *IEEE Trans. on Robotics and Automation*, 12(5):671–683, 1996.

[6] N. J. Ferrier and J. J. Clark. The Harvard binocular head. *Int. J. of Pattern Recognition and Artificial Intelligence*, 7(1):69–84, 1993.

[7] J. Fiala, R. Lumia, K. Roberts, and A. Wavering. TRICICLOPS: a tool for studying active vision. *Int. J. of Computer Vision*, 12(2/3):231–250, 1994.

[8] G. Hager. A modular system for robust positioning using feedabck from stereo vision. *IEEE Trans. on Robotics and Automation*, 13(4):582–595, 1997.

[9] G. Hager and K. Toyama. XVISION: A portable substrate for real-time vision applications. *Computer Vision Graphics and Image Processing: Image Understanding*, 69(1):23–37, 1998.

[10] K. Hashimoto, T. Ebine, and H. Kimura. Visual servoing with hand–eye manipulator–optimal control approach. *IEEE Trans. on Robotics and Automation*, 12(5):766–774, 1996.

[11] K. Hashimoto and H. Kimura. LQ optimal and nonlinear approaches to visual servoing. In K. Hashimoto, editor, *Visual Servoing*, volume 7, pages 165–198. World Scientific Series in Robotics and Automated Systems, World Scientific, Singapore, 1993.

[12] J. Hill and W. T. Park. Real time control of a robot with a mobile camera. In 9^{th} *ISIR*, pages 233–246, Washington D.C., March 1979.

[13] C. Y. Huang, O. I. Camps, and T. Kanungo. Object recognition using appearance-based parts and relations. In *Proc. IEEE Computer Vision and Pattern Recognition*, pages 877–883, 1997.

[14] S. Hutchinson, G. D. Hager, and P. I. Corke. A tutorial on visual servo control. *IEEE Trans. on Robotics and Automation*, 12(5):651–670, 1996.

[15] D. Kriegman, G. Hager, O. Ben-Shahar, and A. S. Georghiades. Towards domain independent navigation: Dynamic vision and control. Proc. IEEE Conf. Dec. Control, to appear, 1998.

[16] Wei-Min Lu. Attenuation of persistent $\mathcal{L}^\infty$ bounded disturbances for nonlinear systems. In *34 IEEE Conf. Decision and Control*, pages 829–834, New Orleans, december 1995.

[17] M. Morari and E. Zafiriou. *Robust Process Control.* Prentice Hall, New Jersey, 1989.

[18] A. Packard and J. Doyle. The complex structured singular value. *Automatica*, 29(1):71–110, 1993.

[19] N. P. Papanikolopoulos. Integrating computer vision and control for vision assisted robotics tasks. In *1995 American Control Conf.*, pages 904–908, Seattle, WA, June 1995.

[20] N. P. Papanikolopoulos, P. K. Khosla, and T. Kanade. Visual tracking of a moving target by a camera mounted on a robot: A combination of control and vision. *IEEE Trans. on Robotics and Automation*, 9(1):14–35, 1993.

[21] R. Sánchez Peña and M. Sznaier. *Robust Systems, Theory and Applications.* John Wiley, New Jersey, 1998.

[22] H. Rotstein and E. Rivlin. Optimal servoing for active foveated vision. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 177–182, June 1996.

[23] M. Sznaier. An exact solution to SISO mixed $\mathcal{H}_2/\mathcal{H}_\infty$ problems via convex optimization. *IEEE Trans. Autom. Contr.*, 39(12):2511–2517, December 1994.

[24] M. Sznaier. Mixed $l_1/\mathcal{H}_\infty$ controllers for SISO discrete time systems. *Systems & Controls Letters*, 23:179–186, 1994.

[25] M. Sznaier and J. Bu. Mixed $\ell_1/\mathcal{H}_\infty$ control of mimo systems via convex optimization. *IEEE Trans. Autom. Contr.*, to appear 1998.

[26] F. Wu and K. Grigoriadis. LPV systems with parameter–varying time delays. In 36^{th} *IEEE CDC*, pages 966–971, San Diego, CA, December 1997.

[27] F. Wu, X. H. Yang, A. Packard, and G. Becker. Induced $\mathcal{L}_2$ norm control for LPV systems with bounded parameter variation rates. *Int. J. Robust and Nonlinear Contr.*, 6(9/10):983–998, 1996.