Robust Identification and Model (In)Validation of Active Vision Systems

T. Inanc[1], M. Sznaier[2], and O. Camps[2]

[1] ECE Department, University of Louisville, Louisville, KY 40292,
t.inanc@louisville.edu
[2] ECE Department, Northeastern University, Boston, MA 02115,
{msznaier,camps}@ece.neu.edu

Summary. Recent hardware advances have rendered active vision a viable option for a diverse spectrum of applications ranging from MEMS manufacture to assisting individuals with disabilities. However, there are relatively few instances where these techniques have been successfully applied in uncontrolled environments. This can be traced to the *fragility* of active vision systems designed using classical methods. In this chapter we show how a combination of robust identification and model (in)validation tools can be used to obtain models of these systems that combine the dynamics of the physical hardware and the image processing algorithms, along with an associated worst–case uncertainty description, suitable to be used in the context of modern robust control tools. As shown here, this combination leads to systems that achieve good tracking performance under a wide range of conditions. These results are experimentally validated using a BiSight/UniSight robotic head–eye platform.

Key words: Active Vision, Visual Servoing, Robust Control, Robust Identification, Model (In)Validation.

1 Background

1.1 Introduction and Motivation

In the past few years, active vision systems – i.e. systems incorporating vision as an integral part of the loop – have emerged as a viable option for a large number of applications, ranging from vision–based assembly [21, 22, 55, 67, 79] to vision–assisted surgery [34, 38, 45, 77, 78], assisting individuals with disabilities [41, 66, 68, 76], and intelligent vehicle highway systems [5, 24, 65, 74]. In practice, using active systems in dynamic scenes requires both real-time visual processing and real-time closed-loop control. Recent hardware developments have make this now possible, leading to a number of systems [15, 16, 19, 26, 56, 62].

Active vision systems appeared as far back as late 1970's [32] and have been the subject of much research since. An excellent survey of the state–of–the art as of 1996 and as of 2006 and a comprehensive literature reviews up to then can be found in [33] and [8, 9], respectively. Earlier active vision systems dealt with stability issues by detuning the controller, at the expense of performance, until stability was accomplished [33]. Latter approaches combined PID controllers with some prediction to explicitly address time delays [6, 23]. However, these predictors can tolerate only small amounts of uncertainty [53]. Furthermore, the combination PID controller/predictor had to be tuned empirically, a process that entailed considerable experimentation [17].

Set–point tracking can be improved by using a two–degrees of freedom (2–DOF) controller [17]. However, this approach can improve neither robustness nor disturbance rejection [44]. Optimal controllers have the potential to improve performance [29, 30, 51], but can lead to fragile systems [51]. Empirical results [50, 64] show that adaptive controllers can accommodate calibration errors. However, this approach does not allow for achieving an a–priori established robustness level or to balance robustness versus performance.

A switching approach to visual servo control was introduced in [25]. In this approach two controllers were designed: one that relied on the homography between initial and goal images, and a second one based on an affine transformation approximating the motion between the initial and goal camera configurations. However, stability of the switched system was not formally addressed and it is difficult to prove.

Recent work has recognized the fact that robustness issues are central to the success of active vision systems. Robustness to calibration errors and estimation noise has been addressed in [27, 73] and [61], respectively. 2.5D or Homography–based visual servo regulation and tracking was addressed in [20] and [10], respectively, where a Lyapunov–based adaptive control strategy was used to compensate for unknown depth measurements. However, a fixed camera position was assumed (for regulation) and uncertainties in the camera calibration parameters as well as mechanical system dynamics were not taken into account. Furthermore, the intrinsic[3] camera calibration matrix was needed for tracking.

Nonlinear tracking controllers were introduced in [80], where two position–tracking controllers were developed: One addressed nonlinear robot dynamics and the second compensated for parametric uncertainties in the dynamics as well as in the camera calibration parameters. However, this approach assumed a fixed camera configuration and a limited camera orientation, and required calculating the inverse Jacobian. Moreover, higher overshoots occured during the transient response due to the high number of parameter estimates needed by the adaptive controller. More recently, the adaptive controller proposed in [1] avoided overparameterization and eliminated the need for restricting the camera orientation, but it still required the calculation of the inverse Jacobian.

[3] The camera calibration parameters consist of intrinsic and extrinsic parameters. The intrinsic parameters are the image center, the camera scale factor and the camera magnification factor. The extrinsic parameters are the camera position and orientation.

An alternative approach relies on the use of self–calibration methods to measure the camera intrinsic parameters[14, 42, 54]. However, these techniques fail for camera–in–hand (camera translation) applications. Indeed, [69] showed that in these cases, it is impossible to measure the camera internal parameters. Visual servoing invariant to changes in camera calibration parameters was considered in [40].

An additional drawback of the approaches mentioned above, is the implicit use of a *"separation principle"*, where the computer vision and control aspects of the problem are treated, to a very large extent, independently. The interaction between these components was explicitly taken into account in [57], but neither model uncertainty nor measurement errors were considered.

In [71] we used a simplified scenario, restricted motion of a single target in the absence of clutter, as a vehicle to illustrate how to address the issues indicated above using robust control tools. Motivated by these results and those in [70], in this chapter we show that *how recently developed robust identification, , model (in)validation and robust control synthesis techniques can be brought to bear on the problem to synthesize robust active vision systems capable of delivering good performance for a range of conditions.*

This chapter is organized as follows: in Section 1 we introduce the notation that we use and give a summary of the relevant theoretical and algorithmic tools. In Section 2 we describe the identification and control issues that arise in the context of designing active vision systems and propose a solution based on a combination of robust identification/robust control tools. In Section 3 we provide experimental results validating tthis approach In section 4 we comment on the shortcomings of existing tools, indicate possible solutions, and benchmark the the resulting systems against those obtained using classical design methods. Finally, in section 5 we summarize our results and indicate directions for future research.

1.2 Notation

By $\mathcal{L}_\infty$ we will denote the Lebesgue space of complex valued matrix functions essentially bounded on the unit circle, equipped with the norm:

$$\|G(z)\|_\infty \doteq ess \sup_{|z|=1} \overline{\sigma}\,(G(z))$$

where $\overline{\sigma}$ denotes the largest singular value. By $\mathcal{H}_\infty$ we denote the subspace of functions in $\mathcal{L}_\infty$ with a bounded analytic continuation outside the unit disk, equipped with the norm $\|G(z)\|_\infty \doteq ess\sup_{|z|>1} \overline{\sigma}\,(G(z))$. Also of interest is the Banach space $\mathcal{H}_{\infty,\rho}$ of transfer matrices in $\mathcal{H}_\infty$ which have analytic continuation outside the disk of radius $\rho < 1$, i.e. the space of exponentially stable systems with a stability margin of $(1-\rho)$, equipped with the norm $\|G(z)\|_{\infty,\rho} \doteq \sup_{|z|>\rho} \overline{\sigma}\,(G(z))$. $\mathcal{H}_{\infty,\rho}(K)$ denotes the open K–ball in $\mathcal{H}_{\infty,\rho}$, i.e. $\{H \in \mathcal{H}_{\infty,\rho} \colon \|H\|_{\infty,\rho} < K\}$. In the sequel, we will use simply $\mathcal{BH}_\infty$ for the case where $K = \rho = 1$.

Finally, ℓ^2 denotes the space of bounded real sequences $h = \{h_i\}$ equipped with the norm $\|h\|^2_{\ell^2} \overset{\text{def}}{=} \sum_{i=0}^\infty h_i^2$. Similarly, ℓ^∞ denotes the space of bounded real sequences equipped with the norm $\|h\|_{\ell^\infty} \overset{\text{def}}{=} \sup_k |h_k|$.

1.3 Theoretical background

In this section we summarize, for ease of reference, the theoretical and algorithmic results used to identify and validate a *combined* model of the active vision system, and to synthesize a robust controller based on this model. A more detailed description can be found in several textbooks such as [11] and [53].

Robust Identification

In principle a detailed model of an active vision system can be obtained by using first principles to model the dynamics of both, the platform where the cameras are mounted and of the imaging processing algorithms. However, the resulting model will depend on several parameters, including the optical parameters of the imaging systems and the mass of a camera mounting structure such as a pan/tilt unit in our case. In addition, such an approach does not fully capture the interaction between the imaging and mechanical portions of the system, leading to poor performance or even instability. Finally, classical identification procedures [39] are not well suited to be used in combination with robust control techniques (see [53], Chapter 10).

To avoid these difficulties in this chapter we will use recently developed non–parametric robust identification techniques that, starting from experimental data and some mild *a priori* assumptions on the plant, generate a nominal model as well as bounds on the worst case identification error suitable to be used by robust control synthesis methods. In particular, since the experimental data used in this chapter originates from time–domain experiments and the controller synthesis algorithm requires $\mathcal{H}_\infty$ bounds on the uncertainty description, we will use $\mathcal{H}_\infty/\ell^1$ identification [12]. In the sequel we briefly cover the fundamentals of this technique. A tutorial to the robust identification field and key to the relevant references can be found for instance in [53], Chapter 10.

The experimental data used by the $\mathcal{H}_\infty/\ell^1$ method consists of the first N_t samples, corrupted by additive noise, of the time response of the system $y(k) = h(k) + \eta_t(k)$, $k = 0, \ldots, N_t - 1$ corresponding to a known input $u(k)$. The *a priori* assumptions are:

1. The system to be identified belongs to the class $\mathcal{H}_{\infty,\rho}(K)$ i.e. the set of exponentially stable systems with a stability margin of $(1 - \rho)$, and a peak response to complex exponential inputs of K.
2. A bound ϵ_t of the measurement noise $\eta_t(k)$, i.e. $\|\eta_t\|_{\ell^\infty} \leq \epsilon_t$ is known.

The goal of the method is twofold:

1. Firstly, to establish whether or not the experimental data and the *a priori* assumptions are consistent, i.e, whether or not there exist models in $\mathcal{H}_{\infty,\rho}(K)$ that interpolate the data points within the experimental errors (the "*consistency*" problem) and,
2. If so, to obtain one such model as well as a bound on the worst–case identification error.

It can be shown [52] that the consistency problem reduces to a (convex) Linear Matrix Inequality (LMI) feasibility problem that can be efficiently solved using commerciable available code. Specifically, the *a posteriori* experimental data and the *a priori* information are consistent if and only if there exists a vector $\mathbf{h} = [h_o, h_1, \dots, h_{N_t-1}]$ such that the following LMIs hold:

$$M_R(\mathbf{h}) = \begin{bmatrix} R^{-2} & \frac{\mathcal{F}^T}{K} \\ \\ \frac{\mathcal{F}}{K} & R^2 \end{bmatrix} > 0 \qquad (1)$$

$$-\epsilon_t < \mathcal{F}^T \mathbf{u} - \mathbf{y}^t < \epsilon_t$$

where $R = \text{diag}[\, 1 \;\rho\; \rho^2 \;\dots\; \rho^{N_t-1}\,]$ and

$$\mathcal{F} = \begin{bmatrix} h_0 & h_1 & \dots & h_{N_t-1} \\ 0 & h_0 & \dots & h_{N_t-2} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & h_0 \end{bmatrix} \qquad (2)$$

$$\mathbf{y} = [y_0, y_1, \dots, y_{N_t-1}]^T$$

$$\mathbf{u} = [u_0, u_1, \dots, u_{N_t-1}]^T$$

Once the generalized Pick Matrix M_R (1) is found, the set of all models consistent with both the *a priori* assumptions and the *a posteriori* experimental data can be parametrized as a Linear Fractional Transformation (LFT) of a free parameter $q(z) \in \mathcal{BH}_\infty$, $G(z) = F_\ell\,[L(z), q(z)]$, where $L(z)$ depends only on M_R. In particular, the choice $q(z) = 0$ leads to the central model, of order less than or equal to the number of experimental data points N_t . In this case an explicit state–space realization of $G(z)$ is given by [35]:

$$\begin{aligned} G(z) &= \left[\begin{array}{c|c} A_G & B_G \\ \hline C_G & D_G \end{array}\right] \\ A_G &= A - [C_-^T C_- + (A^T - I)M_R]^{-1} C_-^T C_-(A - I) \\ B_G &= -[(A^T - I)M_R + C_-^T C_-]^{-1} C_- \\ C_G &= C_+[(A^T - I)M_R + C_-^T C_-]^{-1} C_-^T C_-(A - I) - C_+(A - I) \\ D_G &= C_+[(A^T - I)M_R + C_-^T C_-]^{-1} C_-^T \end{aligned} \qquad (3)$$

where

$$A = \begin{bmatrix} 0 & I_{N_t \times N_t} \\ 0 & 0 \end{bmatrix},\; C_- = [\overbrace{1 \quad 0 \quad \dots \quad 0}^{N_t}],\; C_+ = \frac{\mathbf{h}^T R}{K}$$

Finally, the worst case identification error is bounded by [11]:

$$\|e\|_\infty \leq \sum_{k=0}^{N-1} \min\left\{\epsilon \sum_{i=0}^{k} |w_i|.\frac{K}{\rho^k}\right\} + \frac{K}{\rho^{N-1}(\rho-1)} \tag{4}$$

where w_i depends only on the known input signal[4].

Model (In)Validation

In the previous section we have outlined a robust identification algorithm that generates nominal models and uncertainty descriptions suitable to be used by robust control synthesis methods. However, before this description can be used by the control engineer, it must be *validated*, based on additional experimental data. This leads to the following model (in)validation problem:

- Given experimental data, corrupted by additive noise, find whether or not this data could have been produced by the combination of the nominal model and some uncertainty in the uncertainty set.
- If the answer is negative, then the assumed model does not provide a correct description of the physical system.

In addition these techniques allow for refining the uncertainty description, by establishing the lowest uncertainty level that does not invalidate the model. These bounds are usually much tighter than the one provided by equation (4), leading to less conservative controller design.

Model (in)validation of LTI systems has been extensively studied in the past decade (see for instance [11] and references therein). For ease of reference, we quote the main result showing that in the case of unstructured uncertainty entering the plant as an LFT (Linear Fractional Transformation), model (in)validation reduces to a convex optimization problem that can be efficiently solved.

Consider the interconnection shown in Figure 1, where $G_o = \left(\begin{array}{c|c} P & Q \\ \hline R & S \end{array}\right)$ is a known given system, $\varDelta$ represents unstructured bounded dynamic uncertainty, and the signals u and y represent a known test input and the corresponding output corrupted by measurement noise η. The *a priori* information consists on bounds on the "size" of the uncertainty, $\|\varDelta\|_\infty \leq \gamma$ and the magnitude of the measurement noise $\|\omega\|_\infty \leq \epsilon_N$. In this context it can be shown [13] that, given measurements of the input $\mathbf{u}$ and output $\mathbf{y}$, the model is not invalidated by this experimental information if and only if there exist a vector $\zeta = \begin{pmatrix}\zeta_1 \ldots, \zeta_n\end{pmatrix}$ such that the following set of LMIs are feasible:

[4] In the simplest case where $u = \delta$ then $w_0 = 1, w_i = 0, i \geq 1$

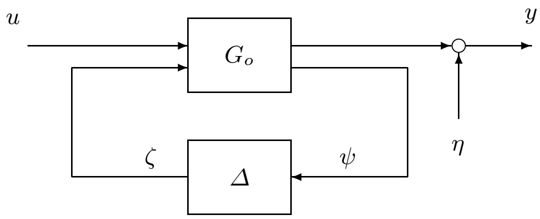


Fig. 1. Setup for Model (In)Validation

$$M(\zeta) \doteq \begin{bmatrix} (T_R T_u)^* T_R T_u + (T_R T_u)^* T_S T_\zeta + T_\zeta^* T_S^* T_R T_u & T_\zeta^* \left(\frac{1}{\gamma^2} I - T_S^* T_S\right)^{\frac{1}{2}} \\ \left(\frac{1}{\gamma^2} I - T_S^* T_S\right)^{\frac{1}{2}} T_\zeta & I \end{bmatrix} > 0 \qquad (5)$$

$$\|\mathbf{y} - T_P \mathbf{u} + T_Q \zeta\|_{\ell^\infty} \leq \epsilon_N$$

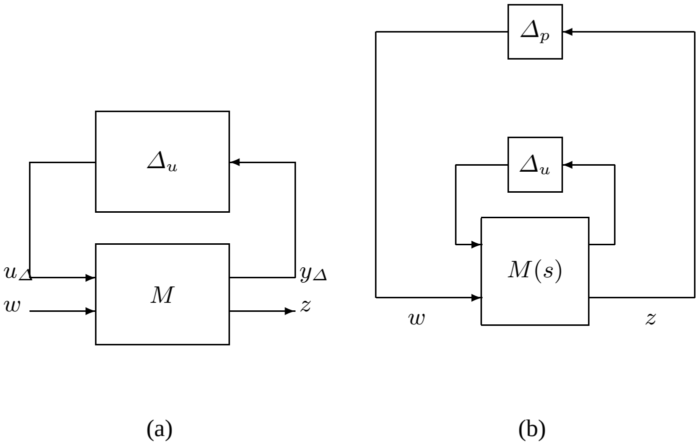
where, for a given sequence $h \in \ell^2$, T_h represents its associated lower Toeplitz matrix:

$$T_h = \begin{bmatrix} h_0 & 0 & \dots & 0 \\ h_1 & h_0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ h_{n-1} & h_{n-2} & \dots & h_0 \end{bmatrix} \qquad (6)$$

and, for a given linear operator $\mathcal{L}\colon \ell^2 \to \ell^2$, $T_{\mathcal{L}}$ denotes the lower Toeplitz matrix associated with its convolution kernel. Note in passing that since (5) is linear in $\frac{1}{\gamma^2}$, the problem of finding the tightest uncertainty bound such that the model is not invalidated by the experimental data leads to a convex LMI optimization problem.

Overview of Robust Controller Design via μ-synthesis

Once a nominal model of the plant has been identified and validated using the techniques outlined above, standard control techniques can be used to synthesize suitable controllers that achieve good performance for the *nominal* system. However, as noted in [31], the resulting closed loop system can be *fragile* to uncertainty and changes in the parameters of the camera, such as the focal length f. To avoid this difficulty, in this chapter we will use robust control tools (μ–synthesis) to design a controller, based on both the nominal model *and* the uncertainty description provided by the identification step, that achieves robust performance.



(a) (b)

Fig. 2. (a) LFT uncertainty description. (b) Equivalence between robust performance and stability.

Specifically, consider the generalized system interconnection shown in Figure 2, consisting of a stable transfer function matrix M (in our case the combination of the pant–tilt unit, the vision sensor and the controller) and a "feedback" term Δ_u, representing model uncertainty with a block diagonal structure of the form:

$$\begin{aligned}\Delta_u(s) \in \mathbf{\Delta} \equiv \Big\{ \; \Delta(s) = \text{block} - \text{diag}\big\{\Delta_1(s), \\ \Delta_2(s), \ldots, \Delta_n(s)\big\}, \;\; \Delta_i(s) \text{ stable}\Big\}\end{aligned} \tag{7}$$

It can be shown that this interconnection is stable for all $\|\Delta_u(s)\|_\infty \leq 1$ if and only if $\mu_{\Delta_u}(M) < 1$, where the Structured Singular Value μ is defined as[5] [18]:

$$\mu_{\Delta_u}(M) = \begin{cases} \frac{1}{\min\limits_{\Delta_u \in \mathbf{\Delta}} \{\|\Delta_u\|_\infty \;:\; det(I+M(j\omega)\Delta_u(j\omega))=0 \text{ for some } \omega\}} \\ 0 \text{ if no } \Delta_u \in \mathbf{\Delta} \text{ destabilizes } M \end{cases} \tag{8}$$

Robust performance (i.e. guaranteed performance for all possible model uncertainties in the set) can be addressed by recasting the problem into an augmented robust stability problem by introducing an additional fictitious perturbation block Δ_p, as shown in Figure 2(b), where w and z represent exogenous inputs and outputs subject to performance specifications, respectively. It can be shown (see the main loop theorem in [49]) that *robust performance* is achieved if and only if:

$$\mu_{RP} = \sup_\omega \mu_\Delta(M) < 1$$

where $\Delta = diag\{\Delta_p, \Delta_u\}$ contains now both the uncertainty and the performance blocks.

[5] Roughly speaking, μ is the inverse of the "size" of the smallest destabilizing perturbation.

μ provides a useful tool for robustness analysis that combines unstructured and structured uncertainty, robust stability and robust performance in a unified, non–conservative, framework. It can even be extended to cover parametric uncertainty (real μ). Unfortunately, at the present time there are no efficient algorithms for computing the exact value of μ for general perturbation structures. Instead, the following upper bound is used [18, 58]:

$$\mu_{\Delta}(M) \leq \inf_{D \in \mathcal{D}} \|DMD^{-1}\|_{\infty} \tag{9}$$

where D represents a set of positive definite Hermitian matrices with a diagonal block structure that commutes with that of Δ. It can be shown that problem (9) can be recast as a convex optimization problem, leading to efficient computational algorithms. Moreover, this upper bound coincides with the exact value of μ for perturbation structures having up to 3 blocks. For more than 3 blocks, the bound is no longer tight and recent results show that the gap can increase linearly in the number of blocks [43]. However, in most cases arising in practice the gap is small (less than 15% [49]).

From the discussion above it follows that controllers guaranteeing robust stability or robust performance can be synthesized by solving the following optimization problem:

$$\min_{\text{K stabilizing}} \mu_{\Delta}\Big\{M(K)\Big\}$$

where the notation $M(K)$ is used to indicate explicitly that the closed loop transfer matrix M is a function of the controller K. Due to the difficulties in computing the exact value of μ, the upper bound (9) is used instead, yielding the following optimization problem (in K and D):

$$J = \min_{\text{K stabilizing}} \Big\{ \inf_{D \in \mathcal{D}} \|DM(K)D^{-1}\|_{\infty} \Big\} \tag{10}$$

Robust stability or robust performance is achieved if $J < 1$.

While the optimization problem (10) is convex either in the scales D or in the controller K, *it is not jointly* convex in D and K. Thus, there potentially exist local minima where an optimization algorithm may get trapped. The solution method currently used alternates between finding the tightest possible upper bound by optimizing the scales D while holding the controller constant (an infinite dimensional convex optimization problem); and finding an internally stabilizing controller that minimizes this upper bound (a standard $\mathcal{H}_{\infty}$ control problem). This algorithm, known as the "D–K" iteration, is implemented both in the Robust Control Toolbox [59] and μ Analysis and Synthesis Toolbox [2], and can be summarized as follows:

1. $\mathcal{H}_{\infty}$ Synthesis. Holding D fixed, use $\mathcal{H}_{\infty}$ synthesis to solve:

$$\min_{K \text{ stabilizing}} \|DM(K)D^{-1}\|_{\infty} \tag{11}$$

In the first iteration D is often set to I(identity matrix). After the first iteration the D scale obtained in step 3 is used.

2. μ Analysis. Calculate the upper bound of μ for the closed loop system obtained using the controller K from step 1. This entails solving the following infinite dimensional optimization problem:
$$\inf_{D\in\mathcal{D}} \|DM(K)D^{-1}\|_\infty \tag{12}$$
This problem is approximately solved by finding the value of D over a finite grid of frequency points ω_i.
3. D Fitting. The approximate solution to the optimal scaling problem of step 2 is found by fitting the values $D(j\omega_i)$ with a real-rational, proper, stable, minimum-phase transfer matrix $D(s)$. Note that the order of the controller is that of the augmented plant (plant + weights) + 2× order of D. Thus, in order to obtain controllers with reasonable complexity, the order of D should be kept low (usually first or second order).
4. Go to step 1 until the stop criterion is met, which means that the condition $J < 1$ is satisfied.

As mentioned before, while due to the lack of joint convexity in K and D, this algorithm is not theoretically guaranteed to converge to the global minimum, it works well in practice, and has allowed for solving many difficult engineering problems (see for instance [63]).

2 Description of application example

The control–related issues involved in active vision can be illustrated by considering the problem of smooth tracking of a non–cooperative target, illustrated in the block diagram shown in Figure 3. Here the goal is to internally stabilize the plant and to track the target motion, y_{target}, using as measurements images possibly corrupted by noise.

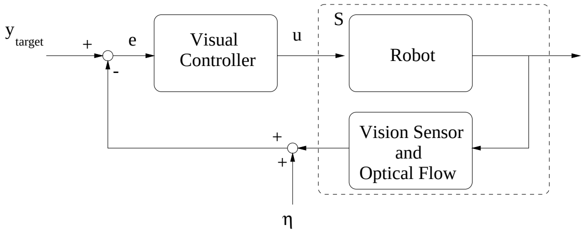


Fig. 3. Block Diagram of a Visual Tracking System.

It can be shown (see [51] for details) that a simplified model of the plant is given by the following state–space model:
$$\begin{aligned} \underline{x}(k+1) &= A_f\underline{x}(k) + B_f\underline{u}(k) + E_f\underline{v}(k) \\ \underline{y}(k) &= \underline{x}(k) + \underline{\eta}(k) \end{aligned} \tag{13}$$

where the state $\underline{x} \doteq (x, y)^T \in R^2$ represents the position of the feature in the image plane, $\underline{v} \doteq (v_x, v_y)^T \in R^2$ models the unknown (but bounded) target velocity, $\underline{u} \doteq (\theta_x, \theta_y)^T \in R^2$ represents the control input (pan and tilt motion of the camera), $\underline{y}$ represents the measured position of the feature corrupted by the noise $\underline{\eta}$, and where the matrices are given by:

$$A_f = I_2,\ E_f = \frac{\tau f}{Z_s(k)} I_2,$$

$$B_f = \tau \begin{bmatrix} \frac{x(k)y(k)}{f} & -(f + \frac{x^2(k)}{f}) \\ (f + \frac{y^2(k)}{f}) & -\frac{x(k)y(k)}{f} \end{bmatrix} \tag{14}$$

where f is the focal length of the camera, Z_s its distance to the object, and τ the sampling interval. In this context, the objective is to design a controller that stabilizes the model (13) and keeps $\|e\|$, the norm of the error signal, small.

If the change in the coordinates of the target feature is small, equation (13) 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 [17, 51], pole–placement [17, 51]) or modern (Linear Quadratic Gaussian Optimal [30, 50, 51]). However, while the resulting closed–loop systems appear to have good performance, experimental results show that actual performance can be far worse than expected based on simulations using the model (13). As discussed in [70], this is largely due to the fact that this approach takes into account neither the dynamical effects of the image processing algorithms used to locate the object in each frame, nor potentially destabilizing modelling errors that include variations in the optical parameters and unmodelled robot dynamics. The goal of this chapter is to illustrate how these effects can be taken into account by exploiting a combination of recently developed robust identification and model (in)validation techniques.

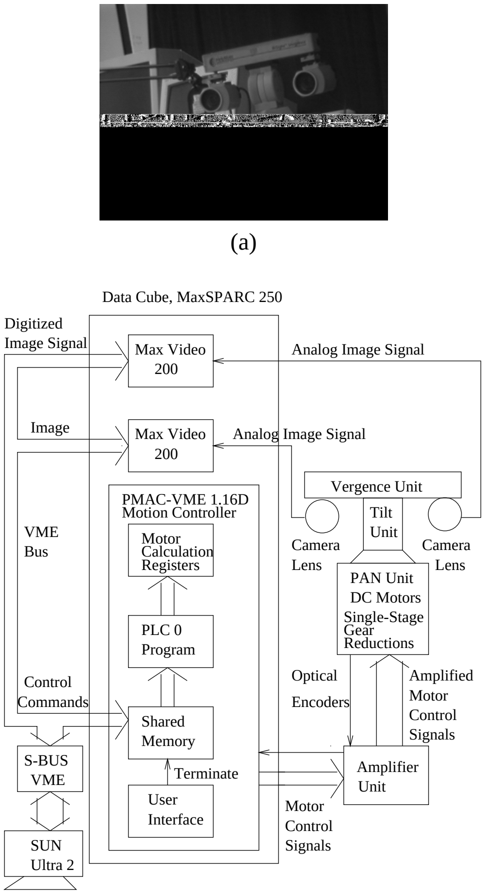
2.1 Hardware description

The hardware setup used in this chapter, shown in Figure 4(a), consists of a UniSight pan/tilt platform equipped with a BiSight stereo head containing two Hitachi KP-M1 cameras and Fujinon H10X11EMPX-31 motorized lenses. The input commands to the head and the lenses are given using a 10 channel PMAC $\delta - \tau$ controller, and the image processing required to capture the images and locate the target are performed using a Datacube MaxSPARC S250 hosted by a dual processor Sun Ultra 2 workstation. A block diagram of the complete system showing the interconnection of the various components is shown in Figure 4(b).

This robotic head has four mechanical degrees of freedom:

- Pan is the rotation of the inter–camera baseline about a vertical axis.
- Tilt is the rotation of the inter–camera baseline about a horizontal axis.
- Right Vergence is the rotation of the right camera about the vertical axis.
- Left Vergence is the rotation of the left camera about the vertical axis.

There is a mechanical coupling between pan and tilt axes: if the pan motor is turned while the tilt motor is held stationary, the tilt axis will also rotate by $1/4$th the amount of the pan axis rotation. However, moving the tilt motor by itself has no effect on the pan axis. To correct this coupling, the tilt motor must be turned by a ratio of the pan motor rotation to opposite direction of the pan motor motion. The system also has six optical degrees of freedom (zoom, focus and aperture of left and right cameras). The mechanical properties of the "head" can be found in [35].



(a)

(b)

Fig. 4. (a) The Experimental Setup. (b) Corresponding Block Diagram.

2.2 Tracking Algorithms

Next, we briefly discuss the choice of the image processing algorithms used in this chapter. The so called *"motion correspondence"* problem – i.e. to determine the image position of the object being tracked in the frames of the sequence–has been extensively studied in the computer vision literature, and a large number of techniques have been proposed, both for known and unknown objects (see for instance [3, 4, 7, 16, 28, 33, 36, 46, 56, 62, 75] and references therein). Correspondences between individual frames are usually integrated over time to exploit the dynamical properties of the target, using for instance Condensation trackers [46]. These trackers generalize Kalman–filter based ones by allowing more general (multimodal) observation noise models, although in some cases can result on impractical computational requirements [37].

Selection of the image processing algorithm entails a compromise between complexity and robustness, since time delays stemming from more sophisticated image processing algorithms have negative impact on the stability and overall performance of the closed–loop system. Thus, as a compromise between complexity and robustness, we selected a *normalized cross–correlation with template update* [36] algorithm to track the target through a sequence of frames. As shown in the sequel, this algorithm–briefly described below–achieves good performance tracking targets at video rate, even in the presence of clutter.

Let T be a template image representing a memorized pattern of the object of interest and I be the input image where the object must be found. The *normalized cross-correlation* between the template image T and a region R of the input image I is given by:

$$N(i,j) = \frac{\sum_{x,y \in R} T(x,y) \cdot I_{i,j}(x,y)}{||T(x,y)|| \cdot ||I_{i,j}(x,y)||} \tag{15}$$

where $I_{i,j}(x,y) = I(x-i, y-j)$, and the denominator is a normalization factor to reduce the effect of illumination variations. Regions R containing a pattern similar to the memorized template result on high values of $N(i,j)$ ($N(i,j) = 1$ corresponds to a perfect match). Thus, searching for the object of interest can be reduced to the problem of finding a region R in the input image I that has a high normalized cross-correlation value with the stored template T. Search time can be reduced by looking in a small window around the expected location of the target.

The approach outlined above allows for tracking targets without prior models, it is invariant to translation of the target and it is robust to illumination changes. Additional robustness to orientation and scale variations, to a certain degree, is obtained by using *template updating*. That is, the template is replaced by the best matched region from the current frame. However, in order to prevent *correlator walk-off* due to occlusion and image clutter, the template is updated only when the cross-correlation factor of the current frame drops from the previous value by more than a threshold. In our experiments we have used 128×128 images, 18×18 template size, 9×9 search window with increments of 2 pixels, and template update threshold of 0.04.

3 Experimental Results

Synthesizing a robust active vision controller requires first obtaining a model of the plant, as well as an estimate of the associated uncertainty. In the sequel we describe the results of applying the robust identification and model (in)validation algorithms outlined in sections 1.3 to find a *combined model of the system that includes both the dynamics of the BiSight pan and tilt unit and the computer vision module (image processing algorithms)*[6]. For the sake of briefness an abbreviated description is given of the tilt–axis results, since the procedure is similar to the one used for the pan–axis.

3.1 Identification results

In order to identify the transfer function from the command input u to the pan and tilt units to the displacements y_p and y_t of the target in the image, *measured in pixels*, the system was sequentially excited in each axis with a step input of amplitude 67 *encoder units*[7] (roughly corresponding to an angular displacement of 1.5^o) and the position of a target (originally located at the center of the image) was measured in pixels[8]. This process was repeated for eight different zoom settings ranging from 0 to 70% in increments of 10%, with the goal of obtaining an accurate description of the system for this range of zoom values. The corresponding images of the target for some representative zoom values are shown in Figure 5.

Finally, the *a priori* information was determined by repeatedly measuring the location of the target in the absence of input, to obtain an experimental noise level of approximately $\epsilon_t = 2$ pixels[9], and by measuring the time–constants of the pan and tilt, leading to an estimated value $\rho \equiv 2$. Using this data, the robust identification algorithm outlined in the previous section was run in each case of 0% through 70% zoom levels with 10% increments[10]. Higher level zoom values (above 70%) are not considered since very high level zoom values caused too much blurring in our lab. environment due to distance, the resulting images were useless. For each case, $N_t = 26$ data samples were used which followed by a model reduction step.

Figure 6 compares the step responses of the resulting identified models against the experimental data points (normalized by the input) for different zoom values. Here 'o' denotes an experimental data point used in the identification, while '$*$' denotes additional experimental data, plotted for validation purposes[11]. As shown in

[6] Identifying a single model combining the dynamics of the pan/tilt unit and the computer vision module captures better the interaction among them, while avoiding artificially inflating the order of the resulting model.

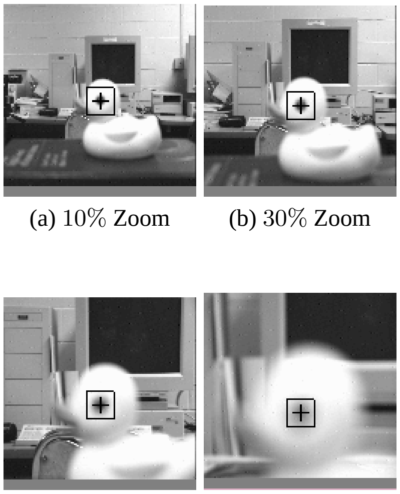
[7] Encoder units are the units used in the servo level PMAC controller commands to drive the servo motors of the pan and tilt unit of BiSight Head–Eye platform.

[8] Note that the input to the system is in the encoder units while the output of the system is in pixels (output of the normalized cross–correlation algorithm).

[9] This experimental error is mainly due to fluctuating conditions such as ambient light.

[10] It is important to note that our proposed method does not require calibration of cameras. Zoom changes were simply obtained through PMAC controller unit. Camera focal length is not measured.

[11] Due to space constraints, only the pan axis is shown, results for the tilt axis are similar.



(a) 10% Zoom (b) 30% Zoom

(c) 50% Zoom (d) 70% Zoom

Fig. 5. Sample Target Images for Different Zoom Values.

the plots, these responses interpolate the experimental data points within the normalized error bounds ($\frac{2pixels}{67encoderunits}$). Similar results, omitted for space reasons, were obtained for the remaining values of the zoom. It is also worth noticing the existence of a time–delay between the time when the input was applied and the system started to react. This delay, due to the image processing, fluctuates between 67 and 96 milliseconds.

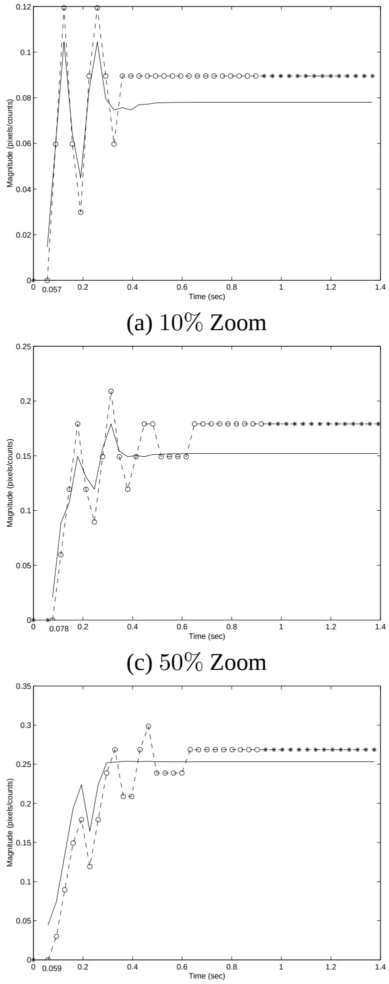
In the sequel we will use $f = 50\%$ as the nominal value of the zoom[12] for robust controller design.

3.2 Uncertainty Description and Model Validation

Figures 7 (a) and (b) show the frequency response corresponding to models identified for different values of f. From these plots it follows that *changes in f result in significant changes in the dynamics*, particularly in the low frequency range. In turn, these changes can lead to loss of stability or significant performance degradation[13]. In order to synthesize a robust controller capable of accommodating these changes in the plant dynamics, in this chapter we will model them as multiplicative dynamic uncertainty. Specifically, the transfer functions $G_{f,pan}(z)$ and $G_{f,tilt}(z)$ corresponding to a chosen nominal value of the f will be modeled as:

[12] As shown in the sequel, the frequency response of the corresponding transfer function is roughly in the center of the family of models.

[13] As we will show in the sequel, a classical PID controller designed for f=50%, goes unstable when f increases to 70%. A PID controller designed for f=70% maintains stability for the entire zoom range, but leads to very slow closed–loop systems for the lower values of f due to the small value of the resulting loop gain.



(a) 10% Zoom

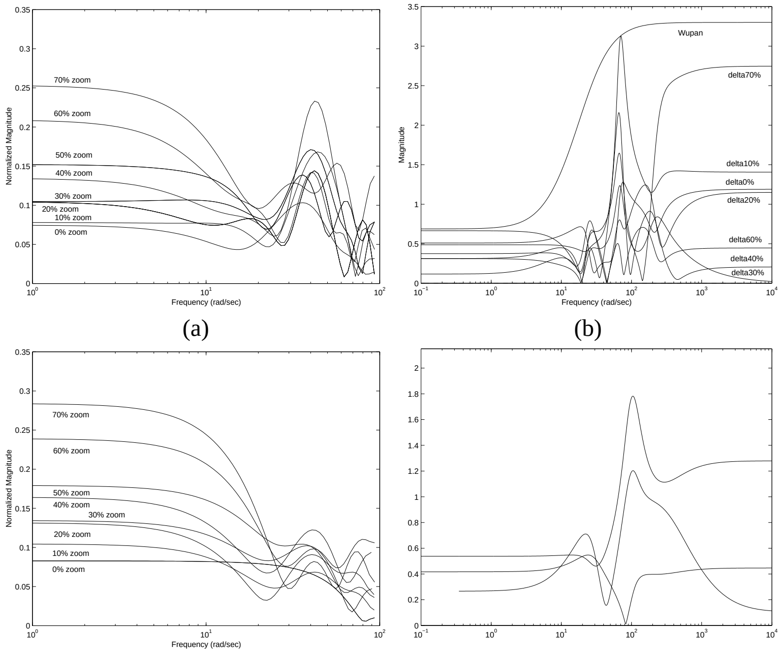
(c) 50% Zoom

(d) 70% Zoom

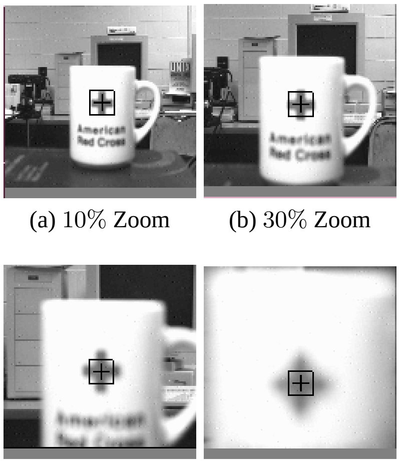
Fig. 6. Model of the Pan Axis (solid line) versus Experimental Data (dashed line) for Different Zoom Values.

$$\begin{aligned} G_{f,pan}(z) &= G_{nom,pan}(z)(1 + \Delta_{pan}(z) W_{u,pan}(z)) \\ G_{f,tilt}(z) &= G_{nom,tilt}(z)(1 + \Delta_{tilt}(z) W_{u,tilt}(z)) \end{aligned} \tag{16}$$

where $G_{nom,pan}$ and $G_{nom,tilt}$, the nominal transfer functions are given in (18), $W_{u,pan}(z), W_{u,tilt}(z)$ are fixed weighting functions containing all the information


Normalized Magnitude
Frequency (rad/sec)
70% zoom
60% zoom
50% zoom
40% zoom
30% zoom
20% zoom
10% zoom
0% zoom
Magnitude
Wupan
delta70%
delta10%
delta0%
delta20%
delta60%
delta40%
delta30%

(a) (b)



(a) 10% Zoom (b) 30% Zoom

(c) 50% Zoom (d) 70% Zoom

Fig. 8. A Different Target Used to Validate the Models.

Model and Uncertainty (In)Validation

In order to validate the models and uncertainty descriptions, the experiments were repeated using a different target, shown in Figure 8. The new data collected was used to construct the model (in)validation LMIs given in equation (5). Table 1 shows the results of minimizing γ subject to feasibility of these LMIs. Since in all cases $\gamma < 1$ and $\epsilon_N \leq \epsilon_t$ this new data does not invalidate the model. Similar results were obtained with other targets.

Table 1. Model Validation Results

	PAN AXIS		TILT AXIS	
Zoom	γ	ϵ_N	γ	ϵ_N
0%	0.4448	0.02985	0.5594	0.02985
10%	0.4383	0.02984	0.4884	0.02985
20%	0.2777	0.02985	0.3331	0.02985
30%	0.2199	0.02984	0.2283	0.02985
40%	0.2626	0.02985	0.1420	0.02985
50%	0.0468	0.02984	0.0000	0.02984
60%	0.2327	0.02985	0.5368	0.02982
70%	0.5921	0.02985	0.5900	0.02971

4 Discrepancies/misfits between theory and experimental results

In this section, we briefly comment on several issues where the existing theory failed to give a complete solution to the identification problem and indicate the tools used to overcome these difficulties.

4.1 Reduced order models

As indicated in section 1, all models consistent with the *a priori* information and *a posteriori* experimental data can be parametrized as a Linear Fractional Transformation (LFT) of a fixed transfer matrix, generated by the identification algorithm, and a free parameter: a transfer matrix $q(z) \in \mathcal{BH}_\infty$. In particular, the central interpolant (corresponding to the choice $q = 0$) leads to a model of order N_t, the number of experimental data points, with a state–space realization given by (3). Thus, in this case the central interpolant has order 26. Since popular robust control methods lead to controllers having at least the same order as the plant, using these models would lead to complex controllers. In principle, this difficulty can be avoided by using the additional degrees of freedom available in q to search for lower order models inside the consistency set. However, the resulting problem is difficult to solve, since the order of the model (or equivalently the rank of the associated Hankel operator) is not a convex function of q. Thus, rather than pursuing this approach, we simply model reduced the central interpolant using balanced truncations. The resulting, reduced order transfer functions (recall that these correspond to the non–delayed portion of the model) are given by:

$$G_{nom} = [G_{nom,pan} \; G_{nom,tilt}]^T$$

$$G_{nom,pan}(z) = \frac{0.0206z^7 + 0.0765z^6 + 0.0551z^5}{1.0000z^7 + 0.412z^6 + 0.3976z^5}$$
$$\frac{+0.0806z^4 + 0.0233z^3 + 0.0174z^2 + 0.045z + 0.045}{+0.1902z^4 + 0.2079z^3 + 0.0939z^2 + 0.0628z + 0.026}$$
$$G_{nom,tilt}(z) = \frac{0.0297z^5 + 0.0552z^4 + 0.0515z^3}{1.0000z^5 - 0.1554z^4 + 0.0375z^3}$$
$$\frac{+0.0142z^2 - 0.0179z + 0.0358}{+0.0419z^2 + 0.0138z + 0.0023} \tag{18}$$

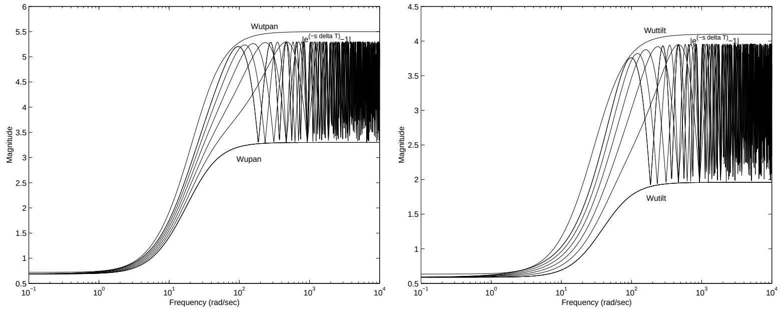
4.2 Existence of variable time delays

Recall that the experimental data shows the existence of a variable time delay that fluctuates between two and three sampling periods[14], depending on the amount of time required to locate the target in the image. This time delay cannot be captured

[14] Sampling frequency of our system is 33Hz.

by existing identification methods (since it cannot be modelled by a finite dimensional transfer function). To circumvent this difficulty, we modelled this variable time delay as a combination of (i) a *fixed*, two sampling periods time delay, and (ii) multiplicative model uncertainty. This approach leads to a model of the form (16) where the weighting functions $W_{u,pan}(z)$ and $W_u, tilt(z)$ are increased to account for the additional terms of the form $|e^{-s\Delta T} - 1|$ arising from the time delays. The final weighting functions, taking into account both model and time–delay uncertainty are given below, with the corresponding Bode plots shown in Figures 9 (a) and (b).

$$W_{ut} = \begin{pmatrix} W_{ut,pan}(z) \\ \\ W_{ut,tilt}(z) \end{pmatrix} = \begin{pmatrix} \frac{3.9315z-3.4584}{z-0.3436} \\ \\ \frac{2.7045z-2.1925}{z-0.1944} \end{pmatrix} \tag{19}$$



(a) (b)

Fig. 9. Model uncertainty including the time delay and overall uncertainty weights. (a) Pan Axis; (b): Tilt Axis.

4.3 Controller Implementation Considerations

Recall that in our setup the pan and tilt unit is controlled by a 10 channel δ–τ PMAC controller board hosted by a Sun UltraSPARC workstation. In order to avoid time delays associated with trajectory pre–planning introduced by the PMAC controller board, the δ–τ PMAC board is driven at the servo level, i.e. by directly accessing its commanded position (CP) registers. A difficulty with this approach is that the relatively low sampling rate (33Hz) can lead to jerky camera motions. This effect was prevented by distributing the set–point changes over the entire sampling period, resulting in smooth motions [60]. Distribution of these changes was accomplished by running a PMAC *plc0* background program that every 4 servo cycles (approximately 1.77 ms) increments the value of the CP registers an amount equal to the desired total set–point change divided by a suitable step size[15], leading to the block diagram shown in Figure 10.

[15] Our specific implementation uses 10 steps.

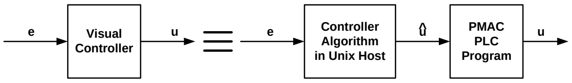


Fig. 10. Controller Implementation to Avoid Excessive Jerk.

It follows that the *plc0* program can be modeled as: $u(n) = u(n-1) + \hat{u}(n)$. Taking the z–transform of both sides yields the following transfer function:

$$H_{plc}(z) = \frac{z}{z-1} \tag{20}$$

Hence, the net effect of the program is to add an *integrator* at the input to the system, thus smoothing out high frequency vibrations and guaranteeing zero steady–state tracking error to step inputs. Combining this effect with the time delay and the transfer functions identified in section 3 leads to the following overall model of the plant:

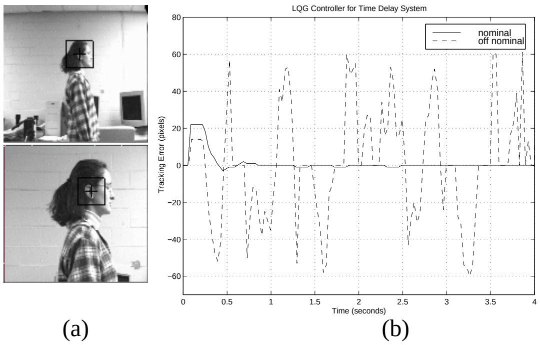
$$\begin{aligned}
\hat{G}_{nom,i} &= \tfrac{1}{z(z-1)} G_{nom,i} \\
G_{nom,pan}(z) &= \tfrac{0.0206z^7+0.0765z^6+0.0551z^5+0.0806z^4+0.0233z^3+0.0174z^2+0.045z+0.045}{1.0000z^7+0.412z^6+0.3976z^5+0.1902z^4+0.2079z^3+0.0939z^2+0.0628z+0.026} \\
G_{nom,tilt}(z) &= \tfrac{0.0297z^5+0.0552z^4+0.0515z^3+0.0142z^2-0.0179z+0.0358}{1.0000z^5-0.1554z^4+0.0375z^3+0.0419z^2+0.0138z+0.0023}
\end{aligned} \tag{21}$$

where $\hat{G}_{nom,i}$ denotes the identified, non–delayed transfer function corresponding to the axis under consideration.

4.4 Limitations of Classical Controllers

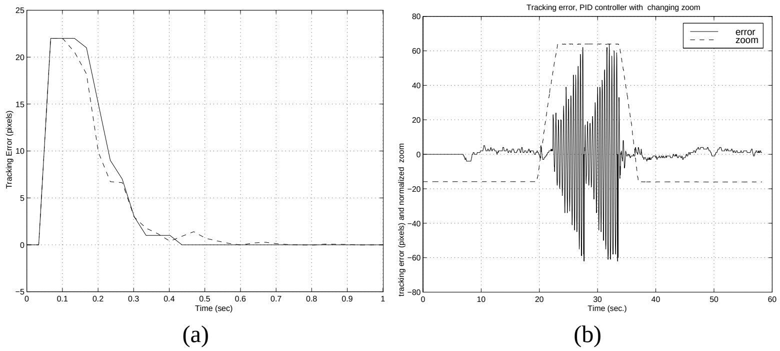
Once a model of the plant has been identified, standard control techniques can be used to synthesize suitable controllers. Note however that when using these techniques, the time delay T_d must be taken into account in order to guarantee stability [70].

As shown in Figure 11(b), an optimal LQG controller designed taking this time delay into account achieves good performance for the *nominal* system. However, these results also illustrate its fragility: a change in the focal length f destabilizes the closed–loop system [31]. A similar situation arises when using a PID controller, experimentally tuned to achieve a compromise between settling time and overshoot. As shown in Figure 12 (a) this controller also achieves good performance for the nominal system. However, this is achieved by tuning the controller to the nominal model G_{nom}, and, as in the LQG case, this leads to potentially *fragile* closed loop systems. This effect is illustrated in Figure 12 (b), showing the result of an experiment where the focal length f of the system is increased while tracking, to zoom–in a feature of interest, and then reduced to its previous value. As shown there, while the system exhibits good performance for the nominal value of f, it becomes unstable as f is changed.



(a) (b)

Fig. 11. Smooth Pursuit: (a) Target for different values of f. (b) Tracking error of an LQG controller (experimental).



(a) (b)

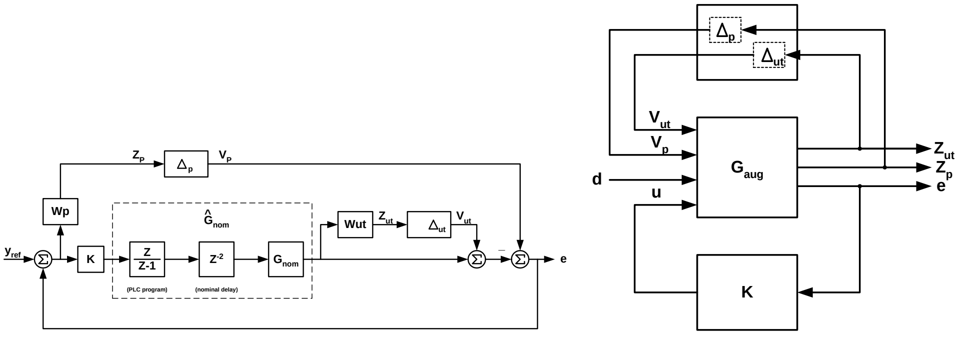
Fig. 12. Pan Axis Tracking Error Using a PID Controllers: (a) Nominal Model, Simulation (solid) and Experimental (dashed). (b) Nominal Versus Off Nominal f, Experimental

The loss of stability can be easily explained from the experimental data shown in Figure 16, showing that (at the low frequency range) f enters the model virtually as a static gain multiplying the control action. Thus, roughly speaking, the system becomes unstable when this “gain” exceeds the gain margin.

In principle stability can be guaranteed by designing a PID controller for the maximum, rather than medium value of f, hence guaranteeing that the gain margin is never exceeded. However, this leads to very slow closed–loop systems for the medium and minimum values of f due to the small value of the gain that is required. As we show next, to a large extent this performance loss can be avoided by using the robust control tools described in section 1 to synthesize controllers that guarantee a given performance level for all possible values of the parameters.

4.5 Robust Controller Design and Performance Weight Selection

In this section we describe how to combine the nominal model (21) and the associated uncertainty description (19) with μ–synthesis in order to obtain a controller that achieves robust performance. The first step toward accomplishing this is to recast the plant into a suitable form and to select weighting functions that encapsulate the control objectives.



(a) (b)

Fig. 13. (a) Block Diagram of the Augmented Plant for Each Axis. (b) Block Diagram with the Uncertainty “Pulled Out” of the Loop.

Figure 13 (a) shows a block diagram of the active vision system taking into account the uncertainty. As discussed in section 1.3, the additional block Δ_p represents a fictitious perturbation block, which in conjunction with the weighting function W_p allows for imposing performance specifications on the tracking error $e(t)$. In particular, in our case W_p should be selected to achieve the following specifications (for all values of $0 \leq f \leq 70\%$):

1. Settling time to step inputs on the order of 1 second.
2. Fast rise time, with minimal overshoot, to avoid correlator walk-off problems.
3. Small control action to avoid actuator saturation.
4. Zero steady state error tracking to step inputs[16].

Unfortunately there is no explicit way to exactly map these time domain performance specifications to frequency domain weighting functions. Rather, there exist some well established practical rules. Specifically, suppose that it is desired to achieve the following performance specifications:

- Steady–state error $\leq$ A
- Closed–loop bandwidth $\geq W_B$

[16] This specification is automatically satisfied as long as $W_p(1) \neq 0$, due to the integral action introduced by the plc program.

- High frequency noise amplification $\leq$ P

Then the magnitude Bode plot of the sensitivity function should resemble the plot shown in Figure 14 (a), which in turn leads to the performance weighting function shown in Figure 14 (b), [47].

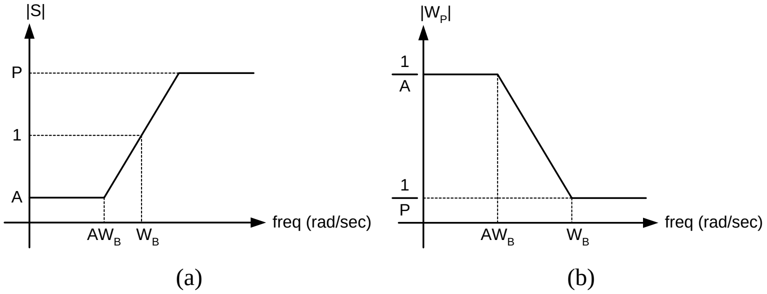


Fig. 14. (a)Desired Sensitivity Function. (b) Desired Performance Weight Function.

$$W_p \cong \frac{1}{P}(\frac{\tau_{cl}s+P}{\tau_{cl}s+A}) \qquad \text{where} \quad \tau_{cl}=\frac{1}{W_B}$$

In addition, commonly used rules of thumn are to

- Penalize low frequencies for good steady–state and settling time.
- Penalize high frequencies for good transient response.

Taken into account these considerations, a good compromise between the performance specifications was accomplished by using the following (experimentally tuned) function:

$$W_p(z)=\frac{0.2604z+0.1412}{z+0.2550} \tag{22}$$

As shown in the Bode plot, Figure 15, this function has a low–pass characteristic, penalizing the low frequency components and leading to a closed–loop bandwidth larger than 30Hz, in order to achieve the settling time specifications.

Figure 13 (b) shows a block diagram of the plant in the standard form used for μ–synthesis. Here the augmented plant G_{aug} has the following realization:

$$\begin{bmatrix} Z_{ut} \\ Z_p \\ e \end{bmatrix} = \begin{bmatrix} 0 & 0 & W_{ut,i}\hat{G}_{nom,i} \\ -W_p & W_p & -W_p\hat{G}_{nom,i} \\ -1 & 1 & -\hat{G}_{nom,i} \end{bmatrix} \begin{bmatrix} V_{ut} \\ V_p \\ u \end{bmatrix} \tag{23}$$

where $\hat{G}_{nom,i}$, the transfer function of the axis under consideration, and $W_{ut,i}$, the associated uncertainty description, are given in (21) and (19), respectively. Using μ–synthesis with first order scales, followed by a model reduction step leads to the following controllers:

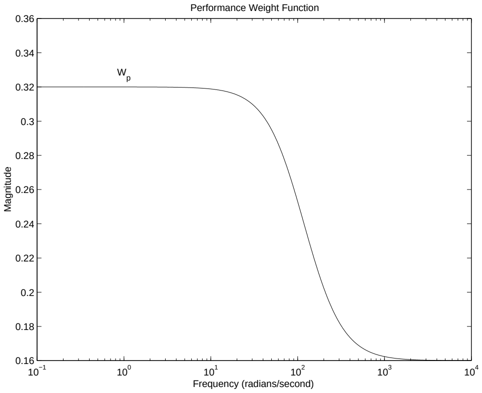


Fig. 15. Performance Weight Function:

$$K_{\mu_{pan}}(z) = \frac{1.4181z^6 - 0.7553z^5 + 0.5886z^4}{z^6 + 0.1265z^5 + 0.2633z^4}$$
$$\frac{-0.0262z^3 + 0.0439z^2 + 0.0519z + 0.0143}{+0.3171z^3 + 0.2723z^2 + 0.1087z + 0.3008} \quad (24)$$

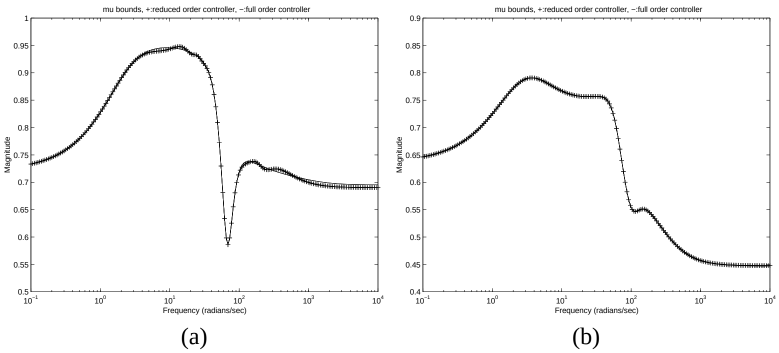
$$K_{\mu_{tilt}}(z) = \frac{1.7939z^6 - 0.5281z^5 + 0.1520z^4}{z^6 + 0.6059z^5 + 0.4315z^4}$$
$$\frac{+0.0588z^3 + 0.0112z^2 + 0.0117z - 0.0052}{+0.2693z^3 + 0.1625z^2 + 0.1446z + 0.1123} \quad (25)$$

The values of μ corresponding to the full (solid line) and reduced order $(+)$ controllers for the pan and tilt axes are shown in Figures 16 (a) and (b) respectively. Since in all cases we have that $\mu < 1$, it follows that these controllers achieve robust performance.

4.6 Experimental Validation

In order to validate the controllers, we performed the following sets of experiments:

- **Step Response:** The step responses (simulation and experimental) of the closed loop system corresponding to several values of f are shown in Figure 17. As illustrated there, as opposed to both the LQG and PID controllers, these controllers achieves tracking throughout the entire range, without the need to sacrifice nominal performance or to de–tuning in order to guarantee stability. Similar results, omitted for space reasons, were obtained with other targets. Note in passing that there is very good agreement between the predictions of the linear simulation based on the models (18) and the actual experimental data.



(a) (b)

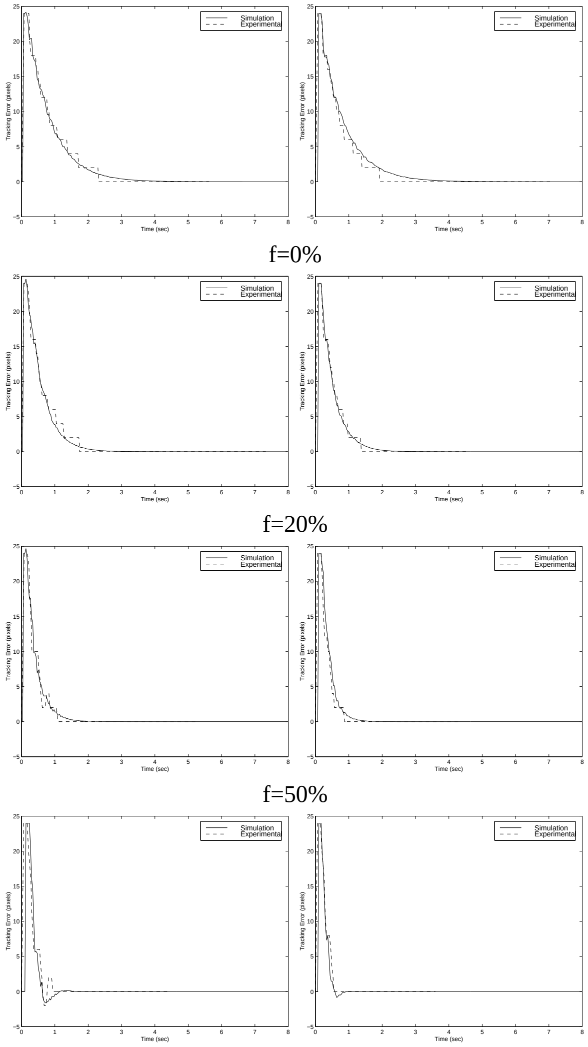
Fig. 16. (a) μ Bounds for Full (solid line) and Reduced Order (+) Controllers. (a) Pan Axis. (b) Tilt Axis.

- **Smooth Pursuit:** Figures 19 and 20 show the results of experiments where the person shown in Figure 18 is tracked while zooming in and out of her features. As before, the robust controllers were able to achieve good tracking performance in spite of the substantial change in the dynamics of the plant due to the change in f, the uncertain time–delay, and the presence of clutter.

5 Conclusions

Recent hardware developments have opened up the possibility of applying active vision techniques to a broad range of problems, such as Intelligent Vehicle Highway Systems, robotic–assisted surgery, 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 errors, stemming for instance from poorly calibrated cameras, blurring or only partially determined feature correspondences between images. Arguably, at this time one of the critical factors limiting widespread use of active vision techniques is the fragility of the resulting systems. This lack of robustness can be traced, to a large extend, to substantial variations in the dynamics of the plant due to variations in its parameters, and to the existence of unmodelled dynamics.

In this chapter we illustrated how recently developed robust identification and model (in)validation techniques can be brought to bear on the problem. Rather than producing a single model of the plant, these methods generate a nominal model as well as an uncertainty description guaranteed to cover both the effects of unmodelled dynamics, modelling errors and the variations due to changes in various parameters such as camera zoom values. As shown in section 4.5, this description can be directly combined with robust control techniques to synthesize controllers that guarantee acceptable performance for a large range of conditions. A salient feature of the



f=0%

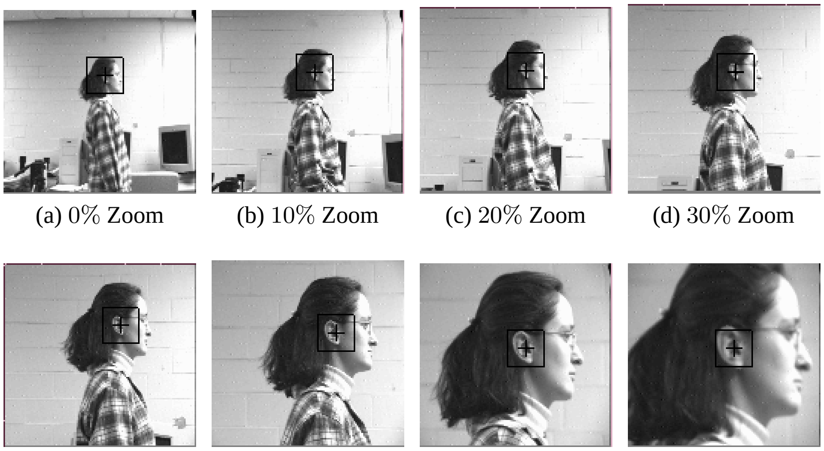
f=20%

f=50%

f=70%

(a) (b)

Fig. 17. Tracking error for different zoom values, step response experiments: (a) pan axis, (b) tilt axis. (Solid: simulation, Dashed: experimental)



(a) 0% Zoom (b) 10% Zoom (c) 20% Zoom (d) 30% Zoom

(d) 40% Zoom (e) 50% Zoom (f) 60% Zoom (g) 70% Zoom

Fig. 18. Sample Target Images for Zoom Values 0% through 70%.

proposed approach is that it requires very few assumptions on the system. For instance neither the order nor the structure of the model needs to be fixed *a priori* , and it produces *a complete model that includes both the dynamics of the head and the computer vision module without the need to separate these components (and thus potentially missing some of their coupling).* Finally, it leads to simple, computationally tractable LMI optimization problems.

The methodology was experimentally validated using an uncalibrated setup to successfully track an uncooperative target in a cluttered environment.

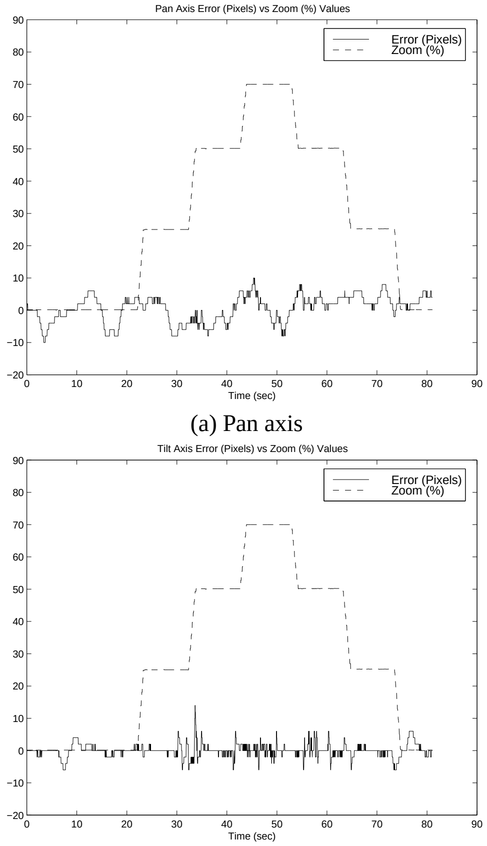
It is worth noticing that, for each axis, a single controller is able to yield good tracking performance for a wide range of conditions. Naturally, this is achieved at the cost of mild degradation of nominal performance. In addition, the systems corresponding to lower values of f (and hence lower gain) exhibit slower responses than those corresponding to higher values. The results [72] indicate that these effects can be addressed by considering a family of parameter varying robust controllers, indexed by the parameter f. Research in this direction is currently being explored.

Acknowledgements

Support from NSF under grants ECS–0221562, IIS–0117387, and ITR–0312558 and AFOSR under grant FA9550–05–1–0437 is gratefully acknowledged.

References

[1] Akella, M. R. (2005). Vision-based adaptive tracking control of uncertain robot manipulators. *IEEE Trans. on Robotics* **21**(4), 747–753.



(a) Pan axis

(b) Tilt axis

Fig. 19. Target is moving and zoom is being changed from 0% to 70% in 3 steps.

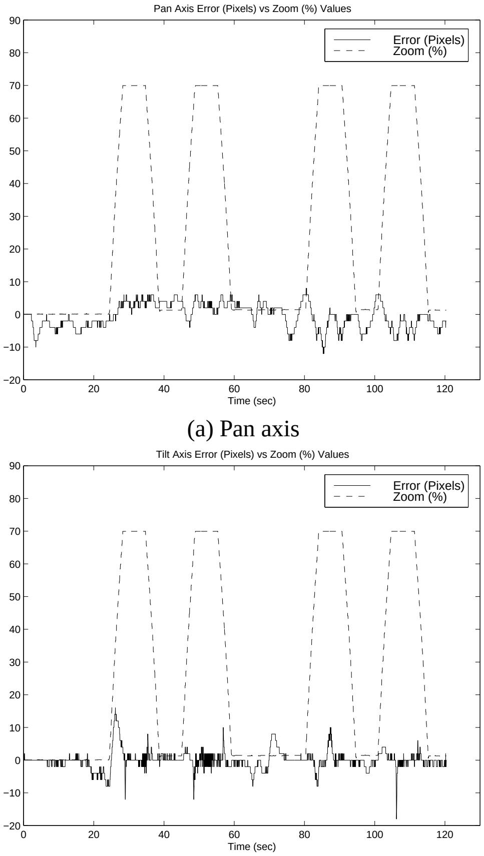
[2] Balas, G., Doyle, J. C., Glover, K., Packard, A., and Smith, R. (1991). *μ-Analysis and Synthesis Toolbox*. The MathWorks Inc.

[3] Black, M. J. and Jepson, A. D. (1998). Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *International Journal of Computer Vision* **26**(1), 63–84.

[4] Blake, A. and Isard, M. (1998). *Active Contours*. Springer Verlag, Berlin.

[5] Broggi, A., Cellario, M., Lombardi, P., and Porta, M. (2003). An evolutionary approach to visual sensing for vehicle navigation. *IEEE Trans. on Industrial Electronics* **50**(1), 18–29.

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



(a) Pan axis

(b) Tilt axis

Fig. 20. Target is moving and zoom is being changed from 0% to 70% in 1 step.

[7] Calabi, E., Olver, P. J., Shakiban, C., Tannenbaum, A., and Haker, S. (1998). Differential and numerically invariant signature curves applied to object recognition. *International Journal of Computer Vision* **26**(2), 107–135.

[8] Chaumette, F. and Hutchinson, S. (2006). Visual servo control part i: Basic approaches. *IEEE Robotics and Automation Magazine* **13**(4), 82–90.

[9] Chaumette, F. and Hutchinson, S. (2006). Visual servo control part ii: Advanced approaches. To Appear.

[10] Chen, J., Dawson, D. M., Dixon, W. E., and Behal, A. (2005). Adaptive homography-based visual servo tracking for a fixed camera configuration with a camera-in-hand extension. *IEEE Trans. on Control Systems Technology* **13**(5), 814–825.

[11] Chen, J. and Gu, G. (2000). *Control Oriented System Identification, An $\mathcal{H}_\infty$ Approach*. John Wiley, New York.

[12] Chen, J. and Nett, C. (1995). The Carathéodory-Fejér problem and $\mathcal{H}_\infty/\ell^1$ identification. *IEEE Transactions on Automatic Control* **40**(4), 729–735.

[13] Chen, J. and Wang, S. (1996). Validation of linear fractional uncertain models: Solutions via matrix inequalities. *IEEE Transactions on Automatic Control* **41**(6), 844–849.

[14] Chesi, G. and Hashimoto, K. (2002). A self-calibrating technique for visual servoing. *IEEE Int. Conf. on Intelligent Robots and Systems*, pp. 2681–2686. Las Vegas, Nevada.

[15] Clark, J. J. and Ferrier, N. J. (1988). Modal control of an attentive vision system. *International Conference on Computer Vision*, pp. 514–523.

[16] Coombs, D. and Brown, C. (1993). Real-time binocular smooth pursuit. *International Journal of Computer Vision* **11**(2), 147–164.

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

[18] Doyle, J. (1982). Analysis of feedback systems with structured uncertainties. *IEE Proceedings, Part D* **129**, 242–250.

[19] Espiau, B., Chaumette, F., and Rives, P. (1992). A new approach to visual servoing in robotics. *IEEE Trans. on Robotics and Automation* **8**(3), 313–326.

[20] Fang, Y., Dixon, W. E., Dawson, D. M., and Chawda, P. (2005). Homography-based visual servo regulation of mobile robots. *IEEE Trans. on Systems, Man and Cybernetics, Part B* **35**(5), 1041–1050.

[21] Feddema, J. (1997). Microassembly of micro–electromechanical systems (MEMS) using visual servoing. *Block Island Workshop on Vision and Control*, pp. 257–272.

[22] Ferreira, A., Cassier, C., and Hirai, S. (2004). Automatic microassembly system assisted by vision servoing and virtual reality. *IEEE Trans. on Mechatronics* **9**(2), 321–333.

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

[24] Finnefrock, M., Jiang, X., and Motai, Y. (2005). Visual-based assistance for electric vehicle driving. *IEEE Intelligent Vehicles Symposium*, pp. 656–661.

[25] Gans, N. R. and Hutchinson, S. A. (2003). An experimental study of hybrid switched system approaches to, visual servoing. *IEEE Int. Conf. on Robotics and Automation*, pp. 3061–3068. Taipei, Taiwan.

[26] Gu, L. and Su, J. (2006). Gaze control on humanoid robot head. *IEEE World Congress on Intelligent Control and Automation*, pp. 9144–9148. Dalian, China.

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

[28] Hager, G. and Belhumeur, P. (1997). Efficient region tracking with parametric models of geometry and illumination. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **20**(10), 1025–1039.

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

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

[31] Hayman, E., Reid, I., and Murray, D. W. (1996). Zooming while tracking using affine transfer. *British Machine Vision Conference*, pp. 395–404.

[32] Hill, J. and Park, W. T. (1979). Real time control of a robot with a mobile camera. *9th ISIR*, pp. 233–246. Washington D.C.

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

[34] Hynes, P., Dodds, G. I., and Wilkinson, A. J. (2005). Uncalibrated visual-servoing of a dual-arm robot for surgical tasks. *IEEE Int. Symposium on Computational Intelligence in Robotics and Automation*, pp. 151–156. Espoo, Finland.

[35] Inanc, T. (2002). *A Novel Approach to Active Vision Systems: Modelling, Control and Real Time Tracking*. Ph.D. thesis, The Pennsylvania State University.

[36] Jain, R., Kasturi, R., and Schunck, B. G. (1995). *Machine Vision*. Mac Graw Hill.

[37] Jensfelt, P., Wijk, O., Austin, D. J., and Andersson, M. (2000). Experiments on augmenting condensation for mobile robot localization. *IEEE International Conference on Robotics and Automation*, pp. 2518–2524.

[38] Krupa, A., *et al.* (2003). Autonomous 3–d positioning of surgical instruments in robotized laparoscopic surgery using visual servoing. *IEEE Trans. on Robotics and Automation* **19**(5), 842–853.

[39] Ljung, L. (1987). *System Identification: Theory for the User*. Prentice-Hall, Englewood Cliffs, N.J.

[40] Malis, E. (2004). Visual servoing invariant to changes in camera-intrinsic parameters. *IEEE Trans. on Robotics and Automation* **20**(1), 72–81.

[41] Martens, C., Ruchel, N., Lang, O., Ivlev, O., and Graser, A. (2001). A friend for assisting handicapped people. *IEEE Robotics and Automation Magazine* **8**(1), 57–65.

[42] Maybank, S. and Faugeras, O. (1992). A theory of self-calibration of a moving camera. *Int. J. Comput. Vis.* **8**(2), 123–151.

[43] Megretski, A. (1993). On the gap between structured singular values and their upper bounds. *Proc. 32*$^{\mathrm{nd}}$ *IEEE CDC*, pp. 3461–3462. San Antonio, TX.

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

[45] Nageotte, F., Zanne, P., Doignon, C., and de Mathelin, M. (2006). Visual servoing-based endoscopic path following for robot-assisted laparoscopic surgery. *IEEE Int. Conf. on Intelligent Robots and Systems*, pp. 2364–2369. Beijing, China.

[46] North, B., Blake, A., Isard, M., and Rittscher, J. (2000). Learning and classification of complex dynamics. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **22**(9), 1016–1034.

[47] P. Lundstrom, S. S. and Wang., Z. (1991). Performance weight selection for h_∞ and μ–control methods. *IEEE Trans. of the Institute of Measurements and Control* **13**(5), 241–252.

[48] P. Lundstrom, S. S. and Wang., Z. (1991). Uncertainty weight selection for h_∞ and μ–control methods. *IEEE Int. Conf. on Decision and Control*, pp. 1537–1542.

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

[50] Papanikolopoulos, N. P. (1995). Integrating computer vision and control for vision assisted robotics tasks. *1995 American Control Conference*, pp. 904–908. Seattle, WA.

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

[52] Parrilo, P. A., Sznaier, M., and Pena, R. S. S. (1998). Mixed time/frequency domain based robust identification. *Automatica* **34**(11), 1375–1389.

[53] Pena, R. S. S. and Sznaier, M. (1998). *Robust Systems Theory and Applications*. John Wiley, New Jersey.

[54] Pollefeys, M., Koch, R., and Gool, L. V. (1999). Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. *Int. J. Comput. Vis.* **32**(1), 7–25.

[55] Ralis, S., Vikramaditya, B., and Nelson, B. (2000). Micropositioning of a weakly calibrated microassembly system using coarse-to-fine visual servoing strategies. *IEEE Trans. on Electronics Packaging Manufacturing* **23**(2), 123–131.

[56] Reid, I. D. and Murray, W. (1996). Active tracking of foveated feature clusters using affine structure. *International Journal of Computer Vision* **18**(1), 41–60.

[57] Rotstein, H. and Rivlin, E. (1996). Optimal servoing for active foveated vision. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 177–182.

[58] Safonov, M. (1981). Stability margins of diagonally perturbed multivariable feedback systems. *IEEE Conf. on Decision and Control*, pp. 251–256.

[59] Safonov, M. and Chiang, R. (1988). *Robust Control Toolbox*. The MathWorks Inc.

[60] Seelen, U. M. C. V. (1997). *Performance Evaluation of An Active Vision System*. University of Pennsylvania.

[61] Shakernia, O., Ma, Y., Koo, T. J., Hespanha, J., and Sastry, S. S. (1999). Vision guided landing of an unmanned air vehicle. *Proc. 1999 IEEE CDC*, pp. 4143–4148.

[62] Sharkey, P. M., Murray, D. W., Vandevelde, S., Reid, I. D., and McLauchlan, P. F. (1993). A modular head/eye platform for real–time reactive vision. *Mechatronics* **3**(4), 517–535.

[63] Skogestad, S., Morari, M., and Doyle, J. (1988). Robust control of ill-conditioned plants: High-purity distillation. *IEEE Trans. Autom. Contr.* **33**(12), 1092–1105.

[64] Smith, C. E., Brandt, S. A., and Papanikolopoulos, N. P. (1997). Eye in hand robotic tasks in uncalibrated environments. *IEEE Transactions on Robotics and Automation* **13**(6), 903–914.

[65] Smith, C. E., Richards, C. A., Brandt, S. A., and Papanikolopoulos, N. P. (1996). Visual tracking for intelligent vehicle–highway systems. *IEEE Trans. Vehicular Tech.* **45**(4), 744–759.

[66] Song, W., Kim, J., and Bien, Z. (2000). Visual servoing for human-robot interaction in the wheelchair-based rehabilitation robot. *IEEE Int. Conf. on Systems, Man, and Cybernetics*, pp. 1811–1816.

[67] Song, Y., Li, M., Sun, L., , and Ji, J. (2005). Global visual servoing of miniature mobile robot inside a micro-assembly station. *IEEE Int. Conf. on Mechatronics and Automation*, pp. 1586–1591. Niagara Falls, Canada.

[68] Starner, T. and Pentland, A. (1998). Real-time american sign language recognition using desk and wearable computer based video. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **20**(12), 1371–1375.

[69] Sturm, P. (2002). Critical motion sequences for the self-calibration of cameras and stereo systems with variable focal length. *Image Vis. Comput.* **20**(5–6), 415–426.

[70] Sznaier, M. and Camps, O. (1998). Control issues in active vision: Open problems and some answers. *1998 IEEE Conf. Dec. Control*, pp. 3238–3244.

[71] Sznaier, M., Inanc, T., and Camps, O. (2000). Robust controller design for active vision systems. *2000 American Control Conference*, pp. 2013–2017.

[72] Sznaier, M., Murphy, B., and Camps, O. (2000). An LPV approach to synthesizing robust active vision systems. *2000 IEEE Conf. Dec. Control*, pp. 2545–2550.

[73] Taylor, C., Ostrowski, J., and Jung, S. H. (1999). Robust visual servoing based on relative orientation. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 574–580.

[74] Taylor, C. J., Kosecka, J., Blasi, R., and Malik, J. (1999). Comparative study of vision-based lateral control strategies for autonomous highway driving. *International Journal of Robotics Research* **18**(5), 442–453.

[75] Thorhallsson, T. and Murray, D. W. (1999). Tensors of three affine views. *IEEE Computer Vision and Pattern Recognition*, pp. 450–456.

[76] Tsotsos, J. K., *et al.* (1998). PLAYBOT: A visually-guided robot for physically disabled children. *Image and Vision Computing* **16**(4), 275–292.

[77] Vitrani, M., Morel, G., and Ortmaier, T. (2005). Automatic guidance of a surgical instrument with ultrasound based visual servoing. *IEEE Int. Conf. on Robotics and Automation*, pp. 508–513. Barcelona, Spain.

[78] Wang, Y. F., Uecker, D. R., and Wang, Y. (1996). Choreographed scope maneuvering in robotically–assisted laparoscopy with active vision guidance. *3*$^{\mathrm{rd}}$ *IEEE Workshop on Applications of Computer Vision*, p. 187. Sarasota.

[79] Xie, H., Chen, L., Sun, L., and Rong, W. (2005). Hybrid vision-force control for automatic assembly of miniaturized gear system. *IEEE Int. Conf. on Robotics and Automation*, pp. 1368–1373. Barcelona, Spain.

[80] Zergeroglu, E., Dawson, D. M., de Queiroz, M. S., and Behal, A. (2001). Vision-based nonlinear tracking controllers with uncertain robot-camera parameters. *IEEE Trans. on Mechatronics* **6**(3), 322–337.