Proceeding of the 2004 American Control Conference
Boston, Massachusetts June 30 - July 2, 2004

A Bisection Algorithm for the Mixed μ Upper Bound and its Supremum

Carl-Magnus Fransson
Control and Automation Laboratory, Department of Signals and Systems
Chalmers University of Technology, SE-412 96 Göteborg, Sweden

Michael A. Saunders
Systems Optimization Laboratory
Department of Management Science and Engineering
Stanford University, Stanford CA 94305-4026, USA

***Abstract*— A new approach to computing the mixed μ upper bound (ν) is presented. The method exploits the fact that a positive de£nite matrix $V(\alpha)$ becomes singular when the scalar parameter α decreases to a critical value for a given frequency. A two-level optimization strategy is used with a bisection algorithm branching on the de£niteness of V in an outer loop, and a Semi-De£nite Programming (SDP) problem is formulated in an inner loop. Three different formulations are posed for the inner loop. The £rst uses a feasibility formulation (no objective function) with a constraint $V \succeq 0$, which tends to make V singular if possible. The second introduces an additional variable that makes the SDP feasible at all times. In the third formulation, the trace of V is minimized with the constraint $V \succeq 0$, which tends to minimize the rank of V and hence make V singular. The method is applied to a distillation column benchmark problem. Although it is computationally more expensive than existing methods when computing ν for a single frequency, it is a conceptually simple method that can be ef£cient when computing the supremum of ν with respect to frequency.**

Keywords: bisection, LMI, μ, ν, performance, rank minimization, robustness, SDP, uncertainty

I. INTRODUCTION

All controller designs are implicitly, or explicitly, a trade-off between performance and robustness in different frequency regions and to different kinds of disturbances. In methods based on a minimization of a single criterion the trade-offs are the result of the choice of weight functions, which in many cases is not a trivial task. Also, performance and robustness speci£cations on the closed loop are typically not included in the synthesis (optimization) but have to be con£rmed afterwards. If the speci£cations are not met, a criterion with new weight functions must be formulated and tried. If the speci£cations *are* met, a controller achieving better closed loop performance, still meeting the speci£cations, could often be found if a different criterion were used. Hence, it is desirable that the speci£cations be included in the synthesis so that the effect of a change of speci£cations can easily be related to a change in performance. One way of formulating this problem is

$$K^* = \arg\min_K J(K), \tag{1}$$

$$\|\Psi_j(K,\Delta)\|_\infty \le c_j, \quad j = 1,\dots,N, \ \forall\Delta \in \boldsymbol{\Delta},$$

where J is a performance measure to be minimized, and Ψ_j are closed loop transfer functions (possibly weighted) that has to meet speci£cations c_j for all Δ de£ning a deviation from the nominal plant model. $\boldsymbol{\Delta}$ is the set of all such possible plant uncertainties, and K belongs to a class of controllers.

The non-convex problem formulation (1) was used in [7] to design controllers robust to parametric uncertainties in the plant model. Each of the constraints in (1) was expressed as a constraint on $\sup_{\omega\in\mathbb{R}} \nu$, the supremum with respect to frequency of the upper bound of the *structured singular value* (denoted μ). The most common approach to computing such constraints is by frequency gridding and subsequent use of the upper bound algorithm by Fan *et al.* [4] (implemented in, e.g., the routine `mu` in the μ-Analysis and Synthesis Toolbox [1] for Matlab [9]), for each frequency. For each iteration in the outer optimization loop of (1), calculating a constraint value thus involves solving a series of inner loop optimization problems. Achieving accurate constraint values is necessary for convergence in the outer loop but also very costly in terms of computation time because we have to calculate ν for multiple frequencies. In fact, calculating ν for a single frequency is NP hard [17] and although solutions can often be found in practice, it is nevertheless a costly computation.

This paper presents an alternative, simple, method for calculating $\sup_{\omega\in\mathbb{R}} \nu$. A two level optimization strategy is proposed where a bisection algorithm is applied in the upper level and a Semi-De£nite Programming (SDP) problem is formulated in the lower level. It is shown that computation time can sometimes be reduced relative to the established routine `mu` in [1].

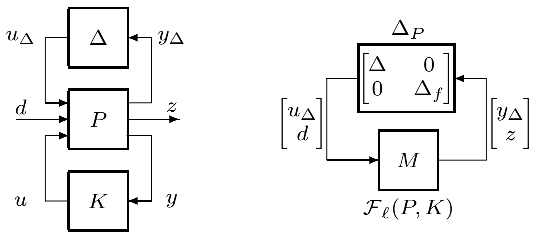

Fig. 1. Closed loop system in an LFT setup (left) and in a robust performance setup (right).

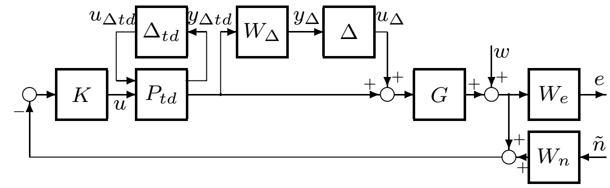

Fig. 2. Feedback system with a nominal plant G, a controller K, an LFT matrix P_{td}, uncertainty blocks Δ and Δ_{td}, disturbance signals w and $\tilde{n}$, and weights W.

II. Preliminaries

Consider a controller $K(s)$ regulating an uncertain *generalized plant* $P_\Delta(s)$, de£ned as the transfer function from some disturbances d and control signals u to performance outputs z and plant outputs y. If the uncertainties can be separated from the plant in a *linear fractional transformation* (LFT) [2] fashion, the system can be described as in the left part of Figure 1. Here, P is the nominal (generalized) plant and Δ is a structured block diagonal matrix representing all admissible uncertainties (de£ned later by the set $\boldsymbol{\Delta}$). Using lower LFT, a nominal closed loop system M can be de£ned as $M = \mathcal{F}_\ell(P, K)$ such that

$$\begin{bmatrix} y_\Delta \\ z \end{bmatrix} = M \begin{bmatrix} u_\Delta \\ d \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} u_\Delta \\ d \end{bmatrix}. \qquad (2)$$

Now, connect z with d by a block matrix Δ_f (sometimes referred to as a £ctitious uncertainty block) having the property $\|\Delta_f\|_\infty \leq 1$, and de£ne $\Delta_P = \mathrm{diag}\{\Delta, \Delta_f\}$. The system can then be drawn as in the right part of Figure 1. To guarantee that the closed loop system $\mathcal{F}_\ell(P_\Delta, K)$ is stable (robust stability) as well as that $\|\mathcal{F}_\ell(P_\Delta, K)\|_\infty < 1$ for all admissible Δ (robust performance), the theory for μ can be used [3], [13]. μ, like the singular values σ, is a frequency-dependent scalar measure of matrix size, the main difference from σ being that it takes structured uncertainties in the matrix elements into account.

This study concerns the mixed uncertainty case where the system may have both real parametric and complex uncertainties. The mixed μ involves three types of blocks: repeated scalar blocks, complex repeated scalar blocks and full blocks with real or complex entries. Introduce integers $k_1, \ldots, k_{S_r},\ r_1, \ldots, r_{S_c}$, and $m_1, \ldots, m_F$ such that the ith repeated real scalar block is $k_i \times k_i$, the jth repeated complex scalar block is $r_j \times r_j$ and the ℓth full block is $m_\ell \times m_\ell$. The set of admissible uncertainties is then de£ned as

$$\begin{aligned} \boldsymbol{\Delta} = \{&\mathrm{diag}[\,\delta_i I_{k_i},\ \phi_j I_{r_j},\ \Delta_\ell\,] : \\ &\delta_i \in \mathbb{R},\ \phi_j \in \mathbb{C},\ \Delta_\ell \in \mathbb{C}^{m_\ell \times m_\ell}\}, \end{aligned} \qquad (3)$$

where $i = 1, \ldots, S_r,\ j = 1, \ldots, S_c,\ \ell = 1, \ldots, F$, $\|\phi_i\|_\infty < 1$, $\|\delta_j\|_\infty < 1$, $\|\Delta_\ell\|_\infty < 1$. For a given frequency ω, let $\overline{\sigma}(\Delta)$ denote the largest singular value of Δ, and let $M \in \mathbb{C}^{n \times n}$. If $\exists\ \Delta \in \boldsymbol{\Delta} : \det(I - M\Delta) = 0$, the structured singular value of M with respect to $\boldsymbol{\Delta}$ is de£ned as

$$\mu_{\boldsymbol{\Delta}}(M) = \frac{1}{\min\{\overline{\sigma}(\Delta) : \Delta \in \boldsymbol{\Delta}, \det(I - M\Delta) = 0\}}. \qquad (4)$$

If no $\Delta \in \boldsymbol{\Delta}$ makes $I - M\Delta$ singular, then $\mu_{\boldsymbol{\Delta}}(M) = 0$. It is not suitable to compute $\mu(M)$ with (4) because the resulting optimization problem may have multiple optima. However, upper and lower bounds for μ can be computed as in [4], [19] (implemented in [1]), for example. Often the lower and upper bounds coincide, but when this is not the case, conservative solutions with respect to the uncertainties can be expected. Since the function $\mu : \mathbb{C}^{n \times n} \rightarrow \mathbb{R}$ does not satisfy the triangle inequality, it is not a norm. Despite this we de£ne

$$\|M\|_\mu^{\boldsymbol{\Delta}} = \sup_{\omega \in \mathbb{R}} \mu_{\boldsymbol{\Delta}}(M(j\omega)).$$

For future reference, also de£ne the sets $\mathcal{D}$ and $\mathcal{G}$:

$$\begin{aligned} \mathcal{D} = \{&\mathrm{diag}[\,D_i,\ \tilde{D}_j,\ d_\ell I_{m_\ell}\,] : D_i = D_i^* \succ 0, \\ &\tilde{D}_j = \tilde{D}_j^* \succ 0,\ d_\ell > 0,\ d_F = 1\}, \\ \mathcal{G} = \{&\mathrm{diag}[\,G_i,\ 0,\ 0\,] : G_i = G_i^*\}, \end{aligned}$$

where $D_i \in \mathbb{C}^{k_i \times k_i},\ \tilde{D}_j \in \mathbb{C}^{r_j \times r_j},\ d_\ell \in \mathbb{R}$, and $G_i \in \mathbb{C}^{k_i \times k_i}$.

III. Robust performance

Let $G(s)$ denote an open loop plant model, $K(s)$ a controller, and $S(s) = (I - G(s)K(s))^{-1}$ and $KS(s) = K(s)(I - G(s)K(s))^{-1}$ the *sensitivity function* and *control sensitivity function* respectively. In the single input-single output (SISO) case, $1/\|S\|_\infty$ is the shortest distance from the open loop (GK) to the instability point in the Nyquist diagram and hence $\|S\|_\infty$ is a natural robustness measure. In the multiple input-multiple output (MIMO) case, $\|S\|_\infty$ can be seen as a robustness measure related to the damping of the system. KS is the transfer function from measurement noise (or the reference signals) to the control signals, implying that $\|KS\|_\infty$ is a measure of the activity in the control signals caused by measurement noise or changes in the reference signals—the *control activity*. To achieve a controller that results in good closed loop performance and robustness it is often desirable to keep these two measures restricted to $\|S\|_\infty \leq c_S$ and $\|KS\|_\infty \leq c_{KS}$ say. These design constraints can be readily extended to hold for all admissible uncertainties Δ by utilizing the theory for μ as follows.

Consider a nominal plant G with input multiplicative uncertainty ΔW_Δ and input time delay uncertainty represented in LFT form by Δ_{td} and P_{td}. Assume that a controller K has been synthesized, and that we want to check if $\|S\|_\infty \le c_S$, and $\|KS\|_\infty \le c_{KS}$ $\forall \Delta \in \mathbf{\Delta}$. Introduce the weights W_e, W_n as in Figure 2, and choose $W_e = c_S^{-1} I$, $W_n = c_{KS}^{-1} I$. Formulate two μ problems: In the £rst choose $d = w$ and $z = e$, and in the second choose $d = \tilde{n}$ and $z = u$. With $T_i = (I + KG)^{-1} KG$, $S_i = I - T_i$, and P_{td} partitioned analogously to M in (2), the nominal closed loop systems for the two problems become

$$\begin{bmatrix} y_\Delta \\ y_{\Delta td} \\ e \end{bmatrix} = M_S \begin{bmatrix} u_\Delta \\ u_{\Delta td} \\ w \end{bmatrix}, \quad \begin{bmatrix} y_\Delta \\ y_{\Delta td} \\ u \end{bmatrix} = M_{KS} \begin{bmatrix} u_\Delta \\ u_{\Delta td} \\ \tilde{n} \end{bmatrix},$$

$$M_S = \begin{bmatrix} -W_\Delta T_i & W_\Delta S_i P_{td,21} & -W_\Delta KS \\ -T_i & (\frac{1}{2} I - T_i) P_{td,21} & -KS \\ c_S^{-1} S_i G & c_S^{-1} S_i G P_{td,21} & c_S^{-1} S_i \end{bmatrix},$$

$$M_{KS} = \begin{bmatrix} -W_\Delta T_i & W_\Delta S_i P_{td,21} & -c_{KS}^{-1} W_\Delta KS \\ -T_i & (\frac{1}{2} I - T_i) P_{td,21} & -c_{KS}^{-1} KS \\ -T_i & -T_i P_{td,21} & -c_{KS}^{-1} KS \end{bmatrix}.$$

Let Δ_P be de£ned as in Section II and Figure 1 such that a full complex matrix Δ_f connects e with w and u with $\tilde{n}$ respectively. Then by the performance robustness theorem [3], the resulting closed loop system is stable and the speci£cations on S and KS are satis£ed for all $\Delta \in \mathbf{\Delta}$ if and only if $\|M_S\|_\mu^{\mathbf{\Delta}_P} \le 1$, and $\|M_{KS}\|_\mu^{\mathbf{\Delta}_P} \le 1$.

IV. The μ upper bound

It is well known that $\mu(M) \le \inf_{D \in \mathcal{D}} \bar{\sigma}(DMD^{-1})$ [3], which forms the basis for computation techniques for the complex μ problem (i.e. no real uncertainties). This bound is often conservative, especially when $S_r > 0$, and an improved upper bound for the mixed μ problem was presented in [4]:

$$\mu(M) \le \inf_{D \in \mathcal{D}} \nu(DMD^{-1}) \le \inf_{D \in \mathcal{D}} \bar{\sigma}(DMD^{-1}), \quad (5)$$

where

$$\nu(M) \equiv \sqrt{\max\{0, \inf_{G \in \mathcal{G}} \bar{\lambda}[M^* M + j(GM - M^* G)]\}}.$$

It was also shown that the improved upper bound can be obtained as

$$\inf_{D \in \mathcal{D}} \nu(DMD^{-1}) = \max\{0, \sqrt{\alpha^{\text{opt}}}\}, \text{ where}$$

$$\alpha^{\text{opt}} = \inf_{D \in \mathcal{D}, G \in \mathcal{G}} \max_{\alpha \in \mathbb{R}} \{\alpha : \bar{\lambda}[-V] \ge 0\}, \quad (6)$$

$$V(\alpha, D, G) \equiv \alpha D - M^* DM - j(GM - M^* G).$$

A. A bisection algorithm for ν

The following lemma uses the property that V is singular at the solution to (6) [4]:

Lemma 1: Let D^{opt} and G^{opt} denote the scaling matrices at the solution to (6). Then,
$\bar{\lambda}[-V(\alpha, D^{\text{opt}}, G^{\text{opt}})] = 0$ at the solution to the optimization problem

$$\max_{\alpha > 0} \{\alpha : \bar{\lambda}[-V(\alpha, D^{\text{opt}}, G^{\text{opt}})] \ge 0\}.$$

Therefore,

$$\max_{\alpha > 0} \{\alpha : \underline{\lambda}[V(\alpha, D^{\text{opt}}, G^{\text{opt}})] = 0\}.$$

is an equivalent formulation. □
In view of lemma 1 and (6), we thus have that

$$\alpha^{\text{opt}} = \inf_{D \in \mathcal{D}, G \in \mathcal{G}} \max_{\alpha > 0} \{\alpha : V(\alpha, D, G) \ \text{singular}\}. \quad (7)$$

Singularity of $V(\alpha, D, G)$ can be asserted in several ways and here, three strategies will be discussed. Since V is linear in D and G, we propose to solve (7) by determining D and G in an inner loop and applying a bisection algorithm for α, based on the de£niteness of V.

Bisection Algorithm for α

1) Set $k = 1$. Initialize $\alpha_k = \alpha_0$ and set upper and lower bounds for α: $0 < \underline{\alpha} < \bar{\alpha}$.
2) Determine D_k and G_k by solving the SDP problem **P1**, **P2**, or **P3** in Proposition 1 and compute $V(\alpha_k, D_k, G_k)$.
3) If $\underline{\lambda}[V] < 0$, $\underline{\alpha} = \alpha_k$; otherwise V is positive semi-de£nite and we can set $\bar{\alpha} = \alpha_k$.
4) If $\bar{\alpha} - \underline{\alpha} < \text{tol}$, return $\alpha^{\text{opt}} = \alpha_k$, $D^{\text{opt}} = D_k$, $G^{\text{opt}} = G_k$; otherwise set $k = k + 1$, update $\alpha_k = (\bar{\alpha} + \underline{\alpha})/2$ and go to 2.

Proposition 1:

$$\inf_{D \in \mathcal{D}, G \in \mathcal{G}} \max_{\alpha_k > 0} \{\alpha_k : V(\alpha_k, D, G) \text{ singular}\} \quad \Leftrightarrow$$
$$\min_{\alpha > 0} \{\alpha_k : V(\alpha_k, D_k, G_k) \text{ singular}\},$$

where D_k and G_k are determined for each α_k by any of the three following SDP problems:

$$\mathbf{P1}: \quad D_k, G_k = \arg \min_{D \in \mathcal{D}, G \in \mathcal{G}} \{0 : V(\alpha_k, D, G) \succeq 0, \ D - \varepsilon I \succeq 0\}$$

$$\mathbf{P2}: \quad D_k, G_k = \arg \min_{D \in \mathcal{D}, G \in \mathcal{G}} \{\gamma : V(\alpha_k, D, G) + \gamma I \succeq 0, \ D - \varepsilon I \succeq 0, \ \gamma > 0\}$$

$$\mathbf{P3}: \quad D_k, G_k = \arg \min_{D \in \mathcal{D}, G \in \mathcal{G}} \{\mathbf{Tr}[V(\alpha_k, D, G)] : V(\alpha_k, D, G) \succeq 0, \ D - \varepsilon I \succeq 0\}$$

$\mathbf{Tr}$ denotes the matrix trace operation $\mathbf{Tr}(V) = \sum_i^n V_{i,i}$ and ε denotes a small number > 0. □
P1, **P2**, and **P3**, which can be solved by any suitable SDP solver (such as the LMI Control Toolbox [8], PENSDP

[10] or SeDuMi [15]) are now discussed in more detail to motivate Proposition 1.

P1: For any $\alpha_k > 0$, we want to £nd out if there exist $D \in \mathcal{D}$ and $G \in \mathcal{G}$ such that $V(\alpha_k, D, G) \succeq 0$; i.e., solve a feasibility problem. It is well known that SDP problems have a tendency to result in semi-de£nite (rather than strictly de£nite) solutions whenever possible. In our case, this corresponds to making $V(\alpha_k, D, G)$ singular, hence motivating **P1**. Also note that for £xed $D \in \mathcal{D}$ and $G \in \mathcal{G}$, $\bar{\lambda}[-V]$ is a strictly decreasing function of α [4] and therefore we also have that $\underline{\lambda}[V]$ is a strictly increasing function of α.

P2: Note that **P1** becomes infeasible when $\alpha_k > \alpha^{\text{opt}}$. Historically, infeasible problems have caused dif£culties for numerical optimization algorithms, including interior point algorithms, although recent software packages such as SeDuMi are able to handle these dif£culties nicely [16] by use of a self-dual embedding technique [18], for example. However, the interpretation of a feasible solution is more clear and it gives some insight to the problem discussed in this study. By introducing a new variable $\gamma \geq 0$, we ensure that the problem is feasible at all times. We now have a choice of branching on the de£niteness of V or on the value of γ. In the latter case, however, the bisection algorithm will have to be modi£ed slightly. It will be shown that γ is a strictly decreasing function of α and that the unique solution α^{opt} is located at the intersection of γ and $\underline{\lambda}[V]$. Thus, α^{opt} is the only value of α for which both $\underline{\lambda}[V] = 0$ and $\gamma = 0$.

P3: The motivation for **P3** relies on minimum-rank theory. The general rank-minimization problem for real matrices can be expressed as

$$\min_{V \in \mathbb{R}^{m \times n}} \{\textbf{Rank}(V) : V \in \mathcal{C}\}, \tag{8}$$

where $\mathcal{C}$ is a convex set, e.g., described by Linear Matrix Inequalities (LMI's). While (8) is hard to solve in general, the following convex optimization problem tends to minimize the rank of V and hence approximate the solution of (8) [5], [12]:

$$\min_{V \in \mathbb{R}^{m \times n}} \{\|V\|_* : V \in \mathcal{C}\}. \tag{9}$$

The *nuclear norm* $\|\cdot\|_*$ is de£ned by

$$\|V\|_* = \sum_{i=1}^{\min(m,n)} \sigma_i(V),$$

where $\sigma_i(V) = \sqrt{\lambda_i(V^T V)}$ denote the singular values of V.

When V is positive semi-de£nite and Hermitian, and the constraint set $\mathcal{C}$ is de£ned by LMI's, (9) can be expressed as an SDP problem [6]:

$$\min_{V \in \mathbb{R}^{m \times n}} \{\textbf{Tr}(V) : V \succeq 0, V \in \mathcal{C}\}. \tag{10}$$

In our case, $V \in \mathbb{C}^{m \times n}$ but by applying the following transformation to V we can obtain real valued matrices needed for the rank minimization technique:

$$V \rightarrow \begin{bmatrix} \text{Re}(V) & \text{Im}(V) \\ -\text{Im}(V) & \text{Re}(V) \end{bmatrix}.$$

V. Numerical example

Our experience is that the semi-de£nite programming/large-scale optimization package SeDuMi, which runs on top of MATLAB [9], is currently the most robust solver available for solving SDP problems. It exploits sparsity and can handle infeasible problems. A MATLAB interface yalmip [11] provides easy translation from the matrix forms in Proposition 1 to the input format of many SDP solvers, including SeDuMi.

A. Plant model

The method is applied to a distillation column benchmark problem (30^{th} IEEE Conference on Decision and Control, Brighton, England, 1991) with independent real gain and time delay uncertainty at the input channels. The uncertain plant is de£ned by

$$G(s) = \frac{1}{1+75s} \begin{bmatrix} 87.8 & -86.4 \\ 108.2 & -109.6 \end{bmatrix} \begin{bmatrix} g_1 e^{-\theta_1 s} & 0 \\ 0 & g_2 e^{-\theta_2 s} \end{bmatrix},$$

$$\text{where } g_1, g_2 \in [0.8, 1.2], \quad \theta_1, \theta_2 \in [0, 1].$$

The time delay was modeled by a second order Padé approximation which proved to be suf£cient in this example, and the nominal plant is de£ned by $g_1 = g_2 = 1$, and $\theta_1 = \theta_2 = 0.5$. By introducing norm-bounded uncertainty blocks $\Delta, \Delta_{td} \in \mathbf{\Delta}$ with $S_r = 4$, $k_1 = k_2 = k_3 = k_4 = 1$, $S_c = 0$, $F = 1$, $m_1 = 2$ (see (3)), we can de£ne the plant as

$$G(s) = G_0(s)(I_2 + W_\Delta \Delta)\mathcal{F}_u(P_{td}, \Delta_{td}),$$

$$\text{where} \quad W_\Delta = 0.2 I_2, \quad \Delta = \begin{bmatrix} \delta_1 & 0 \\ 0 & \delta_2 \end{bmatrix},$$

$$\mathcal{F}_u(P_{td}, \Delta_{td}) = \begin{bmatrix} \frac{4-\delta_3 s}{4+\delta_3 s} & 0 \\ 0 & \frac{4-\delta_4 s}{4+\delta_4 s} \end{bmatrix}, \quad \delta_i \in [-1, 1].$$

B. Computing $\mu_{\mathbf{\Delta}_P}(M)$

We calculate the upper bound of $\mu_{\mathbf{\Delta}_P}(M_S)$ and $\mu_{\mathbf{\Delta}_P}(M_{KS})$ with the bisection algorithm in Section IV-A for the example plant in Section V-A and a controller achieving $\|S\|_\infty \leq 1.7$, and $\|KS\|_\infty \leq 2 \ \forall \Delta \in \mathbf{\Delta}$, and compare with results from the μ-Analysis and Synthesis Toolbox. For this purpose, we chose 100 logarithmically spaced frequencies in the interval $\omega = [0.01, 100]$. The bisection algorithm was initialized with $\underline{\alpha} = 0$, $\bar{\alpha} = 10$, and $\alpha_0 = 5$ for the £rst frequency. Thereafter, α_0 was chosen as the optimal value for the previous frequency. The convergence tolerances for the SDP problem **P1**, **P2**, or **P3** as well as in the bisection search were chosen to be 10^{-4}, and for this value the bisection algorithm converges in 15 iterations. The parameter ε was set to 0.1 to ensure positive de£niteness of D. When computing the upper bound with the μ-Analysis and Synthesis Toolbox, maximum accuracy

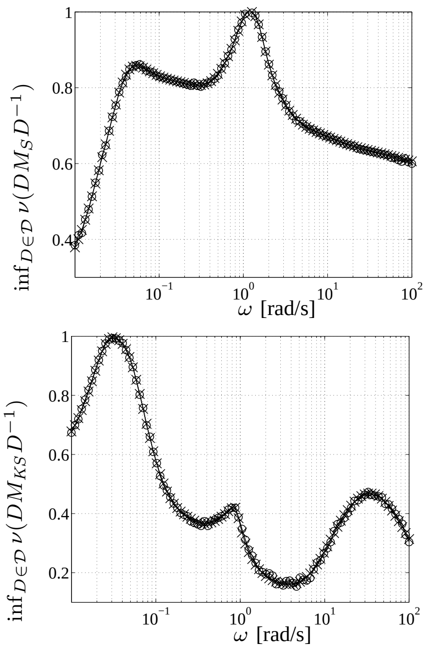

Fig. 3. The μ upper bound vs. ω for M_S (top) and M_{KS} (bottom). Circles correspond to solutions with the bisection algorithm in Section IV-A using the SDP formulation **P1**, and crosses correspond to solutions with the μ-Analysis and Synthesis Toolbox.

Method	Matrix	CPU [s]
Bisection + **P1**	M_S	141
Bisection + **P1**	M_{KS}	95
Bisection + **P2**	M_S	151
Bisection + **P2**	M_{KS}	107
Bisection + **P3**	M_S	177
Bisection + **P3**	M_{KS}	130
μ Toolbox	M_S	32
μ Toolbox	M_{KS}	33

TABLE I

COMPUTATIONAL RESULTS FOR 100 VALUES OF THE μ UPPER BOUND OF M_S AND M_{KS} USING DIFFERENT METHODS. ALL CALCULATIONS WERE DONE WITH A 1GHZ PENTIUM III PROCESSOR.

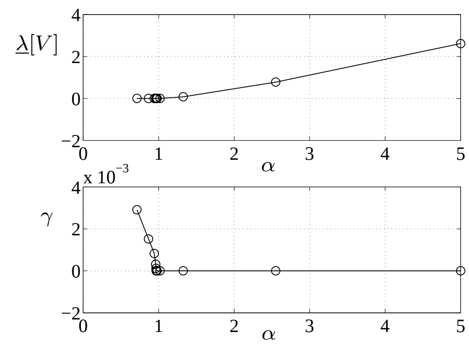

Fig. 4. $\underline{\lambda}[V]$ (top) and γ (bottom) vs. α during a bisection search with **P2** for $\omega = 1$ rad/s and $\alpha_0 = 5$.

(level C9) had to be requested in order to match the results of the bisection algorithm.

Figure 3 shows the results for the bisection algorithm using **P1** (circles) and the μ-Analysis and Synthesis Toolbox (crosses). (The results using **P2** or **P3** are very similar to those for **P1** and are omitted for graphical clarity.) Clearly, the two approaches result in the same upper bound for both $\mu_{\mathbf{\Delta}_P}(M_S)$ and $\mu_{\mathbf{\Delta}_P}(M_{KS})$. With the chosen tolerances, the bisection algorithm is signi£cantly more expensive computationally (for this example) as shown in Table I. Note that since $\|M_S\|_\mu^{\mathbf{\Delta}_P} = 1$ and $\|M_S\|_\mu^{\mathbf{\Delta}_P} = 1$, the speci£cations are achieved precisely. Also note that in practice, one could obtain similar curves with less demanding tolerances.

Figure 4 shows how $\underline{\lambda}[V]$ and γ vary with α for the bisection algorithm with the SDP formulation **P2** and $\omega = 1$ rad/s. The £gure con£rms the statement made earlier that α^{opt} is the only value of α for which both $\underline{\lambda}[V] = 0$ and $\gamma = 0$. Further, the £gure shows that $\underline{\lambda}[V]$ and γ are non-decreasing and non-increasing functions of α respectively.

C. Computing $\|M\|_\mu^{\mathbf{\Delta}_P}$

During controller synthesis it is often suf£cient to have information about the value of $\|M\|_\mu^{\mathbf{\Delta}_P} = \sup_{\omega\in\mathbb{R}} \mu_{\mathbf{\Delta}_P}(M)$ rather than the upper bound at each frequency. Note that the general mixed μ problem is not necessarily a continuous function of ω, and may only be upper semi-continuous [14]. This means that one has to compute μ for in£nitely many frequencies in order to ensure that $\|M\|_\mu^{\mathbf{\Delta}}$ is not underestimated. However, it was shown in [14] that for the robust performance problem, continuity of mixed μ is ensured if $\mu_{\mathbf{\Delta}}(M_{11}) < \mu_{\mathbf{\Delta}_P}(M)$, which can easily be checked for.

To compute $\|M\|_\mu^{\mathbf{\Delta}_P}$, we apply the bisection algorithm for successive frequencies but add another stopping criterion to make the algorithm computationally more ef£cient. Let $\Omega = \{\omega_q\}_{q=1}^N$ denote a set of N discrete frequencies for which we want to compute $\mu_{\mathbf{\Delta}_P}(M)$, and denote by ω_p the current frequency being applied in the bisection algorithm ($1 \leq p \leq N$). Further, de£ne the set $\mathcal{A}_p$ to include the optimal solutions for all $q \leq p$ such that $\mathcal{A}_p = \{\alpha^{\text{opt}}(\omega_q)\}_{q=1}^p$. We then have that the search for $\alpha^{\text{opt}}(\omega_{p+1})$ can be terminated if

$$\alpha_k < \max_{\alpha\in\mathcal{A}_p} \alpha. \tag{11}$$

$\|M_S\|_\mu^{\mathbf{\Delta}_P}$ and $\|M_{KS}\|_\mu^{\mathbf{\Delta}_P}$ were computed by this strategy for the example plant in Section V-A, and Figure 5 compares the results of the bisection algorithm using **P1** (circles) and the μ-Analysis and Synthesis Toolbox (crosses). From Table II, which summarizes the results for all methods, we see that when computing $\|M\|_\mu^{\mathbf{\Delta}_P}$, one can save signi£cant computation time by adding the stopping criterion (11) to the bisection algorithm. Importantly, this example also shows that in some cases, it is computationally cheaper to compute $\|M\|_\mu^{\mathbf{\Delta}_P}$ with a simple bisection algorithm in the outer loop and an SDP solver in the inner loop, than with the μ-Analysis and Synthesis Toolbox. When $\|M\|_\mu^{\mathbf{\Delta}_P}$ appears as constraints in the general controller synthesis problem

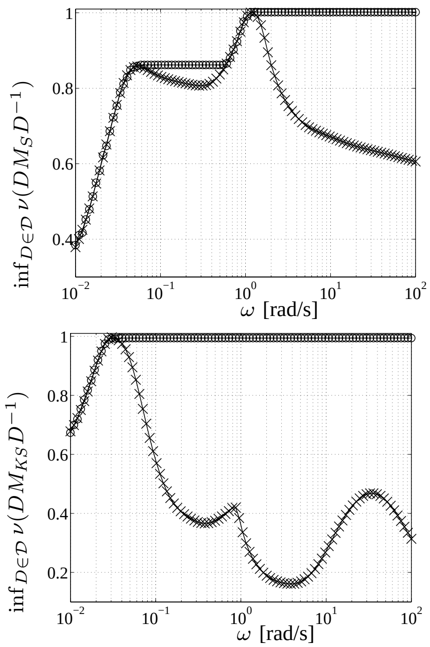

Fig. 5. The μ upper bound vs. ω for M_S (top) and M_{KS} (bottom). Circles correspond to solutions with the bisection algorithm in Section IV-A using the SDP formulation **P1**, and crosses correspond to solutions with the μ-Analysis and Synthesis Toolbox.

Method	Matrix	CPU [s]	$\|M\|_{\mu}^{\mathbf{\Delta}P}$	$\omega^*[\frac{\text{rad}}{\text{s}}]$
Bisection + **P1**	M_S	52	1.00	1.15
Bisection + **P1**	M_{KS}	25	1.00	0.031
Bisection + **P2**	M_S	56	1.00	1.15
Bisection + **P2**	M_{KS}	28	1.00	0.031
Bisection + **P3**	M_S	80	1.00	1.15
Bisection + **P3**	M_{KS}	39	1.00	0.031
μ Toolbox	M_S	32	1.00	1.15
μ Toolbox	M_{KS}	33	1.00	0.031

TABLE II

COMPUTATIONAL RESULTS FOR $\|M_S\|_{\mu}^{\mathbf{\Delta}P}$ AND $\|M_{KS}\|_{\mu}^{\mathbf{\Delta}P}$ USING DIFFERENT METHODS. ALL CALCULATIONS WERE DONE WITH A 1GHZ PENTIUM III PROCESSOR.

(1), the importance of these computational savings may be of importance.

VI. CONCLUSIONS

A new approach to compute the mixed μ upper bound (ν) has been presented. The method exploits the fact that a matrix V is singular at the optimal ν value for a given frequency, and is based on a two-level optimization strategy where a bisection algorithm, branching on the de£niteness of V, is used in an outer loop, and an SDP problem is formulated in an inner loop. The inner loop computes scaling matrices D and G for a £xed value of ν and the outer loop searches for the optimal value of ν for £xed matrices D and G.

It has been shown that the SDP problem in the inner loop may take various forms and in particular, three different problems (**P1**, **P2**, **P3**) were posed, see Section IV-A and Proposition 1. The £rst formulation was shown to be most ef£cient although the second has the nice property that it is feasible at all times. The third formulation, which relies on rank-minimization theory, was the slowest.

The method was applied to a distillation column benchmark problem. Although computationally more expensive than an existing method (the μ-Analysis and Synthesis Toolbox) when computing ν for a single frequency, the proposed method is conceptually simpler than the standard method, and can be more ef£cient when computing the supremum of ν with respect to frequency. The method could be combined with a frequency search for the supremum of ν (based on global optimization for example) to make it even more ef£cient. Applying such methods to the general controller synthesis problem, for example, could lead to signi£cant computational savings.

REFERENCES

[1] G. J. Balas, J. C. Doyle, K. Glover, A. K. Packard, and R. Smith. *μ-Analysis and Synthesis Toolbox, User's Guide*. Natick, MA, 1995.

[2] J. C. Doyle, A. K. Packard, and K. Zhou. Review of LFTs, LMIs and μ. In *Proc. 30th Conf. Decision and Control*, pages 1227–1232, Brighton, UK, 1991.

[3] J. C. Doyle, J. E. Wall, and G. Stein. Performance and robustness analysis for structured uncertainty. In *Proc. 21st Conf. Decision and Control*, Orlando, FL, 1982.

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

[5] M. Fazel. Matrix Rank Minimization with Applications. Ph.D. thesis, Stanford University, Stanford, CA, 2002.

[6] M. Fazel, H. Hindi, and S. P. Boyd. A rank minimization heuristic with application to minimum order system approximation. In *Proc. Amer. Control Conference*, pages 4734–4739, Arlington, VA, June 2001.

[7] C. M. Fransson, B. Lennartson, T. Wik, and K. Holmström. Multi criteria controller design for uncertain MIMO systems using global non-convex optimization. In *Proc. 40th Conf. Decision and Control*, pages 3976–3981, Orlando, FL, December 2001.

[8] P. Gahinet, A. Nemirovski, A. J. Laub, and M. Chilali. *LMI Control Toolbox, User's Guide*. The Math Works Inc., Natick, MA, 1995.

[9] The Mathworks Inc.¢ *MATLAB, Version 6, User's Guide*. The Mathworks Inc. Natick, MA, 2000.

[10] M. Koĉvara and M. Stingl. *PENSDP User's Guide*. Univ. of Erlangen, Germany, 2002.

[11] J. Löfberg. *YALMIP: Yet Another LMI Parser*. Linköping Institute of Tech., Linköping, Sweden, 2001. URL: www.control.isy.liu.se/ johanl/yalmip.html.

[12] M. Mesbahi. On the semi-de£nite programming solution of the least order dynamic output feedback synthesis. In *Proc. Amer. Control Conference*, pages 2355–2359, San Diego, CA, June 1999.

[13] A. K. Packard and J. C. Doyle. The complex structured singular value. *Automatica*, 29:71–109, 1993.

[14] A. K. Packard and P. Pandey. Continuity properties of the real/complex structured singular value. *IEEE Trans. Autom. Contr.*, 38:415–428, 1993.

[15] J. F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11-12:625:653, 1999. fewcal.kub.nl/sturm/software/sedumi.html.

[16] J. F. Sturm. Implementation of interior point methods for mixed semide£nite and second order cone optimization problems. *Optimization Methods and Software*, 17(6):1105–1154, 2002. Special 10th anniversary issue.

[17] O. Toker and H. Özbay. On the NP-hardness of solving bilinear matrix inequalities and simultaneous stabilization with static output feedback. In *Proc. Amer. Control Conference*, Seattle, WA, 1995.

[18] Y. Ye, M. J. Todd, and S. Mizuno. An $O(\sqrt{n}L)$-iteration homogeneous and self-dual linear programming algorithm. *Mathematics of Operations Research*, 19:53–67, 1994.

[19] P. M. Young. Robustness with parametric and dynamic uncertainty. Ph.D. thesis, California Institute of Technology, Pasadena, CA, 1993.