DSC-Vol. 43

ADVANCES IN ROBUST AND NONLINEAR CONTROL SYSTEMS

EDITED BY
E. A. MISAWA

DSC-Vol. 43

Advances in Robust and Nonlinear Control Systems

PRESENTED AT
THE WINTER ANNUAL MEETING OF
THE AMERICAN SOCIETY OF MECHANICAL ENGINEERS
ANAHEIM, CALIFORNIA
NOVEMBER 8–13, 1992

SPONSORED BY
THE DYNAMIC SYSTEMS AND CONTROL DIVISION, ASME

EDITED BY
EDUARDO A. MISAWA
OKLAHOMA STATE UNIVERSITY

THE AMERICAN SOCIETY OF MECHANICAL ENGINEERS
345 East 47th Street United Engineering Center New York, N.Y. 10017

ISBN No. 0-7918-1114-X

Library of Congress
Catalog Number 92-56569

FOREWORD

This volume presents some of the recent advances in control theories and applications for nonlinear and/or uncertain systems. These papers were presented in three technical sessions during the 1992 Winter Annual Meeting of The American Society of Mechanical Engineers (ASME) held in Anaheim, California. The technical sessions and the editing of this volume were sponsored by the System Theory Technical Panel of the Dynamic Systems and Control Division of the ASME.

All of the papers presented in the volume deal with the theory and application of robust control and estimation system design methodologies for dynamic nonlinear systems with structured and/or unstructured uncertainties.

The first section presents recent advances in the theory and application of robust control systems design for structured undetainties.

The second section presents the application of nonlinear control systems to mechanical systems, namely flexible structures and induction motors.

The third section discusses applications of robust nonlinear controllers (sliding mode control, robust adaptive control, feedback linearization and fuzzy logic control methodologies) and estimators, both from the application and design perspectives.

The editor is sincerely grateful to all of the authors and the technical session organizers for their valuable contributions, which made this book possible. The editor would also like to thank Ms. Leah Kinnaird for secretarial assistance.

Eduardo A. Misawa
Oklahoma State University

CONTENTS

DSC-Vol. 43, Advances in Robust and Nonlinear Control Systems
ASME 1992

ON ROBUST STABILITY AND PERFORMANCE WITH TIME-VARYING CONTROL

Mohammed Dahleh
Department of Mechanical and Environmental Engineering
University of California
Santa Barbara, California

Abstract

In this paper we will show that time-varying compensation provides no improvement over time-invariant compensation for the stability and performance robustness of discrete-time systems with structured norm-bounded perturbations. The perturbations are causal, time-varying and bounded in the ℓ_2-induced norm sense.

1 Notation

ℓ^2 Space of all square-summable sequences of real numbers.

ℓ^2_n Space of n-tuples of elements of ℓ^2.

$\ell^2_{n,e}$ Extended ℓ^2_n space. It is equal to the space of all n-tuples of sequences of real numbers.

S_k Right shift by k positions. If $x = \{x(i)\}_{i=0}^{\infty}$ is any sequence and k is a nonnegative integer, then $S_k x = \{\overbrace{0, \ldots, 0}^{k}, x(0), x(1), \ldots\}$. On the other hand, $S_{-k}x = \{x(k), x(k+1), \ldots\}$. Hence $S_{-k}S_k = I$ but $S_k S_{-k} \neq I$.

$\mathcal{L}_{TV}^{p\times q}$ The space of all bounded linear causal operators mapping ℓ^2_q to ℓ^2_p. If $R \in \mathcal{L}_{TV}^{p\times q}$ then $\|R\| := \sup_{x\neq 0} \frac{\|Rx\|_2}{\|x\|_2}$ which is the induced operator norm.

$\mathcal{L}_{TI}^{p\times q}$ Subspace of $\mathcal{L}_{TV}^{p\times q}$ consisting of time-invariant operators.

$\mathbf{\Delta}$ The set of all causal operators mapping ℓ^2 to itself, with induced ℓ^2 norm less than or equal to one. Hence, $\mathbf{\Delta} := \left\{\Delta : \sup_{x\neq 0} \frac{\|\Delta x\|_2}{\|x\|_2} \leq 1\right\}$.

$\mathcal{D}(n)$ The set of all diagonal operators of the form $\Delta = diag(\Delta_1, \ldots, \Delta_n)$ where $\Delta_i \in \mathbf{\Delta}$.

$\mathcal{T}(n)$ The set of all diagonal matrices of the form $T = diag(t_1, \ldots, t_n)$ where $t_i > 0$.

2 Introduction

The possible advantages of time-varying control for a certain class of robust stabilization problems have been the subject of research in the past few years. Starting with unstructured perturbation problems in the ℓ_2 setting it was shown that under certain conditions time-varying control offers no benefit for such problems (see Khargonekar and Poolla (1989), or Chapellat and Dahleh (1992) for a description of such results). For structured uncertainty problems in the ℓ_∞ setting a similar conclusion was reached in Khammash and Dahleh (1992).

In this paper we will show that the analogous result to Khammash and Dahleh (1992) in the ℓ_∞ case is also true in the ℓ_2 case.

3 Setup

The setup for treating the stability robustness of time-varying systems in the presence of structured norm-bounded uncertainty is shown in figure 1. M is the interconnection of nominal linear time-varying plant/plants and linear time-varying stabilizing controller. $\Delta = diag(\Delta_1, \dots, \Delta_n)$ is a structured perturbation consisting of n perturbation blocks representing the system uncertainty which is assumed to take place in n different locations in the interconnection. Each perturbation block, Δ_i belongs to the class $\mathbf{\Delta}$ and is therefore norm bounded. The Δ_i's are independent of each other reflecting the situation when the uncertainty has different sources. We will assume that M is strictly causal to guarantee the well-posedness of the interconnection. Robust stability is achieved if the mapping $(u_1, u_2) \mapsto (e_1, e_2, y_1, y_2)$ maps $\ell_n^2 \times \ell_n^2$ into $\ell_n^2 \times \ell_n^2 \times \ell_n^2 \times \ell_n^2$ for all $\Delta \in \mathcal{D}(n)$. Since M and Δ are each ℓ^2- stable , robust stability is achieved if and only if $(I - M\Delta)^{-1}$ is ℓ^2- stable for all $\Delta \in \mathcal{D}(n)$. Throughout this paper, this will be our robust stability condition.

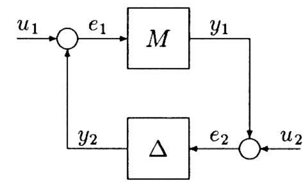

Figure 1: Stability Robustness Problem for Time-Varying Systems

4 Time-Varying vs. Time-Invariant Compensation

In what follows, if M is a time-varying system and m is any integer, $M^{(m)}$ is defined to be $S_{-m}MS_m$.

Theorem 1 (Shamma (1992)) *Let $M \in \mathcal{L}_{TV}^{n \times n}$. $(I - M\Delta)^{-1}$ is ℓ^2- stable for all $\Delta \in \mathcal{D}(n)$ only if*

$$\inf_{T \in T(n)} \inf_{m \geq 0} \|TM^{(m)}T^{-1}\| < 1.$$

Proof. see Shamma (1992). ∎

Let $M(Q) := H - UQV$, where H, U, and V belong to $\mathcal{L}_{TI}^{n \times n}$, $\mathcal{L}_{TI}^{n \times p}$, and $\mathcal{L}_{TI}^{q \times n}$ respectively, while $Q \in \mathcal{L}_{TV}^{p \times q}$. The following theorem states that when structured perturbations are present, time-varying compensation *cannot* improve on the stability robustness obtained through the use of time-invariant compensation.

Theorem 2 *There exists $Q \in \mathcal{L}_{TV}^{p \times q}$ such that $(I - M(Q)\Delta)^{-1}$ is ℓ^2- stable for all $\Delta \in \mathcal{D}(n)$ if and only if there exists $Q' \in \mathcal{L}_{TI}^{p \times q}$ such that $(I - M(Q')\Delta)^{-1}$ is ℓ^2- stable for all $\Delta \in \mathcal{D}(n)$.*

Proof. The sufficiency is trivial since $\mathcal{L}_{TI}^{p \times q} \subset \mathcal{L}_{TV}^{p \times q}$. So we prove the necessity.

Suppose $Q \in \mathcal{L}_{TV}^{p \times q}$ such that $(I - M(Q)\Delta)^{-1}$ is ℓ^2- stable for all $\Delta \in \mathcal{D}(n)$. From theorem 1, this implies that for some integer m, we have:

$$\inf_{T \in T(n)} \inf_{m \geq 0} \|TM^{(m)}(Q)T^{-1}\| < 1.$$

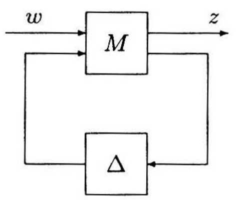

Figure 2: Performance Robustness Problem

Since $S_{-k}(H - UQV)S_k$ is equal to $H - US_{-k}QS_kV$, we have :

$$\inf_{T \in \mathcal{T}(N)} \inf_{m \geq 0} \|TM(Q^{(m)})T^{-1}\| < 1.$$

$Q^{(m)} \in \mathcal{L}_{TV}^{p \times q}$, and there always exists $Q' \in \mathcal{L}_{TI}^{p \times q}$ such that $\|TM_i(Q')T^{-1}\| \leq \|TM_i(Q^{(m)})T^{-1}\|$ (see Shamma and Dahleh (1991) and Chapellat and Dahleh (1992) for a proof). This is a necessary and sufficient condition for $(I - M(Q')\Delta)^{-1}$ to be ℓ^2- stable for all $\Delta \in \mathcal{D}(n)$. This completes the proof. ■

Now we will analyze the effect of time-varying compensation on the performance robustness. Fig. 2 shows the setup used for robust performance problems. w will be the exogenous inputs which are assumed to be in ℓ^2, and z will be the regulated output. M and Δ_i's are as before, except that M now has $n+1$ inputs and $n+1$ outputs. Robust performance is achieved if robust stability is achieved and the induced ℓ^2 norm of the operater mapping w to z is less than one for all $\Delta \in \mathcal{D}(n)$.

Theorem 3 *There exists $Q \in \mathcal{L}_{TV}^{p \times q}$ for which the system in Fig. 2 achieves robust performance if and only if there exists $Q' \in \mathcal{L}_{TI}^{p \times q}$ for which the same system achieves robust performance.*

Proof. Here again, the sufficiency is trivial. For the necessity, suppose there exists $Q \in \mathcal{L}_{TV}^{p \times q}$ for which the system in the figure achieves robust performance. It follows from the Small Gain Theorem that $(I - M(Q)\Delta)^{-1}$ is ℓ^2- stable for all $\Delta \in \mathcal{D}(n+1)$. By Theorem 2, there exists $Q' \in \mathcal{L}_{TI}^{p \times q}$ for which $(I - M(Q')\Delta)^{-1}$ is ℓ^2- stable for all $\Delta \in \mathcal{D}(n+1)$. This implies that for Q' the system in Fig. 2 achieves robust performance. ■

The results in theorems 2 and 3 are true in the discrete-time case, and almost identical proofs will yield the same conclusion for the continuous-time case.

5 Conclusion

We have shown that linear time-invariant controllers perform as well as time-varying controllers for the problem of robust stability and performance for systems with structured norm bounded perturbations. The resutls are an extension of the ℓ_∞-case (Khammash and Dahleh, 1992).

6 Acknowledgment

Research Supported by NSF Grant ECS-9111058.

References

[1] Chapellat, H. and M. Dahleh, 1992, "Analysis of Time-Varying Control Strategies for Optimal Disturbance Rejection and Robustness," *to appear in the IEEE Trans. on Automatic Control.*

[2] Khammash, M. and M. Dahleh, 1992, "Time-Varying Control and the Robust Performance of Systems with Structured Norm-Bounded Perturbations," *Automatica*, Vol. 28, no. 4, pp. 819-821, July, 1992.

[3] Khargonekar, P. P. and K. Poolla, 1989, "Robust Control of Linear Time-Invariant Plants Using Switching and Nonlinear Feedback," *Proceeding of the 28th Conference on Decision and Control*, Tampa, Florida, pp. 2205-2207, 1989.

[4] Shamma, J. S. and M. A. Dahleh, 1991, "Time-Varying vs. Time-Invariant Compensation for Rejection of Persistent Bounded Disturbances and Robust Stabilization," *IEEE Trans. Automat. Contr.*, vol. AC-36, pp. 838-847, 1991.

[5] J. S. Shamma, 1992, "Robust Stability with Time-Varying Structured Uncertainty,"*to appear in 31st IEEE Conference on Decision and Control.*

DSC-Vol. 43, Advances in Robust and Nonlinear Control Systems
ASME 1992

LET'S GET REAL

Peter M. Young, Matthew P. Newlin, and John C. Doyle
Department of Electrical Engineering
California Institute of Technology
Pasadena, California

Abstract

This paper gives an overview of promising new developments in robust stability and performance analysis of linear control systems with real parametric uncertainty. The goal is to develop a practical algorithm for medium size problems, where medium size means less than 100 real parameters, and "practical" means avoiding combinatoric (nonpolynomial) growth in computation with the number of parameters for all of the problems which arise in engineering applications. We present an algorithm and experimental evidence to suggest that this goal has, for the first time, been achieved. We also place these results in context by comparing with other approaches to robustness analysis and considering potential extensions, including controller synthesis.

1 Introduction

Robust stability and performance analysis with real parametric uncertainty can be naturally formulated as a Structured Singular Value, or μ, problem, where the block structured uncertainty description is allowed to contain both real and complex blocks. It is assumed that the reader is familiar with this type of robustness analysis, as space constraints preclude covering both the engineering motivation and the computational issues. For more engineering motivation of the use of these approaches, see [1, 2] and the references therein.

The approach to mixed μ computation taken in this paper involves upper and lower bounds [3, 4] which are usually reasonably close, together with a Branch and Bound scheme to refine the bounds when necessary. The upper bound was presented in [3] and involves minimizing the eigenvalues of a Hermitian matrix, while the lower bound can be tackled with a modified power algorithm [4]. Practical computation schemes for these bounds are described in section 5 and are available in conjunction with the μ-Tools toolbox [1]. The quality of these bounds, and their computational requirements as a function of problem size, are explored in [5] and in section 5.3. While the bounds are usually accurate enough for engineering purposes, occasionally they are not. This is in contrast with the purely complex nonrepeated case, where no examples of problems with large gaps have been found. The use of Branch and Bound schemes to improve upon existing bounds has been suggested by several authors (see [6, 7, 8] and the references therein).

It is now well known that real μ problems can be discontinuous in the problem data [9] and that computation for the general mixed μ problem is NP complete [10, 11, 12]. Neither result is surprising in retrospect, and in section 4, we will consider the implications of these results for computation of mixed μ. We will argue that the discontinuities have little relevance for problems motivated by engineering applications, but the NP completeness results are extremely important, having adverse implications for certain research directions.

Roughly speaking, the fact that mixed μ is NP complete means that it cannot be computed exactly in the worst case without entirely unacceptable growth in computation cost with problem size. To obtain acceptable computation one is forced either to consider special cases, or relax the requirement for exact computation on worst-case problems. In section 4.3 we consider several special cases, including those involved in Kharitonov's theorem [13] and its extensions. Essentially all of these special cases can be viewed as problems where μ is equal to its upper bound, which is relatively easy to compute. Unfortunately, few engineering problems fit any of these special cases, so they are of limited practical value, and the NP completeness results strongly suggest that they cannot be usefully extended.

Since the general mixed μ problem is NP complete, we do not attempt to solve it exactly but rather obtain good bounds. Furthermore, recent results [11] suggest that even approximate methods are also NP complete, so we will not expect good worst case behavior but rather aim for good typical behavior. Practical algorithms for other NP complete problems exist and typically involve approximation, heuristics, branch-and-bound, or local search. Results presented in [5] and this paper strongly suggest that an intelligent combination of all these techniques can yield a practical algorithm for the mixed μ problem. On the other hand, routine application of any of these methods generally seems to produce algorithms with clearly exponential growth rates even on small problems. For example, using Branch and Bound with crude bounds (see [6]) appears to have exponential growth on typical problems [5].

A selection of results from a fairly extensive numerical study of these issues is presented in section 6. These numerical experiments suggests that if one is interested in solving fairly large problems, then one can only expect the Branch and Bound scheme to achieve a degree of accuracy that the bounds usually get anyway, which in this case is approximately 20%. Thus the Branch and Bound scheme is not being used as a general computation scheme per se, but only to fix the occasional problems for which the bounds are poor, and for these problems to achieve the degree of accuracy which the bounds typically get. This reinforces the results in [5] and emphasizes the necessity for good bounds. Fortunately, computing μ to within 20% accuracy is generally quite adequate for engineering purposes.

2 Notation and Definitions

The notation used here is fairly standard and is essentially taken from [3] and [4]. For any square complex matrix M we denote the complex conjugate transpose by M^*. The largest singular value and the structured singular value are denoted by $\overline{\sigma}(M)$ and $\mu_{\mathcal{K}}(M)$ respectively. The spectral radius is denoted $\rho(M)$ and $\rho_R(M) = max\{|\lambda| : \lambda \; is \; a \; real \; eigenvalue \; of \; M\}$, with $\rho_R(M) = 0$ if M has no real eigenvalues. For a Hermitian matrix M, then $\overline{\lambda}(M)$ and $\lambda_{min}(M)$ denote the largest and smallest (real) eigenvalues respectively. For any complex vector x, then x^* denotes the complex conjugate transpose and $|x|$ the Euclidean norm. We denote the $k \times k$ identity matrix and zero matrix by I_k and O_k respectively.

The definition of μ is dependent upon the underlying block structure of the uncertainties, which is defined as follows. Suppose we have a matrix $M \in \mathbb{C}^{n\times n}$ and three non-negative integers m_r, m_c, and m_C (with $m := m_r + m_c + m_C \leq n$) which specify the number of uncertainty blocks of each type. Then the block structure $\mathcal{K}(m_r, m_c, m_C)$ is an m-tuple of positive integers

$$\mathcal{K} = (k_1, \ldots, k_{m_r}, k_{m_r+1}, \ldots, k_{m_r+m_c}, k_{m_r+m_c+1}, \ldots, k_m) \qquad (1)$$

This m-tuple specifies the dimensions of the perturbation blocks, and we require $\sum_{i=1}^{m} k_i = n$ in order that these dimensions are compatible with M. This determines the set of allowable perturbations, namely define

$$\begin{aligned} X_{\mathcal{K}} = \{\Delta = block\ diag(&\delta_1^r I_{k_1}, \ldots, \delta_{m_r}^r I_{k_{m_r}}, \delta_1^c I_{k_{m_r+1}}, \ldots, \\ &\delta_{m_c}^c I_{k_{m_r+m_c}}, \Delta_1^C, \ldots, \Delta_{m_C}^C) : \\ &\delta_i^r \in \mathbf{R}, \delta_i^c \in \mathbf{C}, \Delta_i^C \in \mathbf{C}^{k_{m_r+m_c+i} \times k_{m_r+m_c+i}}\} \end{aligned} \tag{2}$$

Note that $X_{\mathcal{K}} \in \mathbf{C}^{n \times n}$ and that this block structure is sufficiently general to allow for (any combination of) repeated real scalars, repeated complex scalars, and full complex blocks. The purely complex case corresponds to $m_r = 0$, and the purely real case to $m_c = m_C = 0$.

Note also that all the results which follow are easily generalized to the case where the full complex blocks need not be square, and the blocks may come in any order. We make these restrictions in (2) purely for notational convenience.

Definition 1 ([14]) *The structured singular value, $\mu_{\mathcal{K}}(M)$, of a matrix $M \in \mathbf{C}^{n \times n}$ with respect to a block structure $\mathcal{K}(m_r, m_c, m_C)$ is defined as*

$$\mu_{\mathcal{K}}(M) = \left(\min_{\Delta \in X_{\mathcal{K}}} \{\overline{\sigma}(\Delta) : det(I - \Delta M) = 0\} \right)^{-1} \tag{3}$$

with $\mu_{\mathcal{K}}(M) = 0$ if no $\Delta \in X_{\mathcal{K}}$ solves $det(I - \Delta M) = 0$.

In order to develop the upper and lower bound theory we need to define some sets of block diagonal scaling matrices (which, like μ itself, are also dependent on the underlying block structure).

$$\mathcal{Q}_{\mathcal{K}} = \{\Delta \in X_{\mathcal{K}} : \delta_i^r \in [-1\ 1], \delta_i^{c*}\delta_i^c = 1, \Delta_i^{C*}\Delta_i^C = I_{k_{m_r+m_c+i}}\} \tag{4}$$

$$\begin{aligned} \mathcal{D}_{\mathcal{K}} = \{block\ diag(&D_1, \ldots, D_{m_r+m_c}, d_1 I_{k_{m_r+m_c+1}}, \ldots, d_{m_C} I_{k_m}) : \\ &0 < D_i = D_i^* \in \mathbf{C}^{k_i \times k_i}, 0 < d_i \in \mathbf{R}\} \end{aligned} \tag{5}$$

$$\begin{aligned} \mathcal{G}_{\mathcal{K}} = \{block\ diag(&G_1, \ldots, G_{m_r}, O_{k_{m_r+1}}, \ldots, O_{k_m}) : \\ &G_i = G_i^* \in \mathbf{C}^{k_i \times k_i}\} \end{aligned} \tag{6}$$

$$\begin{aligned} \hat{\mathcal{D}}_{\mathcal{K}} = \{block\ diag(&D_1, \ldots, D_{m_r+m_c}, d_1 I_{k_{m_r+m_c+1}}, \ldots, d_{m_C} I_{k_m}) : \\ &det(D_i) \neq 0, D_i \in \mathbf{C}^{k_i \times k_i}, d_i \neq 0, d_i \in \mathbf{C}\} \end{aligned} \tag{7}$$

$$\hat{\mathcal{G}}_{\mathcal{K}} = \{block\ diag(g_1, \ldots, g_{n_r}, O_{n_c}) : g_i \in \mathbf{R}\} \tag{8}$$

where $n_r = \sum_{i=1}^{m_r} k_i$ and $n_c = n - n_r$.

3 Upper and Lower Bounds for Mixed μ

First consider the computation of a lower bound. Note that one cannot simply 'cover' the real perturbations with complex ones (and then use the complex μ lower bound) since that would include perturbations from outside the permissible set $X_{\mathcal{K}}$, and so would not yield a valid lower bound. The key to obtaining a lower bound lies in the fact that the μ problem may be reformulated as a real eigenvalue maximization. The following theorem is taken from [4].

Theorem 1 ([4]) *For any matrix $M \in \mathbf{C}^{n \times n}$, and any compatible block structure $\mathcal{K}$*

$$\max_{Q \in \mathcal{Q}_{\mathcal{K}}} \rho_R(QM) = \mu_{\mathcal{K}}(M) \tag{9}$$

This immediately gives us a theoretical lower bound since we have that for any $Q \in \mathcal{Q}_{\mathcal{K}}$, $\rho_R(QM) \leq \mu_{\mathcal{K}}(M)$. The idea then is to find an efficient way to compute a local maximum of the function $\rho_R(QM)$ over $Q \in \mathcal{Q}_{\mathcal{K}}$. Note that since this function is non-convex we cannot guarantee to find the global maximum and hence we only obtain a lower bound for μ. The practical computation of such a local maximum is discussed in section 5.1.

Now consider an upper bound for μ. One could, for the purposes of the upper bound, replace the real perturbations with complex ones (and then use the complex μ upper bound) since this would cover the admissible perturbation set $X_{\mathcal{K}}$. However this approach does not exploit the phase information that is present in the real perturbations, and hence the bound is frequently poor. The upper bound presented in [3] does exploit this phase information and gives a bound which is never worse than the standard upper bound from complex μ theory (see [15] for example) and is frequently much better. The following theorem is taken from [3].

Theorem 2 ([3]) *For any matrix $M \in \mathbf{C}^{n \times n}$, and any compatible block structure $\mathcal{K}$, suppose α_* is the result of the minimization problem*

$$\alpha_* = \inf_{\substack{D \in \mathcal{D}_{\mathcal{K}} \\ G \in \mathcal{G}_{\mathcal{K}}}} \left[\min_{\alpha \in \mathbf{R}} \{\alpha : (M^*DM + j(GM - M^*G) - \alpha D) \leq 0\} \right] \tag{10}$$

then an upper bound for μ is given by

$$\mu_{\mathcal{K}}(M) \leq \sqrt{\max(0, \alpha_*)} \tag{11}$$

Since the above minimization involves an LMI (Linear Matrix Inequality), it is convex (so that all local minima are global) and hence this bound is computationally attractive. The practical computation of the upper bound is discussed briefly in section 5.2.

4 Fundamental Properties and Their Implications

The problem of analyzing robustness with respect to real parameter variations has received a great deal of attention in recent years. Although there have been many different approaches to the problem, it is only a mild oversimplification to lump these efforts into two major research programs. One is typified by the approach taken in this paper and may be thought of as attempting to extend the complex μ theory [16] to handle real perturbations. The other research program has focused on extending Kharitonov's celebrated result [13] on interval polynomials to more general uncertainty structures. In this section we will briefly try to put both of these approaches in a common context in order to consider the implications of several recent results that are relevant to both research programs. The polynomial approach can easily be studied using the μ/LFT framework used in this paper, so we will adopt this point of view.

We will first consider the implications of the result that the purely real μ problem is discontinuous in the problem data. We will argue that discontinuities will not actually occur in problems of engineering interest. Nevertheless, these results do suggest that mixed μ computation may sometimes be poorly conditioned. In contrast, we will see that the result that mixed μ is, in general, an NP complete problem has very important and direct implications for practical application of any computational schemes. Indeed, this result strongly suggests that entire classes of algorithms that attempt to compute mixed μ will be prohibitively expensive, even on problems of moderate size.

There are two strategies that one can adopt to deal with this apparent intractability of mixed μ computation. One possibility is to consider special cases, and this will be the final topic of this section. We shall see that most special cases for which there are favorable results happen to occur on problems where it can be guaranteed a priori that μ will be equal to its upper bound, and can therefore be computed as a convex optimization problem. This applies to the major results in both the μ and polynomial approaches. Unfortunately, these special cases are relevant to very few problems of engineering interest, so other strategies such as the one advocated in this paper must be adopted.

4.1 Continuity

It is now well known that real μ problems can be discontinuous in the problem data (see [9]). This clearly adds computational difficulties to the problem, since any method involving some type of search (e. g.

frequency response) must address the possibility of missing a point of discontinuity. More importantly however this sheds serious doubt on the usefulness of real μ as a *robustness* measure in such cases. This is because the system model is always a mathematical abstraction from the real world, and is only computed to finite precision, so that it would seem reasonable to require that any type of robustness measure we use be continuous in the problem data.

It is shown in [17] how to regularize these problems by essentially adding a small amount of complex uncertainty to each real uncertainty. This adds a small amount of phase uncertainty to the gain uncertainty. It is then shown that the new mixed μ problem is continuous. This regularization seems reasonably well motivated from an engineering point of view, where unmodeled dynamics would always produce some phase uncertainty.

Furthermore it is shown in [17] that mixed μ problems containing some complex uncertainty are, under some mild assumptions, continuous even without the regularization procedure outlined above (whereas purely real μ problems are not). This is reassuring from an engineering viewpoint since one is usually interested in robust performance problems (which therefore contain at least one complex block), or robust stability problems with some unmodeled dynamics, which are naturally covered with complex uncertainty. Thus in problems of engineering interest, the potential discontinuity of μ should not arise, although conditioning of μ computation could be a problem and needs more study.

4.2 NP Completeness

Recent results in [10] show that a special case of computing μ with real perturbations only is NP complete. While these results do not apply to the complex only case, new results in [12] show that the general mixed (or real) problem is NP complete as well. The results in [12] are based on the fact that the indefinite quadratic programming problem given by

$$\max_{b_l \leq x \leq b_u} |x^* A x + p^* x + c| \tag{12}$$

for $A \in \mathbf{R}^{n \times n}$, $x, p, b_l, b_u \in \mathbf{R}^n$, and $c \in \mathbf{R}$ can be recast as a mixed μ problem. It can be shown easily from known results that the indefinite quadratic programming problem in (12) is NP complete, and it follows that the mixed μ problem is NP complete as well.

It is still a fundamental open question in the theory of computational complexity to determine the exact consequences of a problem being NP complete, and we refer the reader to [18] for an in depth treatment of the subject. However, it is generally accepted that a problem being NP complete means that it cannot be computed in polynomial time in the worst case. It is important to note that being NP complete is a property of the problem itself, not any particular algorithm. The fact that the mixed μ problem is NP complete strongly suggests that given *any* algorithm to compute μ, there will be problems for which the algorithm cannot find the answer in polynomial time. This means that for all practical purposes even moderately large examples of such problems are computationally intractable.

For the reader not familiar with these concepts, we offer the following illustration. Consider the example in table 1. There we have tabulated two different growth rates versus problem size. For each growth rate we have assumed that it represents two different algorithms, one which can solve a size 10 problem in 10 seconds, and one which can solve a size 10 problem in 0.01 seconds. The first growth rate is n^3 (where n is the problem size). This is a polynomial time growth rate, and is typical of algorithms for eigenvalues, singular values etc. The second growth rate is 2^n. This is an exponential (non-polynomial) time growth rate, and is typical of algorithms which require one to check all the edges or vertices of some polytope.

It is readily seen that given an algorithm with a polynomial time growth rate we can apply the algorithm to larger and larger problems with a reasonable increase in the computational requirements. In contrast, for the exponential time growth rate the increase in computational requirements is quite dramatic, and for even moderate sizes the problem rapidly becomes intractable. It is important to note that even if the exponential time algorithm is much faster on small problems it still rapidly becomes impractical as the problem size increases. The overriding implication of all this is that if we wish to be able to handle fairly large problems, we must have polynomial time algorithms, regardless of the speed on small problems. The fact that the mixed μ problem is NP complete means that we cannot expect to find such algorithms if we attempt to solve the general problem exactly for all cases.

Growth	Problem Size (n)				
Rate	10	20	30	40	50
n^3	0.01 seconds	0.08 seconds	0.27 seconds	0.64 seconds	1.25 seconds
	10 seconds	1.33 minutes	4.50 minutes	10.67 minutes	20.83 minutes
2^n	0.01 seconds	10.24 seconds	2.91 hours	124.3 days	348.7 years
	10 seconds	2.84 hours	121.4 days	340.5 years	3.49×10^5 years

Table 1: Comparison of polynomial and exponential time growth rates

These results strongly suggest that it is futile to pursue exact methods for computing μ in the purely real or mixed case for even moderate (less than 100) numbers of real perturbations. One approach to overcoming this difficulty is to consider special cases of the general problem, which may be easier to solve. The difficulty with this approach is that one would like the resulting algorithm to be widely applicable to a large number of engineering problems, and it may be that the special cases that are easily solvable are too restrictive. For this reason we do not adopt this approach here, but rather concentrate on the general problem. Nevertheless, since special cases have been the focus of so much research, the remainder of this section will consider those special cases for which computation of μ is relatively easy.

4.3 Problems with special structure

In light of the NP completeness results in the previous section, it is natural to ask if there are special cases of the mixed μ problem that are relatively easy to compute. Essentially all such cases can be shown to involve problems where it can be verified a priori that μ is equal to its upper bound, and can therefore be computed as a convex optimization problem. Unfortunately, these special cases are relevant to very few problems of engineering interest.

Although it is somewhat artificial, it is useful to separately consider the nominal system and the uncertainty structure (respectively P and Δ in figure 1), as one can get easily computable special cases from restrictions on each one. In the case of the nominal system, computation is easier when it is highly structured, whereas *less* structure on the uncertainty makes computation easier. Of course, problems motivated by real engineering applications typically have general, unstructured nominal systems combined with highly structured uncertainty, exactly the opposite of what is ideal for computation.

For simplicity, consider the standard problem of robust stability for the system in figure 1 where Δ is assumed to be norm bounded by 1. The least structured Δ would be a single block which would be allowed to be an arbitrary nonlinear, time-varying operator. In that case the small gain condition is necessary and sufficient, and the test is simply $\|P\|_\infty < 1$. This test is also iff when Δ is restricted to be causal, and further restricted to either linear time-varying (LTV) or linear time-invariant (LTI).

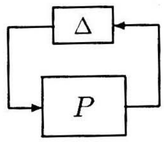

Figure 1: Standard robust stability problem

Additional structure on Δ leads to μ tests of varying complexity, but some special cases exist when μ is equal to its upper bound. If Δ is block diagonal with any number of LTV perturbations then recent results, obtained independently by Shamma and Megretskii, show that the exact test for this case is equivalent to an upper bound for a complex μ problem. Also, if Δ consists of 3 or fewer LTI full blocks, then μ is equal to its upper bound. In general, μ is not equal to its upper bound for more complex uncertainty structures, unless additional structure is imposed on the P. The role of structure on P will be considered in the next section.

4.4 Restrictions on P and "Kharitonov-type" Results

A popular research program over the last few years has focused on extending Kharitonov's celebrated result [13] on interval polynomials, one whose coefficients lie in intervals, to more general uncertainty structures. Kharitonov showed that one need only check 4 polynomials to determine stability of the entire family of interval polynomials. Several additional results have since been proven for other special cases, such as polynomials whose coefficients are affine in some real parameters (see [19] for example), and the solutions typically involve checking the edges or vertices of some polytope in the parameter space. It can be shown that restricting the allowed perturbation dependence to be affine leads to a real μ problem on a transfer matrix which is rank one. Note that this "rank one" assumption is very restrictive. Typically robustness problems motivated by real physical systems do not satisfy this assumption.

The rank one mixed μ problem is studied in detail in [20]. The authors develop an analytic expression for the solution to this problem, which is not only easy to compute, but has sublinear growth in the problem size. They are then able to solve several problems from the literature, noting that these problems can be treated as special cases of "rank one μ problems" and are thus "relatively easy to solve". Even the need to check (a combinatoric number of) edges is shown to be unnecessary.

This rank one case can also be addressed within the framework developed here. The following theorem gives a solution to the rank one mixed μ problem.

Theorem 3 ([21]) *Suppose we have a rank one matrix $M \in \mathbf{C}^{n\times n}$, then $\mu_{\mathcal{K}}(M)$ equals its upper bound from theorem 2.*

Thus for rank one problems μ equals its upper bound and is hence equivalent to a convex problem. There are additional cases where μ is equal to its upper bound, but they are less elegantly characterized.

This theorem reinforces the results of [20] and offers some insight into why the problem becomes so much more difficult when we move away from the "affine parameter variation" case to the "multilinear" or "polynomial" cases [7]. These correspond to μ problems where M is not necessarily rank one, and hence may no longer be equal to the upper bound and so may no longer be equivalent to a convex problem (note that there exist rank two matrices for which μ does not equal its upper bound). This analysis underlines why there are no practical algorithms based on "edge-type" theorems, as the results appear to be relevant only to a very special problem. Furthermore, even in the very special "affine parameter case" there are a combinatoric number of edges to check.

5 Practical Computation of the Bounds

The theoretical bounds described in section 3 form the basis of our computation scheme. However a certain amount of reformulation is required before they can be implemented in an efficient manner, which exploits the structure of the problem. This is described briefly in the remainder of this section and is presented in greater detail in [22]. The algorithm has been implemented in software as a Matlab function (m-file). This has been on β-test at several industrial and academic sites, and is currently available in a test version in conjunction with the μ-Tools toolbox. We also present some numerical experience with the upper and lower bound algorithms, which shows that while they are far from optimal, they serve to demonstrate the practicality of this approach, and should thus motivate more refined algorithms.

5.1 The Lower Bound

In order to compute a lower bound for μ we need to find a local maximum of problem (9) as discussed in section 3. It turns out that this can be done efficiently by means of a power iteration. The iteration scheme usually converges fairly rapidly and each iteration of the scheme is very cheap, requiring only such operations as matrix-vector multiplications and vector inner products. The scheme tested here is a very simple power iteration, and does not converge on all problems, but in such cases one still obtains a candidate mixed perturbation from the iteration scheme. From this one can compute a lower bound (provided that the mixed μ problem contains some complex uncertainty) by simply wrapping in the real perturbations, and then evaluating the spectral radius of the associated complex μ problem, scaled by the candidate complex perturbations. The theoretical development of the power iteration, together with some aspects of its implementation, is fully described in [4] and we will not go into any of the details here.

5.2 The Upper Bound

Since the upper bound from theorem 2 is convex, one could tackle it using a variety of convex programming techniques. For instance we know that gradient search methods will lead us to the minimum eventually, although they may be slow (although the upper bound problem (10) is not in general differentiable if the maximum eigenvalue is repeated, it is possible to compute a generalized gradient which gives a descent direction). We would like to exploit the specific structure of the problem in order to speed up the computation. In particular we can reformulate the problem via the following theorem.

Theorem 4 ([22]) *Suppose we have a matrix $M \in \mathbf{C}^{n\times n}$ and a real scalar $\beta > 0$, then there exist matrices $D \in \mathcal{D}_{\mathcal{K}}, G \in \mathcal{G}_{\mathcal{K}}$ such that*

$$\overline{\lambda}\left(M^*DM + j(GM - M^*G) - \beta^2 D\right) \leq 0 \tag{13}$$

if and only if there exist matrices $\hat{D} \in \hat{\mathcal{D}}_{\mathcal{K}}, \hat{G} \in \hat{\mathcal{G}}_{\mathcal{K}}$ such that

$$\overline{\sigma}\left((I+\hat{G}^2)^{-\frac{1}{4}}\left(\frac{\hat{D}M\hat{D}^{-1}}{\beta} - j\hat{G}\right)(I+\hat{G}^2)^{-\frac{1}{4}}\right) \leq 1 \tag{14}$$

It is clear from this that as an alternative to carrying out the minimization in (10) we could compute the 'minimum' $\beta > 0$ such that

$$\inf_{\hat{D}\in\hat{\mathcal{D}}_{\mathcal{K}},\hat{G}\in\hat{\mathcal{G}}_{\mathcal{K}}} \overline{\sigma}\left((I+\hat{G}^2)^{-\frac{1}{4}}\left(\frac{\hat{D}M\hat{D}^{-1}}{\beta} - j\hat{G}\right)(I+\hat{G}^2)^{-\frac{1}{4}}\right) \leq 1 \tag{15}$$

Note that the theoretical equivalence of the two problems breaks down at $\beta = 0$ (and so for these cases strictly speaking there is no minimum β) but this presents no problem for a practical computation scheme since we merely quit if the upper bound falls below some prespecified tolerance (which can be arbitrarily small). Each of these two different formulations of the upper bound problem has its own advantages. The problem statement from (13) has the advantages that it is linear in the matrices D and G, and is convex (and hence one will not have problems associated with local minima). The problem statement from (14) has the advantages that one is trying to minimize the norm of a given matrix (which offers some numerical advantages), that D enters the problem exactly as in the standard complex μ upper bound, that G enters the problem in a balanced symmetric fashion, and that G is now a real diagonal matrix.

The upper bound algorithm implemented here works by initially tackling the problem in the form of (14). Here we can use some methods from the complex μ bounds, together with various other techniques, to obtain a fairly good estimates of $\hat{D}$,$\hat{G}$ and β. These are then converted into an initial guess for the problem in the form of (13) and the algorithm then proceeds to improve on these. This is covered in greater detail in [22].

5.3 Algorithm Performance

The main issues we are interested in, with regard to the algorithm performance, are the computational requirements of the algorithm, and the accuracy of the resulting bounds. We are interested in the typical performance of the algorithm, rather than the worst case (see the discussion in section 4.2), and so we examine these properties by running the algorithm repeatedly on a class of random problems, and collecting statistical data. The generation of test matrices, and the precise nature of the tests, are discussed in detail in [22].

One test performed was to examine the average computational requirements for the algorithm versus matrix size, and the results are shown in figure 2. The test problems had block structures consisting of all scalar uncertainties, with 90% of them chosen as real and the rest complex (although the results are typical of other block structures). The same data for the appropriate complex μ problem is shown for comparison. The results were obtained running Matlab on a Sparc 1 workstation, and it can be seen that we can reasonably expect to handle problems of size 10 in about 10 seconds, up to problems of size 50 in about 2-3 minutes.

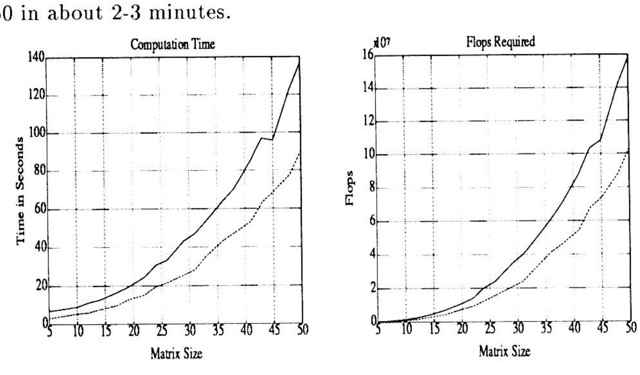

Figure 2: Typical computation requirements versus matrix size for mixed-μ problem (solid) and complex-μ problem (dashed).

It can also be seen that the (experimental) growth rate in computation time for the existing implementation is approximately n^2. This is probably an artifice of the implementation in Matlab, which is an interprative language. A more realistic measure of the computational growth rate is in terms of total floating point operations (flops). If this measure is adopted then it is seen that the (experimental) growth rate in flops is approximately n^3. In any case the algorithm growth rate appears reasonable whether measured in terms of time or flops required.

Another set of tests performed was aimed at evaluating the accuracy of the bounds. This time we compared the upper and lower mixed μ bounds, and also the mixed μ and complex μ upper bounds. The complex μ bounds were obtained by simply replacing all the real perturbations with complex ones, but without changing the matrix. Thus the complex upper bound is strictly larger than the mixed upper bound. Some results from these tests are shown in figure 3. It can be seen that the bounds are reasonably tight, even for the largest ($n = 50$) problems. Note also that we have a fairly wide spread of values for the gap between complex μ and mixed μ.

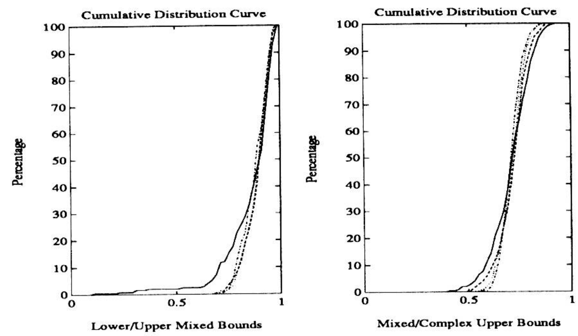

Figure 3: Typical ratios of mixed-μ lower to upper bounds, and mixed-μ to complex-μ upper bounds, for matrices of sizes 10 (solid), 20 (dashed), 30 (dotted), and 50 (dashdot).

5.4 Practical Examples

Whilst the results from the previous subsection are very encouraging, it is the algorithm's performance on actual engineering examples that is the real issue. A number of interesting applications of the software to problems arising from real physical systems have already been undertaken. The control design of a missile autopilot is considered in [23]. The software is used to examine the robustness (in performance) of the control design to perturbations in Mach number (real), angle of attack (real), and unmodeled dynamics (complex). This results in a mixed μ problem with two repeated scalar real parameters and three full complex blocks. The robust performance μ plots for this problem, and the associated complex μ problem (simply 'covering' the real uncertainties with complex ones), are shown in figure 4. It can be seen that

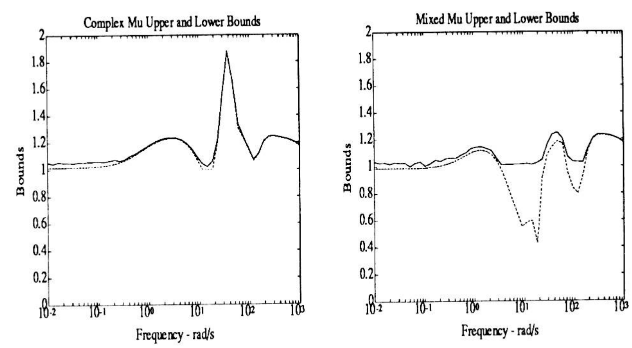

Figure 4: Robust performance μ plots for the missile autopilot problem.

the mixed μ bounds are quite different to the complex μ bounds, and the performance predictions for different controllers were also found to be different. It was found that the controller/performance predictions from the mixed μ bounds were borne out by the simulations.

Control of a flexible structure is considered in [24], and the robustness of the design is evaluated with respect to variations in the natural frequencies of the structural modes (real), as well as unmodeled dynamics (complex). This results in a mixed μ problem with five scalar real parameters and three full complex blocks. The robust performance μ plots for this problem, and the associated complex μ problem are shown in figure 5. Interestingly in this case, because of the way the

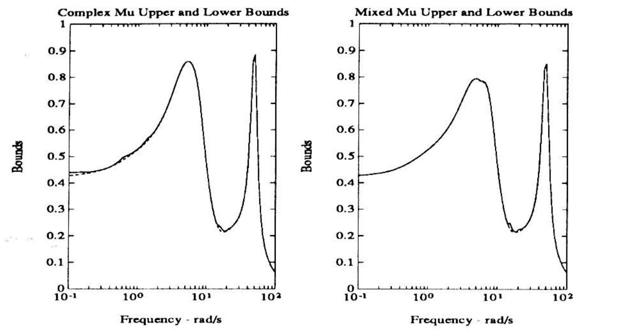

Figure 5: Robust performance μ plots for the flexible structure problem.

uncertainties entered the system, the mixed and complex bounds are seen to be very close. The control design predictions were verified in simulation and experiment. For these (and several other) examples the software worked well, providing tight bounds for the associated mixed μ problems.

5.5 The Next Generation of Algorithms

Note that in the previous subsections we encountered problems (both randomly generated and practically motivated) where the values of mixed μ and complex μ could be far apart or close together. Since it is hard to know a-priori which case one will encounter it is important to have good performance from *both* the upper and lower bound algorithms.

Recall that the lower bound takes the form of a power iteration, whose convergence is not guaranteed in all cases. It is well known that the convergence properties of standard eigenvalue and singular value power algorithms (which can be obtained as special cases of this algorithm) can be improved by inverse iteration, and similar adaptations to the mixed μ power algorithm are being investigated. Preliminary results have shown an improvement in the convergence properties, and it is hoped that further refinements will enable the convergence to a local maximum of (9) to be guaranteed [25].

The mixed μ upper bound (in the form of (10)) can be viewed as a special case of a class of LMI problems. The solution of LMI's is a subject of much research interest right now [26], since they appear in many control problems. This algorithm represents a first attempt at solving one particular LMI. As more refined algorithms for the solution of LMI's appear, then they can be used to improve the μ upper bound computation.

Note that all the previous tests were aimed at evaluating the typical performance of the algorithm, and it appears that the algorithm is performing well for most problems. This does not mean however that one cannot encounter mixed μ problems where the gap between the upper and lower bounds is large, and it can be seen from figure 3 that a few such cases were found. Furthermore it is possible in fact to construct matrices for which the gap between mixed μ and the (theoretical) upper bound from theorem 2 is arbitrarily large (regardless of the computation method). For these cases one must consider improving the bounds themselves. A promising approach is to use the existing bounds as part of a Branch and Bound scheme, which iteratively refines them. This is discussed in the following section.

6 Branch and Bound

The basic idea behind Branch and Bound schemes, in the context of the mixed μ problem, is that one has some algorithm for computing upper and lower bounds for mixed μ, but the bounds may be far apart. In order to refine the bounds one may 'chop' the subspace of real parameters into two subdomains and then evaluate the bounds on each subdomain (branch). One thus obtains upper and lower bounds for each subdomain of the partitioned space, and by choosing the largest of each of these we obtain new upper and lower bounds for the original problem. This process is then repeated as often as necessary, to refine the bounds as accurately as desired.

It is immediately apparent that one has the potential to encounter problems with exponential growth rates using this approach, and in fact we can construct problems which exhibit this behavior. This is not at all surprising since this scheme provides us with an algorithm to compute *guaranteed* bounds for mixed μ, which we know to be an NP hard problem. The real issue is whether or not we can produce a "practical" scheme, whose *typical* computation time is polynomial (despite the fact that the worst case computation time is exponential). This issue is pursued in the remainder of this section and is treated in greater depth in [27].

In order to address this problem one must consider the tradeoff between the computational cost versus accuracy of the bounds themselves, for any given sub-problem, and also the amount of computational cost one is prepared to pay in order to evaluate a good direction to chop the remaining subspace. A preliminary examination of these questions was carried out in [5], and the results strongly suggest that for a practical Branch and Bound scheme the methods for computing the bounds and the chopping criterion are absolutely critical to the performance on even medium sized problems. One is prepared to spend a high computational cost on both of these, provided it is still polynomial time, since one is potentially avoiding exponential time growth in the behavior of the Branch and Bound scheme (note that if any branch yields no improvement in the bounds then the subsequent computation can be doubled, since the same computation may have to be performed for each branch). Experimental results pertaining to these issues are presented in [5], and (in greater detail) in [27].

We would like to know what kind of performance level we can expect to achieve from a Branch and Bounds scheme. It is clear from the above results that we need to use sophisticated bounds (despite their computational expense) if we expect to get any kind of high performance scheme with reasonable computational requirements for fairly large problems. In order to examine the properties of such a scheme we implemented a Branch and Bounds scheme using the best currently available bounds (including a preliminary version of a new lower bound from [25]). This was then used to collect statistical data on the performance, by running the scheme repeatedly on essentially random problems (again, we are interested in typical, rather than worst case, performance for reasons discussed earlier).

This Branch and Bound scheme was used to compute upper and lower bounds for mixed μ problems on a class of random complex matrices. In order to attempt to make these problems representative of ones we might encounter in practice, they were constructed by first generating a random state space system, and then evaluating the transfer matrix at some frequency, usually placed approximately in the middle of the modes. The uncertainties consisted of m_r real scalars, and (approximately) $\frac{m_r}{4}$ complex scalars, where m_r ranged from 2 to 64. The results from one such batch of tests are shown in figure 6. There we have plotted the required number of branches versus number of real parameters for a series of Branch and Bound tests. Thus the curves represent required computational effort versus problem size. For each curve we have plotted the worst problem encountered from a pre-set number of runs, where for each problem the requirement for convergence was to reach a pre-specified tolerance between the upper and lower bounds, as labeled on the curve. Tolerances of 1%, 5%, 10% and 20% were considered, and for any problem the run was terminated if it failed to converge to the required tolerance within 100 branches (hence some of the curves terminate prematurely if the next problem size did not converge in time). Note that the graph is plotted on a log-linear scale, so that any straight line with non-zero slope represents an exponential growth rate.

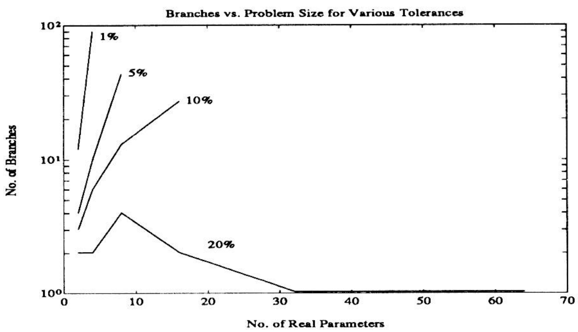

Figure 6: Branch and Bound computational requirements for varying degrees of required accuracy

It is clear from figure 6 that if the tolerance is set tight enough then the typical growth rate is unacceptable (see the 1% curve for example). Thus as the problem size increases the required computation quickly becomes impractical, and so we cannot expect to be able to achieve these tolerances. Note however that for the 20% curve the computational requirements remain modest even for the largest problems tested. Thus we can reasonably expect to be able to achieve this level of accuracy. Fortunately this degree of accuracy is quite sufficient for engineering purposes. It is important to keep in mind that our mathematical models are only approximations to real physical systems, and the uncertainties are intended to cover the deficiencies in our knowledge of that system. Thus it is somewhat naive to think that we can have precise knowledge of the uncertainty levels in real engineering problems.

It is interesting to note that for the 20% level the bounds were usually within tolerance at the first try, so that it was usually not necessary to branch at all. This suggests that if one is interested in solving fairly large problems, then one can only expect the Branch and Bound scheme to achieve a degree of accuracy that the bounds usually get anyway! Thus the Branch and Bound scheme is not being used as a general computation scheme per se, but only to fix the occasional problems for which the bounds are poor, and for these problems to achieve the degree of accuracy which the bounds typically get. This reinforces the results in [5] and emphasizes the necessity for good bounds.

To further illustrate this point consider the plot in figure 7. This plot shows a mixed μ computation for a problem with 4 real and 1 complex scalar uncertainties, where the initial bounds were quite poor (85% relative gap as opposed to a typical level of less than 20%). We have plotted the current upper and lower bounds for the problem versus the number of branches, so that the progress of the Branch and Bound scheme on the problem can be seen. It is readily apparent that initially quite rapid progress is made so that in only 29 steps the new bounds are within 20%. However it is also apparent that the progress of the scheme slows quite dramatically after this point, so that achieving greater levels of accuracy requires substantially more computational effort, and rapidly becomes impractical.

The study of the use of cheap versus sophisticated bounds made in [5] employed Branch and Bound schemes using methods from the extreme ends of the spectrum. In other words the best currently available bounds (which are quite computationally intensive) were compared to some very crude bounds (which are cheap to compute), when employed in a Branch and Bounds scheme. The overwhelming conclusion, as discussed earlier, was in favor of the more sophisticated bounds. In order to examine this question more deeply we compared the use of the best bounds we had available to the next best we could use in a Branch and Bound scheme. The results are plotted in figure 8. The left hand

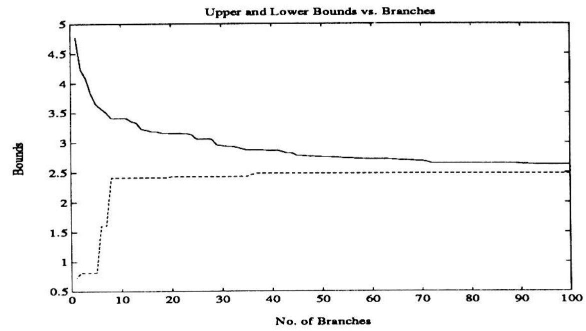

Figure 7: Progress of Branch and Bound for a hard problem

plot was generated using a Branch and Bound scheme employing the bounds previously discussed. We will refer to this as scheme A. The results in the right hand plot came from a scheme employing the same lower bound, and an upper bound obtained by covering the real parameters with complex ones, and then evaluating the complex μ upper bound. Essentially this amounts to enforcing the choice $G = 0_n$ in (10), and so this bound is a little cheaper to compute, but not quite as good, as (10). We will refer to this scheme as scheme B. The results are shown for a series of mixed μ problems with 4 real and 1 complex scalar uncertainties. We have plotted the relative gap between the bounds versus the number of branches on a log-log scale. Thus we see the progress of the Branch and Bound schemes with time, and for clarity a number of tolerance levels between the bounds are labeled. Note that for scheme A all the problems reached tolerances of 10% within 6 branches whereas for scheme B several problems failed to reach 10% within the allowed 100 branches. Furthermore the typical performance for scheme B can be clearly seen to be inferior to scheme A. It is clear that even this level of reduction in the quality of the bounds markedly affects the performance of the overall scheme. Thus we are led to conclude once more that the performance of the bounds is crucial to the performance of the overall scheme, and that for a high performance Branch and Bound scheme it is important to use the best bounds available.

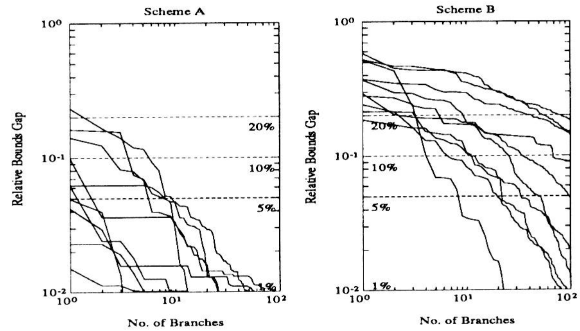

Figure 8: Comparison of Branch and Bound schemes

7 Mixed μ Synthesis

The problem of synthesising a controller which is (optimally) robust to structured mixed uncertainty is very difficult, since the associated optimization problem is not convex. Furthermore it seems intuitively clear that the synthesis problem is at least as hard as the analysis problem, which is known to be NP complete. Some exact solutions have been presented for special cases of the synthesis problem (see [28] for example, which reduces the "rank one" μ synthesis problem to a convex optimization problem), but these are all cases for which the analysis problem also simplifies considerably. As yet there is no globally optimal solution to the general synthesis problem (even in the purely complex case), and no indication that one will be forthcoming in the foreseeable future.

Nevertheless the (complex) μ-synthesis procedure first outlined in [29] has been successfully applied to a large number of engineering problems (see [24] for example). This procedure involves a "D-K iteration" between computing the μ upper bound, and solving for an H_∞ (sub) optimal controller (both of which are convex problems). This procedure, which was developed for μ problems involving *only* complex blocks, does not guarantee to find the globally μ-optimal controller, but has often been found to work well in practice. In light of this it seems that a reasonable approach to the mixed μ synthesis problem is to attempt to extend the above procedure to the mixed case, by exploiting the new analysis tools for the mixed μ upper bound described in the preceding sections. This is a direction of current research.

8 Acknowledgements

The authors would particularly like to thank Jorge Tierno, who assisted in generating the results in section 6, using a new lower bound algorithm he developed. We would also like to thank Andy Packard, Michael Fan, and Stephen Boyd for useful discussions, and Gary Balas who provided us with some engineering problems in section 5.4.

References

[1] G. Balas, J. Doyle, K. Glover, A. Packard, and R. Smith, "The μ analysis and synthesis toolbox." MathWorks and MUSYN, 1991.

[2] J. Doyle, A. Packard, and K. Zhou, "Review of LFTs, LMIs and μ," in *Proceedings of the 30th Conference on Decision and Control*, pp. 1227–1232, 1991.

[3] M. K. H. Fan, A. L. Tits, and J. C. Doyle, "Robustness in the presence of mixed parametric uncertainty and unmodeled dynamics," *IEEE Transactions on Automatic Control*, vol. AC-36, pp. 25–38, 1991.

[4] P. M. Young and J. C. Doyle, "Computation of μ with real and complex uncertainties," in *Proceedings of the 29th Conference on Decision and Control*, pp. 1230–1235, IEEE, 1990.

[5] P. M. Young, M. P. Newlin, and J. C. Doyle, "μ analysis with real parametric uncertainty," in *Proceedings of the 30th Conference on Decision and Control*, pp. 1251–1256, IEEE, 1991.

[6] V. Balakrishnan, S. Boyd, and S. Balemi, "Branch and Bound algorithm for computing the minimum stability degree of parameter-dependent linear systems," tech. rep., Information Systems laboratory, Department of Electrical Engineering, Stanford University, 1991.

[7] A. Sideris and R. S. Sánchez Peña, "Fast computation of the multivariable stability margin for real interrelated uncertain parameters," *IEEE Transactions on Automatic Control,*, vol. 34, pp. 1272–1276, Dec. 1989.

[8] R. R. E. de Gaston and M. G. Safanov, "Exact calculation of the multiloop stability margin," *IEEE Transactions on Automatic Control*, vol. 33, pp. 156–171, 1988.

[9] B. Barmish, P. Khargonekar, Z. Shi, and R. Tempo, "Robustness margin need not be a continuous function of the problem data," *Systems & Control Letters*, vol. 15, pp. 91–98, 1989.

[10] J. Rohn and S. Poljak, "Radius of nonsingularity." to appear in Mathematics of Control, Signals and Systems.

[11] J. Demmel, "The componentwise distance to the nearest singular matrix." to appear in SIAM Journal on Matrix Analysis and Applications.

[12] R. D. Braatz, P. M. Young, J. C. Doyle, and M. Morari, "Computational complexity of mu calculation." submitted to IEEE Transactions on Automatic Control.

[13] V. L. Kharitonov, "Asymptotic stability of an equilibrium position of a family of systems of linear differential equations," *Differential Equations*, vol. 14, pp. 1483–1485, 1979.

[14] J. Doyle, "Analysis of feedback systems with structured uncertainty," *IEE Proceedings, Part D*, vol. 129, pp. 242–250, Nov. 1982.

[15] A. Packard and J. C. Doyle, "Structured singular value with repeated scalar blocks," in *Proceedings of the American Control Conference*, pp. 1213–1218, 1988.

[16] A. K. Packard and J. C. Doyle, "The complex structured singular value." to appear in Automatica.

[17] A. K. Packard and P. Pandey, "Continuity properties of the real/complex structured singular value." submitted to IEEE Transactions on Automatic Control.

[18] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP Completeness*. New York: W. H. Freeman, 1979.

[19] A. C. Bartlett, C. V. Hollot, and H. Lin, "Root locations of an entire polytope of polynomials: It suffices to check the edges," in *Mathematics of Control, Signals and Systems*, Springer Verlag, 1988.

[20] J. Chen, M. K. H. Fan, and C. N. Nett, "The structured singular value and stability of uncertain polynomials: A missing link," *Control of Systems with Inexact Dynamic Models, ASME*, pp. 15–23, 1991.

[21] P. M. Young, "The rank one mixed μ problem and "Kharitonov-type" analysis." Manuscript in Preparation.

[22] P. M. Young, M. P. Newlin, and J. C. Doyle, "Practical computation of the mixed μ problem," in *Proceedings of the American Control Conference*, pp. 2190–2194, 1992.

[23] G. J. Balas and A. K. Packard, "Development and application of time-varying μ-synthesis techniques for control design of missile autopilots." John Hopkins Applied Physics Laboratories, Final Report, January, 1992.

[24] G. J. Balas, P. M. Young, and J. C. Doyle, "μ based control design as applied to a large space structure: Control design for the minimast facility." NASA CSI/GI final report, June 1992.

[25] J. E. Tierno and P. M. Young, "An improved μ lower bound via adaptive power iteration." to appear in 31st Conference on Decision and Control, 1992.

[26] C. Beck and J. C. Doyle, "Mixed μ upper bound computation using LMI optimization." to appear in 31st Conference on Decision and Control, 1992.

[27] M. P. Newlin and P. M. Young, "Mixed μ problems and Branch and Bound techniques." to appear in 31st Conference on Decision and Control, 1992.

[28] A. Rantzer and A. Megretsky, "A convex parameterization of robustly stabilizing controllers," tech. rep., Department of Mathematics, Royal Institute of Technology, Stockholm, Sweden, 1992.

[29] J. C. Doyle, "Structured uncertainty in control system design," in *Proceedings of the 24th Conference on Decision and Control*, pp. 260–265, IEEE, 1985.