A Gain-Based Lower Bound Algorithm for Real and Mixed μ Problems

Pete Seiler *, Gary Balas †, and Andrew Packard ‡

Abstract

In this paper we present a new lower bound algorithm for real and mixed μ problems. The basic idea of this algorithm is to use a related worst-case gain problem to compute the real blocks and, if the block structure is mixed, the standard power iteration to compute the complex blocks. Initial numerical tests indicate that the algorithm is fast and provides good bounds for both real and mixed μ problems of small to moderate size.

1 Introduction

The structured singular value μ, introduced by Doyle [8], can be used to analyze the robustness of linear systems subject to structured uncertainty. It is assumed that the reader is familiar with the engineering motivation for μ (see [1, 19, 11] and references therein for some discussion). It is known that computing μ is NP Hard [4, 7] and for the pure real case, even computing upper bounds with certain desirable properties is NP Hard [12]. Thus there has been extensive research into computational algorithms which are fast and provide good lower/upper bounds for most problems of engineering interest. This paper will discuss an algorithm to compute lower bounds for μ.

For the pure complex μ problem, the power iteration [19, 20] provides good lower bounds and it is quite fast since it relies only on matrix-vector products. The power iteration was extended to mixed μ problems in [26, 27, 29]. Unfortunately this algorithm may fail to converge; a problem which is more common for purely real uncertainty structures [15, 24, 25]. There has been extensive research on alternative lower bound algorithms to address these issues [3, 5, 6, 9, 13, 14, 15, 16, 17, 22, 23, 24].

For the pure real case, there are fundamental difficulties including the fact that real μ can be a discontinous function of the problem data [2, 21]. However, there are real μ problems of engineering interest which are well-posed. Most existing algorithms to solve these problems have a computational cost which grows exponentially with the problem size [5, 6, 9, 22] and hence they are only suitable for small numbers of real parameters. A less computationally intensive approach is to regularize the problem and use the standard mixed μ power iteration. The regularization is typically accomplished by adding a small amount of complex uncertainty to each uncertain real parameter[21].

This paper describes a polynomial-time lower bound algorithm which can be applied to both pure real and mixed μ problems. We refer to this lower bound algorithm as the Gain-Based Algorithm (GBA). The basic idea of the GBA is to use a related worst-case gain problem to compute the real blocks of the perturbation. For mixed μ problems, the standard power iteration is then used to compute the complex blocks since it is fast and has good convergence characteristics for complex μ problems. The GBA uses the wrap-in reals idea which exists in [24, 15] but with two main distinctions. First, the use of the worst-case gain problem to compute the real blocks is a new approach. Second, we do not wrap in the real blocks within the power iteration. Instead, the real block is computed from scratch for each step of the GBA and then held fixed throughout the complex power iteration.

The GBA seems to have better convergence properties than the standard power iteration on real μ and certain classes mixed μ problems. We believe this is because the GBA implicitly regularizes the problem by adding a small complex scalar to one entry of the data matrix (as described in Section 3.1). Moreover, the GBA makes several attempts (N_try) to find a good lower bound and N_{try} can be used to trade off computation time with the quality of the lower bound.

2 Notation

Let $M \in \mathbb{C}^{(n+m)\times(n+m)}$ and $\Delta \in \mathbb{C}^{n\times n}$ be given and partition $M := \left[\begin{smallmatrix} M_{11} & M_{12} \\ M_{21} & M_{22}\end{smallmatrix}\right]$ with $M_{11} \in \mathbb{C}^{n\times n}$ and $M_{22} \in \mathbb{C}^{m\times m}$. If $I - M_{11}\Delta$ is invertible, then we define $F_u(M,\Delta)$ as the linear fractional transformation obtained by closing Δ around the upper channels of M:

$$F_u(M,\Delta) := M_{22} + M_{21}\Delta\left(I - M_{11}\Delta\right)^{-1} M_{12}$$

We also let $\bar{\sigma}(M)$ and $\underline{\sigma}(M)$ denote the maximum and minimum singular values of the matrix M, respectively.

The notation used in this paper for the structured singular value, i.e. μ, is standard. We'll consider block

*Honeywell Labs, peter.j.seiler@honeywell.com

†AEM, University of Minnesota, balas@umn.edu

‡ME, University of California, Berkeley, pack@me.berkeley.edu

structures consisting of r repeated real scalar blocks, c repeated complex scalar blocks, and f square full complex blocks. The restriction to square full blocks is for notational simplicity and the lower bound algorithm described in this paper can be extended to non-square full blocks. Given positive integers $k_1, k_2, \ldots, k_{r+c+f}$ define the following sets of block structured matrices:

$$\begin{aligned}
\mathbf{\Delta}_R &:= \{\Delta = blockdiag(\delta_1 I_{k_1}, \ldots, \delta_r I_{k_r}) \ : \ \delta_i \in \mathbb{R}\} \\
\mathbf{\Delta}_C &:= \{\Delta = blockdiag(\delta_1 I_{k_{r+1}}, \ldots, \delta_c I_{k_{r+c}}, \Delta_1, \ldots, \Delta_f) \ : \\
&\qquad \delta_i \in \mathbb{C}, \ \Delta_i \in \mathbb{C}^{k_{r+c+i} \times k_{r+c+i}}\} \\
\mathbf{\Delta} &:= \{\Delta = blockdiag(\Delta_R, \Delta_C) \ : \ \Delta_R \in \mathbf{\Delta}_R, \ \Delta_C \in \mathbf{\Delta}_C\}
\end{aligned}$$

$\mathbf{\Delta}_R$, $\mathbf{\Delta}_C$, and $\mathbf{\Delta}$ are pure real, pure complex, and mixed real/complex block structures, respectively. The matrices in $\mathbf{\Delta}_R$, $\mathbf{\Delta}_C$, and $\mathbf{\Delta}$ have respective dimensions $n_R \times n_R$, $n_C \times n_C$, and $n \times n$ where $n_R := \sum_{i=1}^{r} k_i$, $n_C := \sum_{i=1}^{c+f} k_{r+i}$, and $n := n_R + n_C$.

The next definition, originally given by Doyle [8] for the pure complex case, is for μ in terms of the block structure defined by the set $\mathbf{\Delta}$. However, it also applies to other block structures such as the pure real ($\mu_{\mathbf{\Delta}_R}$) and pure complex ($\mu_{\mathbf{\Delta}_C}$) cases.

Definition 1 *[8] The structured singular value of $M \in \mathbb{C}^{n \times n}$ with respect to $\mathbf{\Delta}$, denoted $\mu_{\mathbf{\Delta}}(M)$, is defined as*

$$\mu_{\mathbf{\Delta}}(M) := \left(\min_{\Delta \in \mathbf{\Delta}} \{\bar{\sigma}(\Delta) \ : \ \det(I - M\Delta) = 0\} \right)^{-1} \quad (1)$$

if $\exists \Delta \in \mathbf{\Delta}$ such that $\det(I - M\Delta) = 0$ and otherwise $\mu_{\mathbf{\Delta}}(M) := 0$.

3 Gain-Based Algorithm (GBA)

In this section, we first introduce the GBA for pure real μ problems (Section 3.1) and then for mixed-μ problems (Section 3.2). The GBA proposed in Section 3.2 defaults to the standard power iteration for pure complex μ problems and to the GBA described in Section 3.1 for pure real μ problems.

3.1 Real μ GBA

Assume $M_R \in \mathbb{C}^{n_R \times n_R}$. In this section we address the problem of computing lower bounds for the pure real problem $\mu_{\mathbf{\Delta}_R}(M_R)$. It follows directly from the definition of μ that any $\Delta_R \in \mathbf{\Delta}_R$ which satisfies $\det(I - M\Delta_R) = 0$ yields a lower bound:

$$\Delta_R \in \mathbf{\Delta}_R, \ \det(I - M\Delta_R) = 0 \Rightarrow \frac{1}{\bar{\sigma}(\Delta_R)} \leq \mu_{\mathbf{\Delta}_R}(M_R)$$

Thus lower bounds for $\mu_{\mathbf{\Delta}_R}(M_R)$ can be computed by searching for a $\Delta_R \in \mathbf{\Delta}_R$ which causes the following

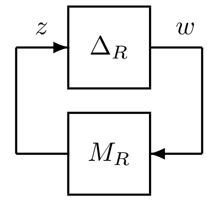

Figure 1: M_R-Δ_R Loop

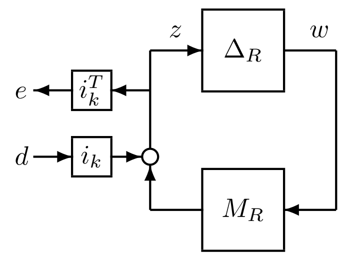

Figure 2: M_R-Δ_R Loop With d-to-e Channels

equations to have a non-zero solution: $z = M_R w$ and $w = \Delta_R z$. These equations are represented by the $M_R - \Delta_R$ loop in Figure 1. The basic intuition for the GBA is to recast this problem into a related worst-case disturbance-to-error problem, shown in Figure 2. In this figure, we have inserted a disturbance at the output of M_R and pulled off an error at the input to Δ_R. i_k denotes the k^{th} standard basis vector in $\mathbb{R}^{n_R}$ and d/e denote scalar disturbance/error signals. Note that the scalar disturbance is only inserted in the k^{th} channel and the scalar error is only pulled off the k^{th} channel.

If $\det(I - M_R\Delta_R) \neq 0$ then the system with input d and output e can be written as:

$$\begin{aligned}
e &= F_u(\tilde{M}_R, \Delta_R) d \\
&= \left[1 + i_k^T M_R \Delta_R (I - M_R\Delta_R)^{-1} i_k\right] d \\
&= \left[i_k^T (I - M_R\Delta_R)^{-1} i_k\right] d
\end{aligned}$$

where $\tilde{M}_R := \begin{bmatrix} M_R & i_k \\ i_k^T M_R & 1 \end{bmatrix}$. If we find a $\Delta_R \in \mathbf{\Delta}_R$ such that the gain from d to e is large then $I - M_R\Delta_R$ will be close to singularity. The precise statement is that $\bar{\sigma}\left(i_k^T (I - M_R\Delta_R)^{-1} i_k\right) \geq \gamma$ implies $\underline{\sigma}(I - M_R\Delta_R) \leq \frac{1}{\gamma}$. Moreover, if we find $\Delta_R \in \mathbf{\Delta}_R$ such that the gain from d to e is infinite then Δ_R satisfies $\det(I - M_R\Delta_R) = 0$ and hence $\frac{1}{\bar{\sigma}(\Delta_R)}$ is a true lower bound on $\mu_{\mathbf{\Delta}_R}(M_R)$. For numerical reasons, we'll restrict our search for Δ_R's which yield a large gain from d to e.

Given an estimated value for the lower bound, lb_{try}, we'd like to solve the following problem:

$$\max_{\Delta_R \in \mathbf{\Delta}_R, \bar{\sigma}(\Delta_R) \leq 1/lb_{try}} \bar{\sigma}\left(F_u(\tilde{M}_R, \Delta_R)\right) \quad (2)$$

If the maximum gain is sufficiently large then we'll use the maximizer, $\Delta_{R,opt}$, to compute a lower bound on $\mu_{\mathbf{\Delta}_R}$. Restricting the search to $\bar{\sigma}(\Delta_R) \leq 1/lb_{try}$ ensures that $\Delta_{R,opt}$ will yield a lower bound which is $\geq lb_{try}$.

Equation 2 is in the form of a worst case performance problem. The lower bound algorithm introduced by Packard, et. al. [18] for worst-case performance assessment will be used to "solve" this problem. The algorithm is an ascent method which returns a lower bound on the maximum. Specifically, Packard, et. al. use an exact maximization for the single parameter problem ($n_R = 1$) and an iterative coordinatewise maximization for the general case ($n_R > 1$). The exact maximization along each coordinate is computed by mimicking the Hamiltonian methods for state-space H_∞ norm calculation. The lower bound algorithm in [18] is presented for $\bar{\sigma}(\Delta_R) \leq 1$. This is without loss of generality since the perturbation can be normalized as shown in Section 3.3 of [18].

The Gain-Based Algorithm (GBA) to compute real μ lower bounds is presented in Table 1. The GBA assumes that upper/lower bounds (ub/lb) on $\mu_{\mathbf{\Delta}_R}(M_R)$ as well as a perturbation (Δ_R) achieving lb are given. The lower bound can simply be initialized to $lb = 0$ and $\Delta_R = 0_{n_R}$. Alternatively, the lower bound and perturbation from the standard power iteration can be used. However, the standard power iteration rarely converges for pure real μ problems and our experience is that it is not worth the computational effort. The upper bound can be computed via standard methods, e.g. via the LMI [10] or Balanced [28, 29] form. The GBA attempts to solve Equation 2 up to N_{try} times using the worst case gain lower bound by Packard, et. al. [18]. Since the value of $\mu_{\mathbf{\Delta}_R}(M_R)$ is unknown, lb_{try} is adaptively chosen at each iteration based on the success or failure of the worst-case gain search. The upper/lower bounds are used to obtain good estimates for lb_{try}. The choice of the channel, k, to insert d and pull off e is cycled through all possible choices. The performance of the GBA is discussed in Section 4.

The GBA returns perturbations which satisfy $|\det(I - M_R\Delta_R)| < tol_{real}$, i.e. they do not strictly cause singularity. We now show that the use of the worst-case gain problem can be viewed as a form of regularizing the problem. The Δ_R returned by the GBA achieved a large gain, $\bar{\sigma}\left(F_u(\tilde{M}_R, \Delta_R)\right) \geq \gamma_{large}$, on one of the output channels of M_R, denoted k_{bad}. Therefore there exists $\delta \in \mathbb{C}$ with $|\delta| \leq 1/\gamma_{large}$ such that $\det(I - \tilde{M}_R\tilde{\Delta}_R) = 0$ where $\tilde{\Delta}_R := blockdiag(\Delta_R, \delta)$. This fact can be simplified via a nonsingular transformation:

$$\begin{aligned}\det(I - \tilde{M}_R\tilde{\Delta}_R) = 0 &\Leftrightarrow \det\left(\begin{bmatrix} I & 0 \\ i_k^T & -1\end{bmatrix}(I - \tilde{M}_R\tilde{\Delta}_R)\right) = 0 \\ &\Leftrightarrow \det\left(\begin{bmatrix} I-M_R\Delta_R & -\delta i_k \\ i_k^T & -1\end{bmatrix}\right) = 0 \\ &\Leftrightarrow \det\left(I - M_R\Delta_R - \delta i_k i_k^T\right) = 0\end{aligned}$$

Thus a small complex perturbation to the (k,k) diagonal entry of magnitude $\leq 1/\gamma_{large}$ will cause $I - M_R\Delta_R$ to become singular. The complex perturbation can be interpreted as a regularization to the problem and the algorithm attempts to minimize the amount of complexity needed to achieve singularity.

Given : $M_R \in \mathbb{C}^{n_R \times n_R},\ \Delta_R \in \mathbf{\Delta}_R,\ lb,\ ub$
Initialize : $lb_{fac} = 3/4,\ cnt = 1$
while $cnt \leq N_{try}$ AND $lb < ub \cdot tol_{stop}$
  $lb_{try} = lb + lb_{fac} \cdot (ub - lb)$
  $k := \mod(cnt-1, n_R) + 1$[a]
  $\tilde{M}_R := \begin{bmatrix} M_R & i_k \\ i_k^T M_R & 1 \end{bmatrix}$
  $\Delta_{R,try} := \arg\max_{\Delta_R \in \mathbf{\Delta}_R,\ \bar{\sigma}(\Delta_R) \leq 1/lb_{try}} \|F_u(\tilde{M}_R, \Delta_R)\|$[b]
  if $|\det(I - M_R\Delta_{R,try})| < tol_{real}$[c]
    $lb = \frac{1}{\bar{\sigma}(\Delta_{R,try})}$
    $\Delta_R = \Delta_{R,try}$
    $lb_{fac} := 1/2$
  else
    $lb_{fac} := \max(1/32, lb_{fac}/2)$
  end
  $cnt = cnt + 1$
end

Table 1: The GBA for Real μ Lower Bounds

[a] mod denotes the modulus after division. Thus, k counts up from 1 to n_R and then rolls back to 1.

[b] The algorithm in [18] is used to solve this maximization. $\Delta_{R,try}$ need not be the maximizer, i.e. it is sufficient for $\Delta_{R,try}$ to yield a lower bound on the maximal cost.

[c] The minimum singular value or inverse condition number can also be used to check for singularity of $I - M_R\Delta_{R,try}$.

3.2 Mixed μ GBA

Assume $M \in \mathbb{C}^{n \times n}$ and partition M conformably with the real and complex blocks of $\mathbf{\Delta}$, $M := \begin{bmatrix} M_R & M_{RC} \\ M_{CR} & M_C \end{bmatrix}$ where $M_R \in \mathbb{C}^{n_R \times n_R}$ and $M_C \in \mathbb{C}^{n_C \times n_C}$. In this section we address the problem of computing lower bounds for the mixed μ problem $\mu_{\mathbf{\Delta}}(M)$. The GBA for mixed μ problems is presented in Table 2. The GBA makes up to N_{try} attempts to find a good lower bound with lb_{try} being updated adaptively. For mixed-μ problems, the power iteration may not converge but it always returns a perturbation which causes $\det(I - M\Delta) = 0$. It is worth the computational effort to run the power iteration first and use the lower bound and perturbation it returns to initialize lb and $\Delta \in \mathbf{\Delta}$ in the GBA for mixed μ problems.

For each attempt of the GBA, the related worst-case gain problem (Figure 2) is used to compute the real block of the perturbation. If the real block alone causes singularity then it is used to compute a valid lower bound. We can always use the complex blocks to ensure $\det(I - M\Delta) = 0$ within numerical tolerance. Hence

```
Given : M := [ M_R  M_RC ; M_CR  M_C ] ∈ C^{n×n},  Δ ∈ 𝚫,  lb,  ub
Initialize : lb_fac = 3/4,  cnt = 1
while cnt ≤ N_try AND lb < ub · tol_stop
    lb_try = lb + lb_fac · (ub − lb)
    k :=  mod (cnt − 1, n_R) + 1
    M̃_R := [ M_R  i_k ; i_k^T M_R  1 ]
    Δ_{R,try} := argmax_{Δ_R ∈ 𝚫_R, σ̄(Δ_R) ≤ 1/lb_try} ‖F_u(M̃_R, Δ_R)‖
    if |det(I − M_R Δ_{R,try})| < tol_complex
        lb = 1/σ̄(Δ_{R,try})
        Δ = blockdiag(Δ_{R,try}, 0)
        lb_fac := 1/2
    else
        M̃_C := F_u(M, Δ_R)
        Power Iteration on M̃_C to find Δ_{C,try} ∈ 𝚫_C^a
        Δ_try := blockdiag(Δ_{R,try}, Δ_{C,try})
        if |det(I − M Δ_try)| < tol_complex AND 1/σ̄(Δ_try) ≥ lb
            lb = 1/σ̄(Δ_try)
            Δ = Δ_try
            lb_fac := 1/2
        else
            lb_fac := max(1/32, lb_fac/2)
        end
    end
    cnt = cnt + 1
end
```

Table 2: The GBA for Mixed μ Lower Bounds

[a] See [19, 20] for the details on the power iteration.

we can set $tol_{complex}$ to be a small factor above numerical tolerance, e.g. 100∗`eps`. The main point is that the mixed μ GBA, unlike the real-μ GBA presented in the previous section, returns perturbations which strictly cause $\det(I - M\Delta) = 0$ within numerical errors.

If the real block does not cause singularity, then it is wrapped into M to form $\tilde{M}_C$. The standard power iteration [19, 20] is run on $\tilde{M}_C$ to compute the complex block, $\Delta_C \in \mathbf{\Delta}_C$. A perturbation, $\Delta \in \mathbf{\Delta}$, is then formed from the real/complex blocks and stored if it increases the current lower bound. Even though lb_{try} is chosen to be strictly larger than the current lower bound, the perturbation Δ might not increase the lower bound. In particular, if the norm of the complex block is too large ($\bar{\sigma}(\Delta_C) > 1/lb$), then the perturbation will not improve the lower bound. It seems that the adaptive selection of lb_{try} naturally balances the norms of the real and complex blocks. If lb_{try} is too large, then we will be overly restricting our search for Δ_R. As a result the gain of $F_u(M_R, \Delta_R)$ may not be very large and hence it will take a larger norm complex block to make the loop singular. If this larger norm complex block causes Δ to achieve a lower bound less than lb_{try} then lb_{try} will be decreased on the next iteration. The performance of the mixed μ GBA is discussed in the next section.

4 Numerical Results

4.1 Real μ GBA

This section gives an example of the GBA performance on a Real μ problem. Let $M_R(s) := \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$ where the state matrices are given in Appendix A. This data is from a flight control robustness analysis done at the Honeywell Labs. Consider a block structure consisting of four 1x1 real uncertainties ($r = 4$ and $k_1 = k_2 = k_3 = k_4 = 1$). $M_R(s)$ was evaluated at 500 frequency points logarithmically spaced between 10 and 1e8 rad/sec. Figure 3 shows the lower and upper bounds computed with the GBA and LMI method, respectively. The GBA parameters were specified as: $N_{try} = 30$, $tol_{stop} = 0.97$, and $tol_{real} = 1e-7$. Also, we stopped the worst-case algorithm if it found a perturbation causing the gain from d to e to exceed $1e12$. The total time to compute upper/lower bounds at all 500 frequency points was 82.3 seconds on a 2GHz processor. The GBA accounted for roughly 65 seconds of this total time. We plugged the perturbations returned by the GBA into $F_u(M_R(s), \Delta_R)$ and they all gave $|\det(I - M_R(j\omega_k)\Delta_k)| < tol_{real} = 1e{-}7$. Most (477 out of 500 points) gave a determinant less than 1e-10. The worst case lower bound is 1.61 achieved at a frequency $\omega_k = 177.2$ rad/sec. The perturbation returned by the GBA at this frequency is $\Delta_R = diag(-.358, .620, -.621, .621)$. This perturbation places two poles of $F_u(M_R(s), \Delta_R)$ at $-5.63e{-}9 \pm 177.2j$. For comparison, it took 55.3 seconds to compute the LMI upper bound and the lower bound from the standard power iteration. The power iteration took roughly 33 seconds of this total time but the iteration converged on only 6 out of 500 frequency points. We have also tried the GBA on real μ examples in the literature as well as randomly generated matrices and it seems to have good convergence properties.

4.2 Mixed μ GBA

The performance of the standard power iteration (SPI) is satisfactory on most randomly generated matrices so it is common to test lower bound algorithms on a class of "hard" mixed μ problems for which it is known *a priori* that $\mu = 1$. The algorithm for generating these "hard" problems is described in [25, 29].

We tested the GBA on this class of problems with a block structure consisting of r 1x1 real uncertainties, two 1x1 complex uncertainties, and a single 2x2 complex full block. This block structure was also used to test lower bound algorithms in [15, 24]. Figure 4 shows the dis-

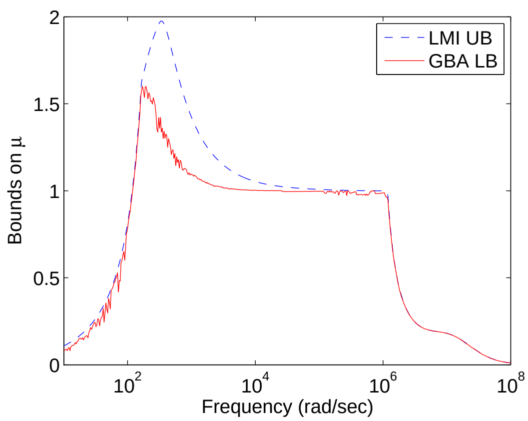

Figure 3: Upper/Lower Bounds for Real-μ Example

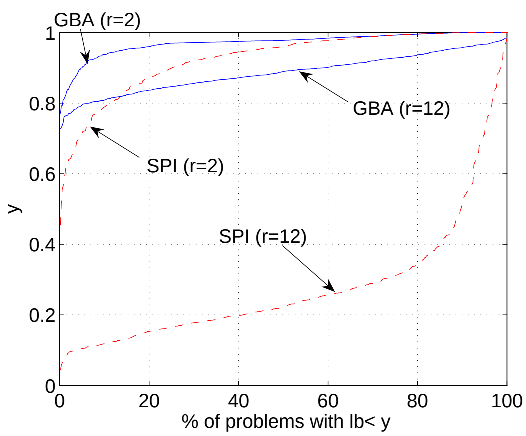

Figure 4: Lower Bounds from Standard Power Iteration and Gain-Based Algorithm

tribution of the GBA lower bounds on 500 problems for $N_{try} = 25$ with the number of real uncertainties growing from 2 to 12. The distribution of lower bounds from the SPI is also shown for the case of two and twelve real uncertainties. The performance of the GBA is much better than the SPI and it degrades gracefully as a function r. These results are also comparable to the Combined Power Algorithm presented in [15]. The GBA yields bounds between 0.8 and 0.95 more often than the Combined Power Algorithm, i.e. it yields both very poor and very good bounds less often than the Combined Power Algorithm. The GBA lower bounds can be improved at the expense of additional computation by increasing either the number of tries (N_{try}) and/or the stop tolerance (tol_{stop}).

Figure 5 shows the increase in computation time for the GBA as a function of the number of real blocks averaged over 100 problems. There is some overhead for small problems, but the curve is basically a straight line on the log-log plot for $r \geq 20$. This implies that the computational cost grows polynomially with the number of real blocks for this block structure.

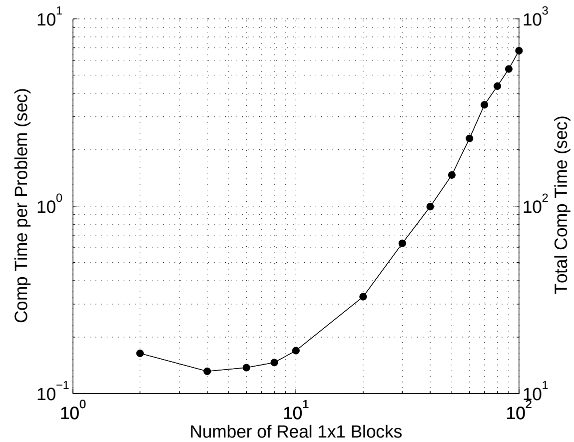

Figure 5: Computation Time for Gain-Based Algorithm

5 Conclusions

We presented a new lower bound algorithm for real and mixed μ problems. The algorithm uses a related worst-case gain problem to compute the real blocks and, for mixed μ problems, uses the standard power iteration to compute the complex blocks. Initial numerical tests indicate that this algorithm is fast and provides good bounds. The algorithm could be modified by incorporating branch and bound methods. The algorithm could also use exact real μ methods for small problems and switch over to the worst-case gain search for larger problems.

A Data for Real μ Example

$$A(1:8,1:4) := \begin{bmatrix} -6.61e7 & 701.1 & -23.66 & -0.4465 \\ 6.695e8 & -2.174e5 & 0 & 0 \\ 3.595e7 & 2.963e4 & -1000 & 0 \\ 3.595e7 & 2.963e4 & -1000 & -18.87 \\ 3.595e7 & 2.963e4 & -1000 & -18.87 \\ 4451 & 3.669 & -0.1238 & -0.002336 \\ 44.51 & 0.03669 & -0.001238 & -2.336e{-5} \\ 8.506e5 & 701.1 & -23.66 & -0.4465 \end{bmatrix}$$

$$A(1:8,5:8) := \begin{bmatrix} 4.507e6 & 4.563e8 & -8.494e11 & -1e4 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -9.524e6 & 0 & 0 & 0 \\ 2.358e4 & -2.388e4 & 0 & 0 \\ 235.8 & 2.388e4 & -2.222e7 & 0 \\ 4.507e6 & 4.563e8 & -8.494e11 & -1e4 \end{bmatrix}$$

$$B := \begin{bmatrix} -0.02366 & -0.02366 & -1 & 0.4969 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 \\ -1 & -1 & 0 & 0 \\ -0.0001238 & -0.0001238 & 0 & 0.0026 \\ -1.238e{-6} & -1.238e{-6} & 0 & 2.6e{-5} \\ -0.02366 & -0.02366 & -1 & 0.4969 \end{bmatrix}$$

$$C := \begin{bmatrix} 0 & 0 & 1000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 18.87 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1e4 \\ 1.712e6 & 1411 & -47.62 & -0.8985 & 9.07e6 & 0 & 0 & 0 \end{bmatrix}$$

$$D := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -0.04762 & -0.04762 & 0 & 0 \end{bmatrix}$$

References

[1] G. Balas, J.C. Doyle, K. Glover, A. Packard, and R. Smith. *μ-Analysis and Synthesis Toolbox*. MathWorks, 1995.

[2] B.R. Barmish, P.P. Khargonekar, and Z. Shi. Robustness margin need not be a continuous function of the problem data. *Systems and Control Letters*, 15(2):91–98, 1990.

[3] D.G. Bates and T. Mannchen. Improved computation of mixed μ bounds for flight control law robustness analysis. *Proc. of the IMECHE, Part I, Journal of Sys. and Control Eng.*, 218(8):609–620, 2004.

[4] R. Braatz, P. Young, J. Doyle, and M. Morari. Computational complexity of μ calculation. *IEEE Trans. on Automatic Control*, 39(5):1000–1002, 1994.

[5] R.L. Dailey. A new algorithm for the real structured singular value. In *ACC*, pages 3036–3040, 1990.

[6] R.R.E. de Gaston and M.G. Safonov. Exact calculation of the multiloop stability margin. *IEEE Trans. on Automatic Control*, 33(2):156–171, 1988.

[7] J. Demmel. The componentwise distance to the nearest singular matrix. *SIAM J. Matrix Anal. Appl.*, 13(1):10–19, 1992.

[8] J. Doyle. Analysis of feedback systems with structured uncertainties. *IEE Proceedings, Part D*, 129(6):242–250, 1982.

[9] M. Elgersma, J. Freudenberg, and B. Morton. Polynomial methods for the structured singular value with real parameters. *International Journal of Robust and Nonlinear Control*, 6:147–170, 1996.

[10] M.K.H. Fan, A.L. Tits, and J.C. Doyle. Robustness in the presence of mixed parametric uncertainty and unmodeled dynamics. *IEEE Trans. on Automatic Control*, 36(1):25–38, 1991.

[11] G. Ferreres. *A practical approach to robustness analysis with aeronautical applications*. Kluwer, 1999.

[12] M. Fu. The real structured singular value is hardly approximable. *IEEE Trans. on Automatic Control*, 42(9):1286–1288, 1997.

[13] M.J. Hayes, D.G. Bates, and I. Postlethwaite. New tools for computing tight bounds on the real structured singular value. *AIAA Journal of Guidance, Control, and Dynamics*, 24(6):1204–1213, 2001.

[14] J. Kim, D.G. Bates, and I. Postlethwaite. An efficient algorithm for calculating lower bounds on μ with large numbers of possible repeated real uncertainties. Submitted to the 2006 CDC.

[15] M.P. Newlin and S.T. Glavaski. Advances in the computation of μ lower bound. In *ACC*, pages 442–446, 1995.

[16] M.P. Newlin and P.M. Young. Mixed μ problems and branch and bound techniques. In *CDC*, pages 3175–3180, 1992.

[17] M.P. Newlin and P.M. Young. Mixed μ problems and branch and bound techniques. *International Journal of Robust and Nonlinear Control*, 7:145–164, 1997.

[18] A. Packard, G. Balas, R. Liu, and J. Shin. Results on worst-case performance assessment. In *ACC*, pages 2425–2427, 2000.

[19] A. Packard and J. Doyle. The complex structured singular value. *Automatica*, 29(1):71–109, 1993.

[20] A. Packard, M.K.H. Fan, and J. Doyle. A power method for the structured singular value. In *CDC*, pages 2132–2137, 1988.

[21] A. Packard and P. Pandey. Continuity properties of the real/complex structured singular value. *IEEE Trans. on Automatic Control*, 38(3):415–428, 1993.

[22] A. Sideris and R.S. Sanchez Pena. Fast computation of the multivariable stability margin for the real interrelated uncertain parameters. *IEEE Trans. on Automatic Control*, 34(12):1272–1276, 1989.

[23] A. Sideris and R.S. Sanchez Pena. Robustness margin calculation with dynamic and real parametric uncertainty. *IEEE Trans. on Automatic Control*, 35(8):970–974, 1990.

[24] J.E. Tierno and P.M. Young. An improved μ lower bound via adaptive power iteration. In *CDC*, pages 3181–3186, 1992.

[25] P.M. Young. *Robustness with Parametric and Dynamic Uncertainty*. PhD thesis, Cal. Tech., 1993.

[26] P.M. Young and J.C. Doyle. Computation of μ with real and complex uncertainties. In *CDC*, pages 1230–1235, 1990.

[27] P.M. Young and J.C. Doyle. A lower bound for the mixed μ problem. *IEEE Trans. on Automatic Control*, 42(1):123–128, 1997.

[28] P.M. Young, M.P. Newlin, and J.C. Doyle. μ analysis with real parametric uncertainty. In *CDC*, pages 1251–1256, 1991.

[29] P.M. Young, M.P. Newlin, and J.C. Doyle. Practical computation of the mixed μ problem. In *ACC*, pages 2190–2194, 1992.