

N90-23024

Modern CACSD using the Robust-Control Toolbox

Richard Y. Chiang and Michael G. Safonov
Department of Electrical Engineering - Systems
University of Southern California
Los Angeles, CA. 90089-0781

Abstract

The *Robust-Control Toolbox* [1] is a collection of 40 "M-files" which extend the capability of PC/PRO-MATLAB to do modern multivariable robust control system design. Included are robust analysis tools like singular values and structured singular values, robust synthesis tools like continuous/discrete H^2/H^∞ synthesis and LQG Loop Transfer Recovery methods and a variety of robust model reduction tools such as Hankel approximation, balanced truncation and balanced stochastic truncation, etc.

In this paper, we will describe the capabilities of our toolbox and illustrate them with examples to show how easily they can be used in practice. Examples include structured singular value analysis, H^∞ loop-shaping and large space structure model reduction.

1 Introduction

The fundamental issue in robust control theory – *to find a stabilizing controller that achieves feedback performance despite the plant uncertainty*, is still the same issue addressed by the classical 1930's feedback theory of Black, Bode and Nyquist (ref. Fig. 1.1). Modern robust control theory has resolved many of the issues concerning the "gap" between the theory and practice that had grown to troublesome proportions in the 1970's. One key to bridging the "gap" has been the singular value Bode plot. Recent progress in Structured Singular Value (SSV), H^∞ optimal control theory and the model reduction techniques utilizing singular values have made the modern robust control theory *highly practical*.

The inavailability of quality software implementing the techniques of robust control theory has, until very recently, significantly limited the access of both researchers and engineering practitioners to these techniques.

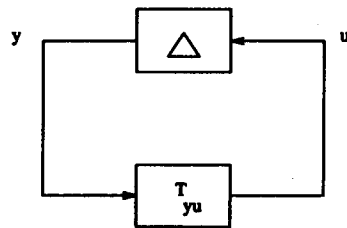
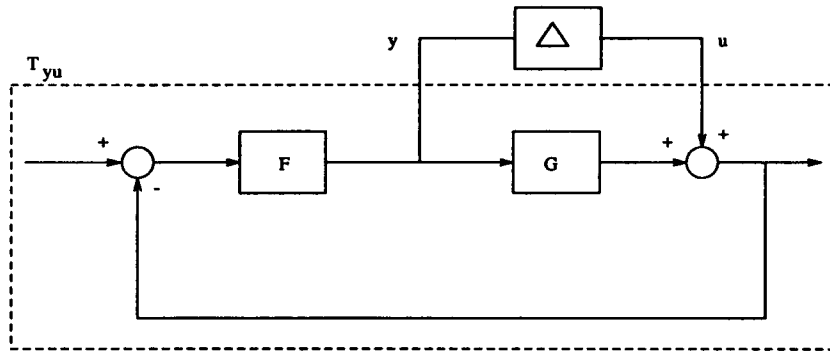


Fig. 1.1 Robust Control Problem.

Our *Robust-Control Toolbox* implements the most up-to-date robust control theory like Perron SSV, optimal descriptor 2-Riccati H^∞ formulae, LQG/LTR and singular value based model reduction techniques, ..., etc. The toolbox consists of a library of 40 functions which extend the capabilities of *PC/PRO - MATLABTM* and the *PC/PRO - MATLAB Control Toolbox*. The toolbox itself represents four man years research work done at USC.

These 40-functions can be catalogued into 3 major areas:

- **Robust Analysis**

- Singular Values
- Characteristic Gain Loci
- Structured Singular Values

- **Robust Synthesis**

- LQG/LTR, Frequency-Weighted LQG
- H^2 , H^∞

- **Robust Model Reduction**

- Optimal Descriptor Hankel (with Additive Error Bound)
- Schur Balanced Truncation (with Additive Error Bound)

In this paper, we will highlight the most important functions in the toolbox, demonstrate how easily they can be used and show what kind of results can be achieved with practical problems. As for the details of the modern robust control theory, they can be found in [7], [2] and the references therein.

2 Robust Analysis

The objective of robust analysis is to find a proper measure of the multivariable stability margin (MSM) against uncertainty. Uncertainty may take many forms, but among the most significant ones are noise/disturbance signals, transfer function modeling errors and unmodeled nonlinear dynamics, etc. Uncertainty in any form is no doubt the major issue in most control system designs.

Several tools to measure MSM are available: [1], [8]

- Singular values (Safonov, 1977; Doyle, 1978)
- Perron eigenvalues (Safonov, 1982)
- Diagonal scaling via nonlinear programming (Doyle, 1982; Tekawy et al., 1989)

Let's define the MSM first:

Definition 1 *Multivariable Stability Margin (MSM)* $K_m, \mu(\cdot)^{-1}$

$$K_m(T_{yu}) \triangleq \mu(T_{yu})^{-1} = \inf_{\Delta} \{ \bar{\sigma}(\Delta) | \det(I - T_{yu}\Delta) = 0 \}.$$

In other words, it's the smallest $\bar{\sigma}(\Delta)$ that can make the determinant $(I - T_{yu}\Delta)$ singular (or the closed-loop system unstable). See Fig. 1.1.

A theorem summarizes the whole MSM idea:

Theorem 1 *The system is stable for all stable Δ_i with $\|\Delta_i\|_{\infty} < 1$, if the MSM $K_m(T_{yu}) > 1$.*

Unfortunately, exact computation of K_m (or μ^{-1}) would require solution of a non-convex optimization problem and is therefore impractical. Fortunately, computable upper bounds on K_m are available, viz.,

$$K_m^{-1}(T_{yu}) = \mu(T_{yu}) \leq \inf_{D \in \mathcal{D}} \|DT_{yu}D^{-1}\|_{\infty} \leq \inf_{D \in \mathcal{D}} \|Dabs(T_{yu})D^{-1}\|_{\infty} \ll \|T_{yu}\|_{\infty}$$

where $\mathcal{D} := \{diag(d_1I, \dots, d_nI) | d_i > 0\}$.

Then using these upper bounds (some may be more conservative than others), one can assure that the system is stable against the norm-bounded uncertainty $\|\Delta\|_{\infty} \leq K_m$.

A comparison of the available upper bounds reveals that some are much easier to compute than others. See table 2.1.

Table 2.1

Method	Property	Computation	Reference
Optimal Diagonal Scaling	$n = 3$, exact K_m $n > 3$, \exists 15 % gap	demanding	Doyle, 1982 Tekawy et al., 1989
Perron Eigenvector Diagonal Scaling	very close to optimal diagonal scaling	easy	Safonov, 1982
Singular Value	can be very conservative	easy	Safonov, 1977 Doyle, 1978

Let's see the following example.

Example: Given a system $G(s)$ with multiplicative uncertainty at its input. Find the MSM.

$$G(s) = \begin{bmatrix} \frac{4}{s+4} & 0 \\ \frac{4s}{s+4} + \frac{8s}{s+8} & \frac{8}{s+8} \end{bmatrix}$$

Theorem 1 implies $\|\Delta\|_\infty \leq (\|G(I+G)^{-1}\|_\infty)^{-1}$.

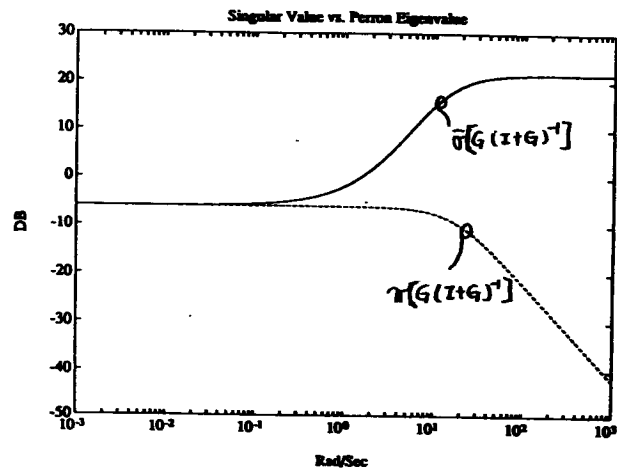
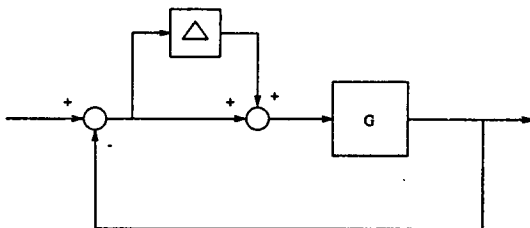


Fig. 2.2 Singular Value vs. K_m (upper bound).

This example reveals that the singular value upper bound is too conservative in "robust analysis". Whereas, Perron SSV is much simpler to compute than diagonally scaled nonlinear programming μ .

3 Robust Synthesis

Classical control system designers often do "loop-shaping" to meet design specifications. So do modern robust control system designers. "Loop-shaping" for mul-

tivariable systems is done via the singular-value Bode plot. However, to shape the loop transfer function $L(s)$ is nothing but to shape the sensitivity function $S(s) = (I + L(s))^{-1}$ and the complementary sensitivity function $T(s) = L(s)(I + L(s))^{-1}$. See Fig. 3.1

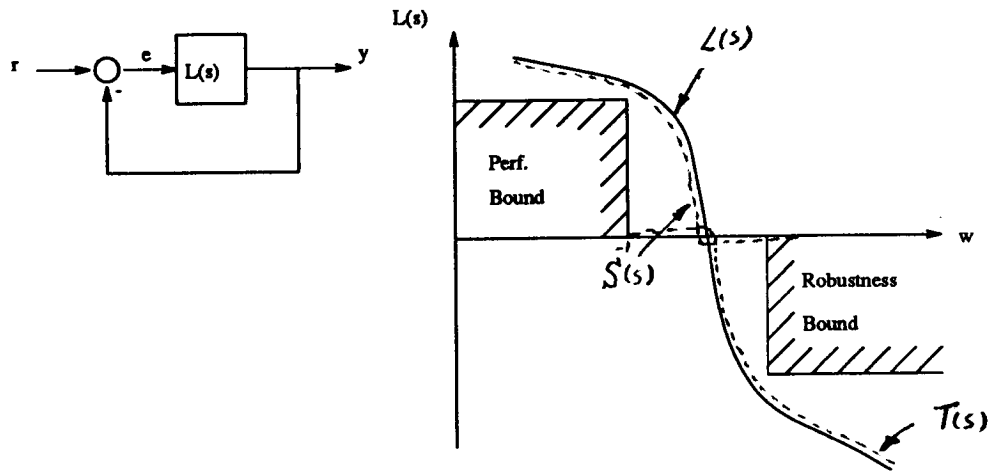


Fig. 3.1 SISO & MIMO Loop-Shaping.

There are several loop-shaping methods available in the *Robust-Control Toolbox* (see Table 3.1), but H^∞ is one of our favorites.

Table 3.1

Methods	Advantages	Disadvantages
LQR (lqr.m)	<ul style="list-style-type: none"> • guaranteed stability margin • pure gain controller 	<ul style="list-style-type: none"> ◦ need full-state feedback ◦ need accurate model ◦ possibly many iterations
LQG (lqg.m)	<ul style="list-style-type: none"> • use available noise data 	<ul style="list-style-type: none"> ◦ no stability margin guaranteed ◦ need accurate model ◦ possibly many iterations
LQG/LTR (ltr.m,ltry.m)	<ul style="list-style-type: none"> • guaranteed stability margin • systematic design procedure 	<ul style="list-style-type: none"> ◦ high gain controller ◦ possibly many iterations ◦ design focus on one point
H^2 (h2lqg.m)	<ul style="list-style-type: none"> • address stability and sensitivity • almost exact loop shaping • closed-loop always stable 	<ul style="list-style-type: none"> ◦ possibly many iterations
H^∞ (hinf.m)	<ul style="list-style-type: none"> • address stability and sensitivity • exact loop shaping • direct one-step procedure 	

Example: Classical loop shaping vs. H^∞ for 2nd order low-damped system.

Given a plant $G(s)$ which is 2nd order with damping 0.05 at 20 rad/sec, find a controller to meet frequency response Bode plot (see Fig. 3.2)

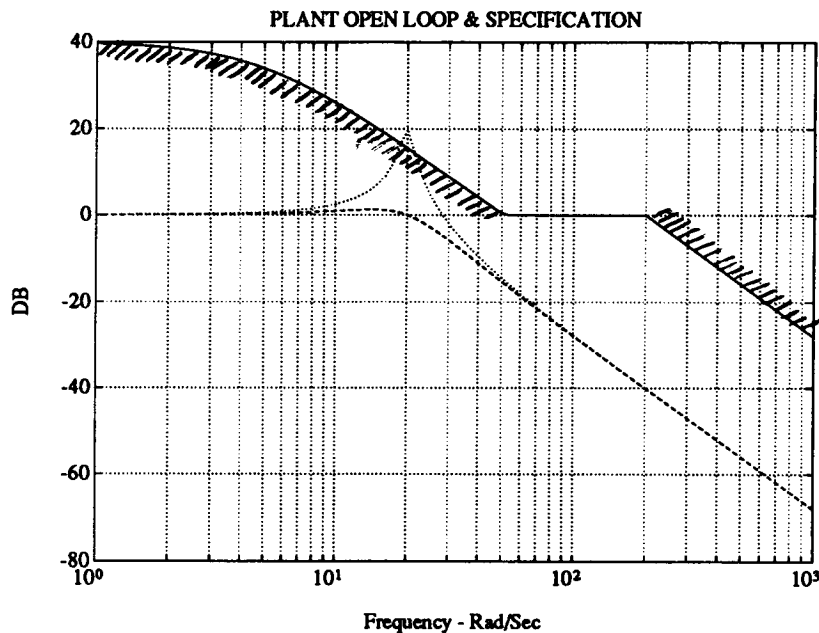


Fig. 3.2 2nd order plant open loop ($\zeta : 0.05, 0.5$) and the $L(s)$ spec.

A classical design might be decomposed into the following: (see Fig. 3.3)

Step 1: Rate feedback to improve damping.

Step 2: Design high frequency (phase margin, BW, roll-off..).

Step 3: Design low frequency (DC gain, disturbance rejection..).

The classical result is shown in Fig. 3.4. Now, let's see how H^∞ approaches the problem.

H^∞ Problem Formulation

We are solving the so-called H^∞ Small-Gain Problem ([3]) using the numerically robust "optimal" descriptor 2-Riccati formulae of Safonov, Limebeer and Chiang [4] [5].

H^∞ Small-Gain Problem:

Given a plant $P(s)$ (ref. Fig. 3.5), find a stabilizing controller $F(s)$ such that the closed-loop transfer function $T_{y_1 u_1}$ is internally stable and its infinity-norm is less than or equal to one.

But what makes H^∞ work is its unique and remarkable "all-pass" property:

At H^∞ optimal, the frequency response of $T_{y_1 u_1}$ is all-pass and equal to one (i.e., $\|T_{y_1 u_1}\| \cong 1$)!

This means that designers can achieve EXACT frequency domain loop-shaping via suitable weighting strategies. For example, one may augment the plant with frequency

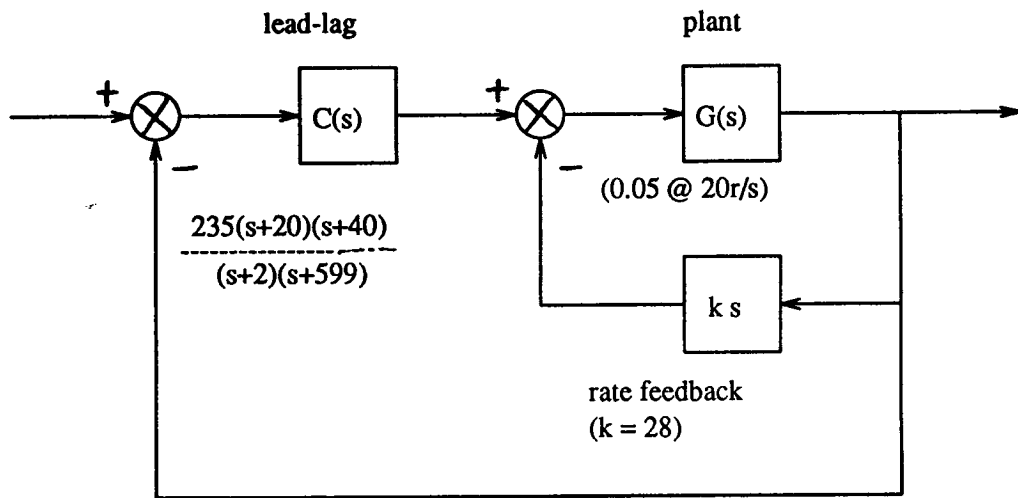


Fig. 3.3 Classical loop-shaping block diagram.

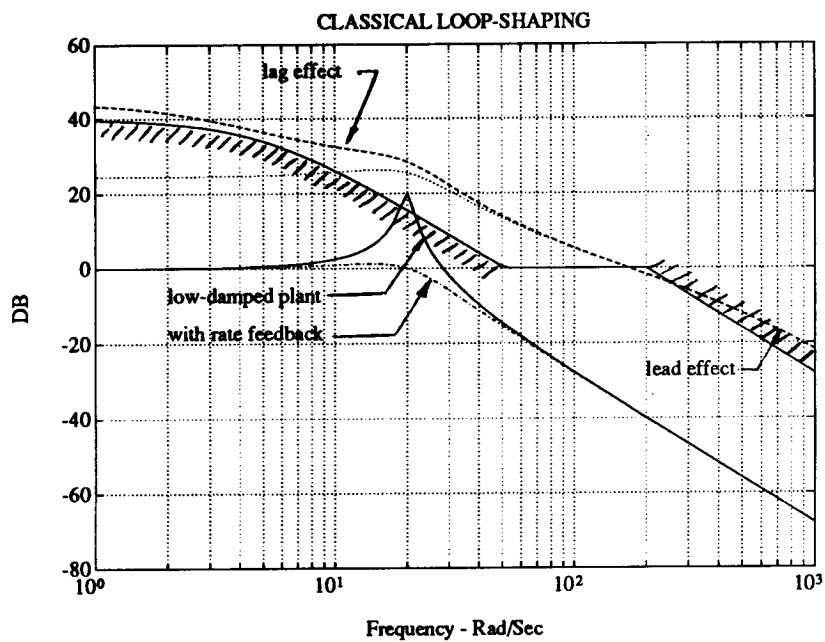


Fig. 3.4 Classical loop-shaping.

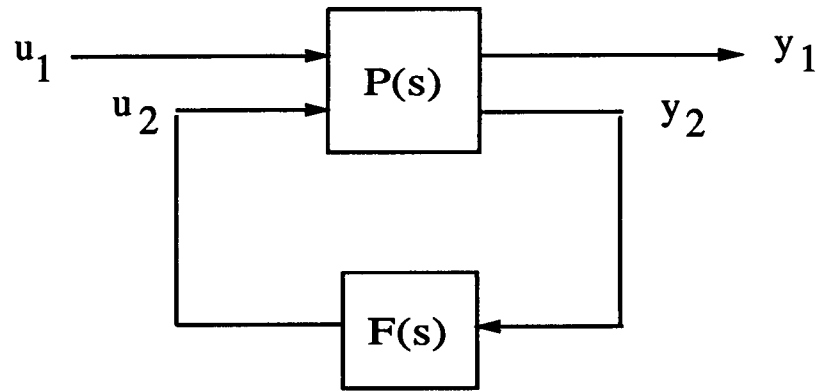


Fig. 3.5 H^∞ Small-Gain Problem.

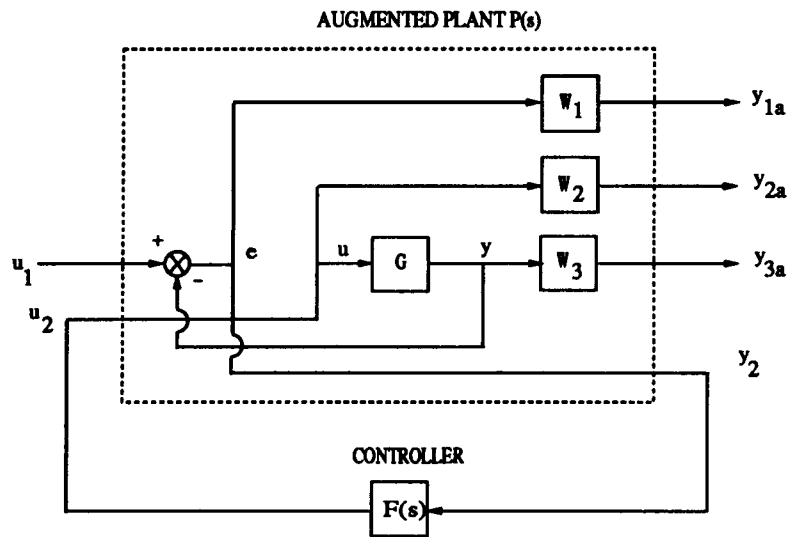


Fig. 3.6 Weighting strategy block diagram.

dependent weights W_1 , W_2 and W_3 as shown in Fig. 3.6. Then, the *Robust-Control Toolbox* functions `augtf.m` and `augss.m` will perform the augmentation and create a state-space for function `hinf.m` to find an H^∞ controller. Of course, these frequency weighting functions have to be chosen so that a stabilizing solution satisfying the H^∞ norm constraint exists.

In a typical application, either $W_2(s)$ or $W_3(s)$ would be absent, leading to weighted H^∞ costs of the forms

$$\min_{F(s)} \left\| \begin{bmatrix} W_1 S \\ W_3 T \end{bmatrix} \right\|_\infty \leq 1 \quad \text{or} \quad \min_{F(s)} \left\| \begin{bmatrix} W_1 S \\ W_2 F S \end{bmatrix} \right\|_\infty \leq 1.$$

In our example, the frequency domain spec. can be split into W_1 and W_3 :

$$W_1^{-1} = \rho \frac{(0.2s + 1)^2}{100(0.005s + 1)^2}; \quad W_3^{-1} = \frac{40000}{s^2}$$

as shown in Fig. 3.7.

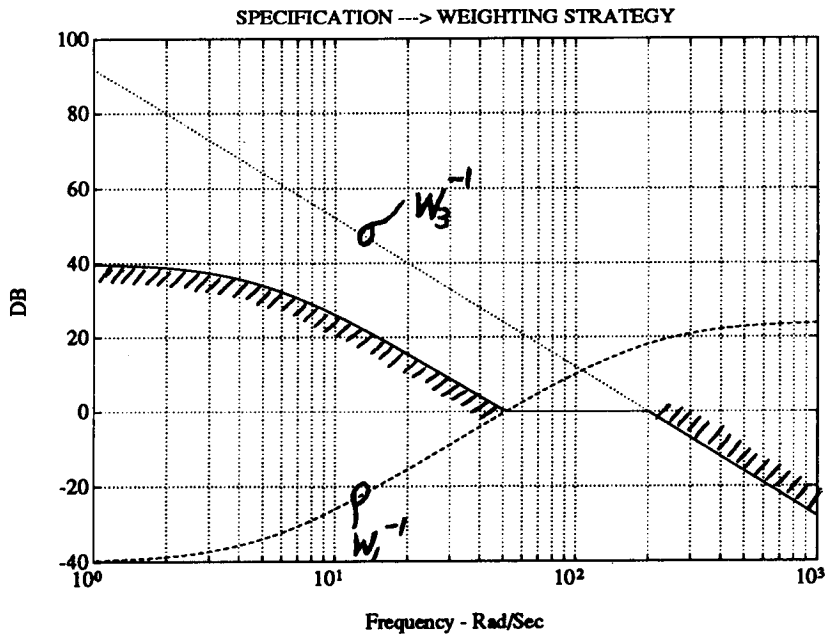


Fig. 3.7 Weighting strategy for 2nd order problem spec.

The results are shown in Fig. 3.8 for different ρ 's. Clearly, in the limit (as ρ goes to 3.16) the cost function becomes "all-pass". The parameter ρ of W_1 is the only parameter on which we iterate for design; the *Robust-Control Toolbox* script-file `hinfgamma.m` automates this iteration.

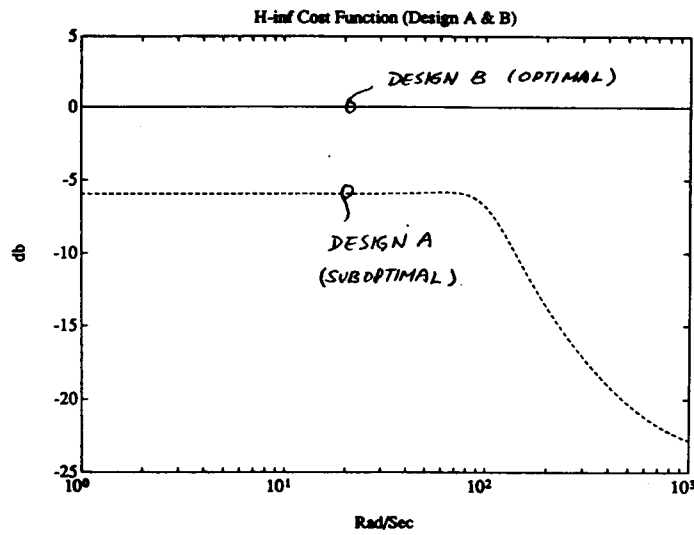
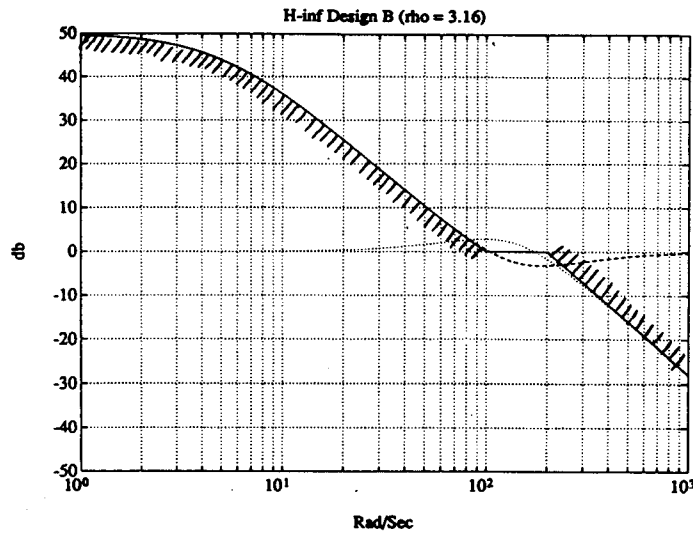
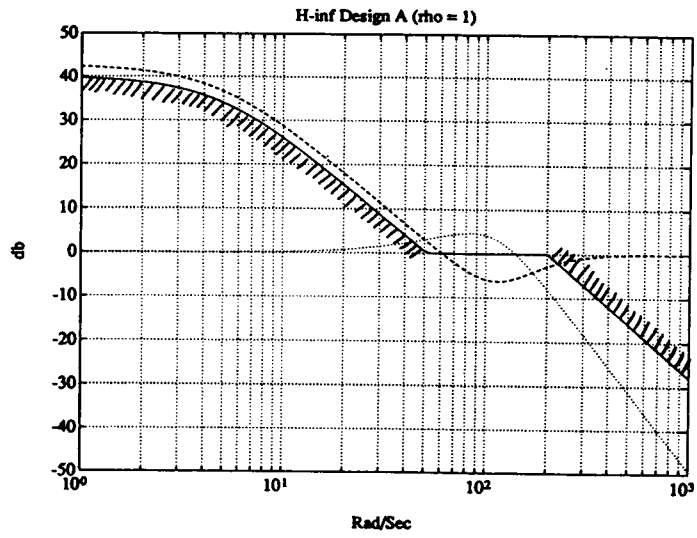


Fig. 3.8 H^∞ results for 2nd order system

3.1 H^∞ Software Execution and Sample Run

To do an H^∞ control design with the *Robust-Control Toolbox* is relatively simple. Table 3.2.1 shows the complete user inputs for our sample problem.

Table 3.2.1

```
>> nug = [0 0 400]; dng = [1 2 400];
>> [ag,bg,cg,dg] = tf2ss(nug,dng);
>> sysg = [ag bg;cg dg]; xg = 2;
>> w1 = [2.5e-5 1.e-2 1;
        0.01*[4.e-2 4.e-1 1]];
>> w2 = [ ];
>> w3 = [1 0 0;0 0 40000];
>> [A,B1,B2,C1,C2,D11,D12,D21,D22] = augtf(sysg,xg,w1,w2,w3);
>> hinf
```

Table 3.2.2 shows the output which appears on the screen for a successful run of hinf.m.

Table 3.2.2

```
<< H-inf Optimal Control Synthesis >>

Computing the 4-block H-inf optimal controller
using the S-L-C loop-shifting/descriptor formulae

- - - Solving for the H-inf controller F(s) using U(s) = 0 (default) - - -

Solving riccati equations and performing H-infinity
existence tests:
  1. Is D11 small enough? OK
  2. Solving state-feedback (P) riccati ...
     a. No Hamiltonian jw-axis roots? OK
     b. A-B2*F stable (P >= 0)? OK
  3. Solving output-injection (S) riccati ...
     a. No Hamiltonian jw-axis roots? OK
     b. A-G*C2 stable (S >= 0)? OK
  4. max eig(P*S) < 1 ? OK

-----
all tests passed -- computing H-inf controller ...
DONE!!!
-----
```

4 Robust Model Reduction via Basis-Free Techniques

In the design of controllers for complicated systems, model reduction arises in several places: 1). Plant model reduction, 2). Controller model reduction, 3). Simulation of large size problem.

However, naive implementations of model reduction methods such as Rosenbrock's stair-case algorithm, Moore's balanced truncation, optimal Hankel approximation and balanced stochastic truncation, etc., can fail on even relatively simple problems due to numerical instability.

The *Robust-Control Toolbox* implements the "basis-free" version of the latter three of the model reduction techniques, which are not only numerically robust but also tend to achieve the ultimate result for practical problems. In particular, they all possess the following special features:

1. They bypass the ill-conditioned balancing transformation, so that they can easily deal with the "non-minimal" systems.
2. They employ Schur decomposition to robustly compute the orthogonal bases for eigenspaces required in intermediate steps.

These methods all enjoy attractive L^∞ error bounds – either an *additive error bound* or a *multiplicative error bound*.

- **Additive Methods:**

- Optimal descriptor Hankel MDA (ohklmr.m).
- Schur balanced truncation (balmr.m, schmr.m).

- **Multiplicative Method:**

- Schur balanced stochastic truncation (reschmr.m).

4.1 Robust Model Reduction Theorems

For robust control system design, it is desirable that the reduced order model satisfies the conditions of one of the following two theorems (ref. [1] [2]). Otherwise, the controller design based on the "blind" reduced order plant model can be unstable !

Theorem 2 Additive Robustness Theorem: (see Fig. 4.1)

If $\bar{\sigma}(\tilde{\Delta}_A) \leq \underline{\sigma}(\tilde{G})$ for $\omega \leq \omega_r$ (with $\tilde{\Delta}_A$ and \tilde{G} open loop stable), then the closed-loop system will be stable provided that the control bandwidth is less than ω_r , where $\omega_r := \max\{\omega \mid \underline{\sigma}(\tilde{G}(j\omega)) \geq \bar{\sigma}(\tilde{\Delta}_A(j\omega))\}$.

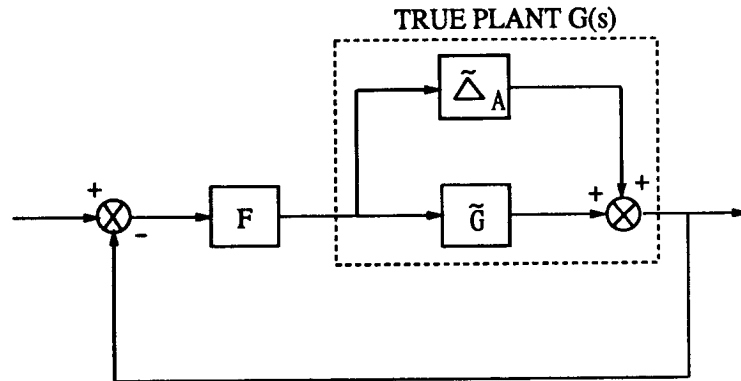


Fig. 4.1 Additive modeling error.

Theorem 3 Multiplicative Robustness Theorem: (see Fig. 4.2)

If $\bar{\sigma}(\tilde{\Delta}_M) \leq 1$ for $\omega \leq \omega_r$ (with $\tilde{\Delta}_M$ and \tilde{G} open loop stable), then the closed-loop system will be stable provided that the control bandwidth is less than ω_r , where $\omega_r := \max\{\omega \mid \bar{\sigma}(\tilde{\Delta}_M(j\omega)) \leq 1\}$.

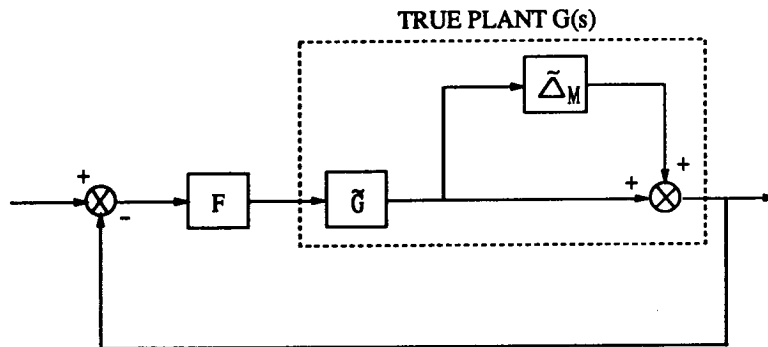


Fig. 4.2 Multiplicative modeling error.

4.2 Examples of Model Reduction

Example 1: Find a 3-state reduced order model for the transfer function

$$G(s) = \frac{0.05(s^7 + 801s^6 + 1024s^5 + 599s^4 + 451s^3 + 119s^2 + 49s + 5.55)}{s^7 + 12.6s^6 + 53.48s^5 + 90.94s^4 + 71.83s^3 + 27.22s^2 + 4.75s + 0.3}$$

with Hankel singular values of the phase matrix

σ_1	σ_2	σ_3	σ_4	σ_5	σ_6	σ_7
0.9959	0.9972	0.9734	0.7166	0.5635	0.0021	0

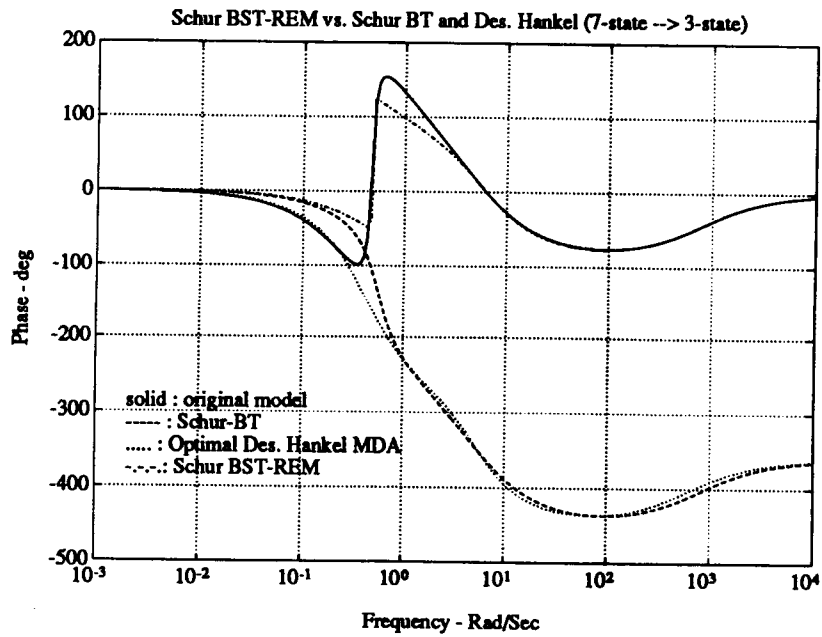
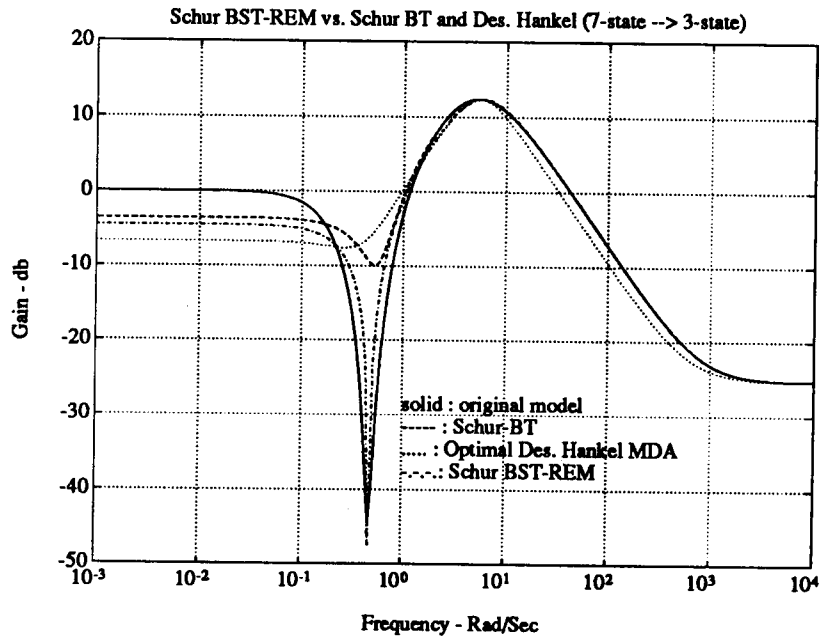


Fig. 4.3 Schur BST-REM vs. Schur BT and Descriptor Hankel.

The results produced by the *Robust-Control Toolbox* functions – `ohklmr.m`, `schmr.m`, `reschmr.m`, are shown in Fig. 4.3. Clearly, the Schur BST-REM method that keeps the reduced model staying inside a prescribed relative error bound produces the ultimate result in model reduction. Note that $\sigma_7 = 0$ indicates only the “basis-free” methods can handle the problem without numerical difficulty.

Example 2: Model reduction for a large space structure [6] (see Fig. 4.4).

Our design requirement is to find a controller to track LOS loops in 300 Hz BW and to reduce plant disturbance response by a factor of 100.

The Hankel singular values after the inner loops are closed indicate that the system is non-minimal. Therefore, only the “basis-free” methods such as – Schur BT (`schmr.m`) or Schur BST-REM (`reschmr.m`) can be used. Again, only the Schur BST-REM can match the original model up to a “robust frequency” which is high enough so that the required BW of 300 Hz can be satisfied (see Fig. 4.5 & Fig. 4.6).

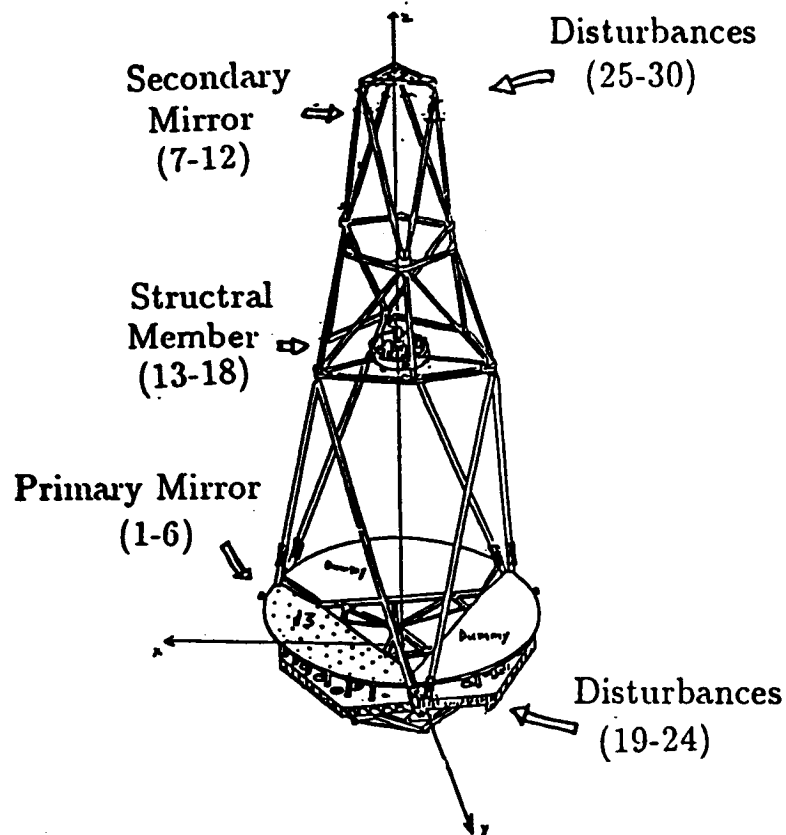


Fig. 4.4 Large space structure.

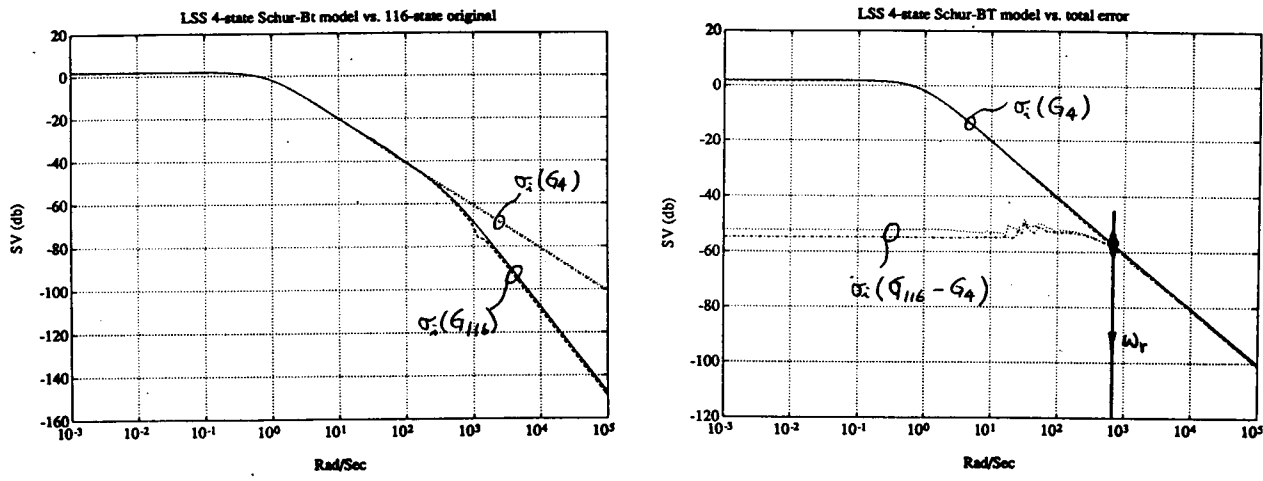


Fig. 4.5 Model reduction using Schur Balanced Truncation.

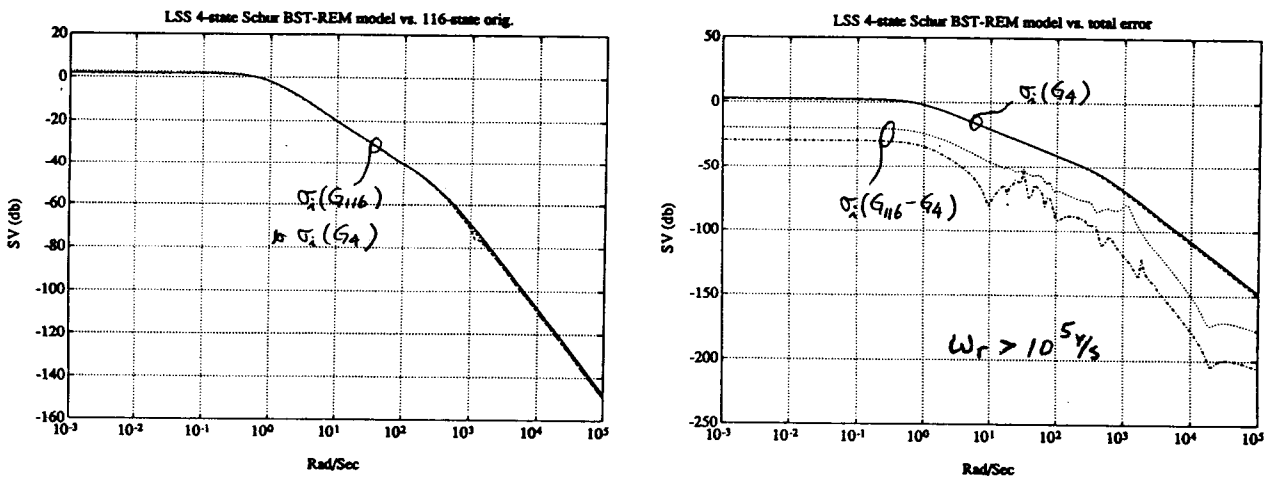


Fig. 4.6 Model reduction using Schur Balanced Stochastic Truncation.

5 Importance to the Control Community

If the ultimate goal of research is to apply the theory to reality, then the contribution of *Robust-Control Toolbox* is clear:

It provides a vital bridge between modern robust control theory and real control applications.

The following diagram (Fig. 5.1) shows how different groups of people with different backgrounds can utilize the *Robust-Control Toolbox* to achieve their personal goals. For example classical control designers can use the toolbox to understand the theory or to apply on a design. Non-robust control researchers can study the code provided by the toolbox together with the papers referenced therein to become familiar with the robust control theory. Students can use the toolbox either for robust control research or for realistic design studies. It seems that the *Robust-Control Toolbox* can serve people from a variety of backgrounds.

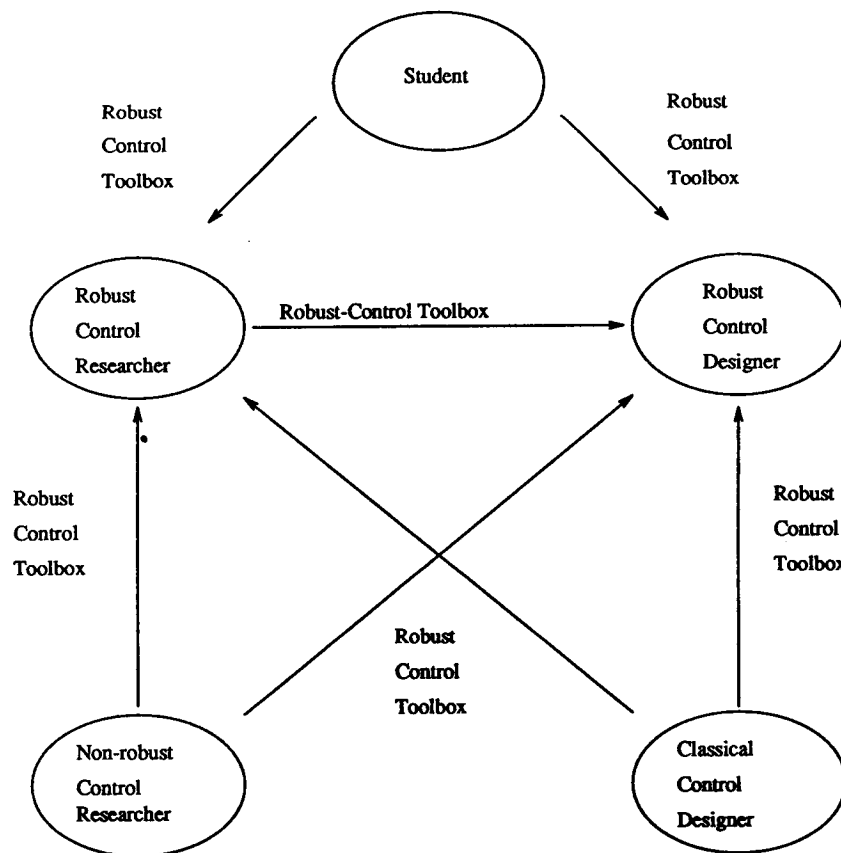


Fig. 5.1 The importance of the *Robust-Control Toolbox*.

6 Summary

The *Robust-Control Toolbox*

- Provides a bridge between modern robust control theory and the real-world applications.
- Has the most up-to-date Robust-Control theories and algorithms.
- Is in *readable* M-files, so it's educational.
- Contains the tools one needs to do robust control system design, analysis and model reduction.
- Is direct, powerful and easy-to-use.

References

- [1] R. Y. Chiang and M. G. Safonov, *Robust-Control Toolbox*. So. Natick, MA: Math-Works, 1988.
- [2] R. Y. Chiang, *Modern Robust Control Theory*. Ph.D. dissertation, USC, 1988.
- [3] M. G. Safonov and R. Y. Chiang, "CACSD using the State-Space L^∞ Theory - A Design Example", *IEEE Trans. on Automat. Contr.*, vol. AC-33, no. 5, pp. 477-479, 1988.
- [4] M. G. Safonov and D. J. N. Limebeer, "Simplifying the H^∞ Theory via Loop-Shifting", *Proc. IEEE Conf. on Decision and Control*, Austin, TX, Dec. 7-9, 1988.
- [5] M. G. Safonov, D. J. N. Limebeer and R. Y. Chiang, "Simplifying the H^∞ Theory via Loop-Shifting, Matrix Pencil and Descriptor Concepts", to appear *Int. J. Control* 1989.
- [6] M. G. Safonov, R. Y. Chiang and H. Flashner, " H^∞ Control Synthesis for a Large Space Structure," *Proc. of American Contr. Conf.*, Atlanta, GA. June 15-17, 1988. Submitted to *J. of Guidance and Control*, June, 1989.
- [7] M. G. Safonov, *Robustness and Stability Aspects of Stochastic Multivariable Feedback System Design*, Ph.D. dissertation, MIT, 1977. Also, M. G. Safonov, *Stability and Robustness of Multivariable Feedback Systems*. Cambridge, MA: MIT Press, 1980.
- [8] J. Tekawy, M. G. Safonov and R. Y. Chiang, "Computer Algorithms for Multivariable Stability Margin", *Proc. of 3rd Annual Conference on Aerospace Computational Control*, Oxnard, CA, Aug. 28-30, 1989.