Pergamon

PII: S0098-1354(98)00004-0

Computers Chem. Engng Vol. 22, No. 7–8, pp. 913–926, 1998

0098–1354/98 $19.00 + 0.00

Model reduction for the robustness margin computation of large scale uncertain systems

Evan L. Russell and Richard D. Braatz*

Large Scale Systems Research Laboratory, Department of Chemical Engineering,
University of Illinois at Urbana-Champaign, 600 South Mathews Avenue, Box C-3, Urbana, IL 61801 U.S.A.

(Received August 1996; revised 18 December 1997)

Abstract

Efficient algorithms are developed for the model reduction of large scale uncertain systems. The polynomial-time algorithms rely primarily on well-conditioned singular value decomposition computations, and allow robustness stability and performance margins to be computed for uncertain systems of high dimensionality. Application to a realistic description of a large scale paper machine control system demonstrates the utility of the algorithms. © 1998 Elsevier Science Ltd. All rights reserved

Keywords: model reduction; algorithms; decomposition computations; robustness

1. Introduction

It is impossible to generate highly accurate phenomenological models for industrial-scale processes because of inaccurate values for the physical parameters of the process, and lack of complete understanding of the underlying physical phenomena (for example, during polymerization, crystallization, and drying). This necessitates the identification of a process model from input–output data collected during process operations. The quantity of this data is usually limited however, since data collection for an industrial-scale process is expensive and time consuming. The quality of this data is deteriorated by noise, unknown disturbances, and imperfectly operating equipment (e.g. stiction and clogging in valves). The limited quality and quantity of input–output data ensures that the identified model will not be an exact representation of the true process. It was recognized in the late 1970s that ignoring model inaccuracies could lead to poor performance or instability (Brosilow, 1979; Morari and Zafriou, 1989; Palmor and Shinnar, 1981; Skogestad and Postlethwaite, 1996).

Robustness analysis consists of testing whether a given controller provides the desired performance for all potential model parameter variations. The function, μ, is a nonconservative measure for analyzing system robustness (Doyle, 1982; Safonov, 1982), and is a direct generalization of the concept of the *gain margin* taught to engineers in their undergraduate process controls course (Ogunnaike and Ray, 1994; Seborg *et al.*, 1989; Stephanopoulos, 1983).

Large scale systems have large numbers of inputs and outputs, and include whole chemical plants as well as some unit operations, such as paper machines, polymer film extruders, and adhesive coaters (Braatz, 1997). The importance of ensuring robustness of the closed loop system to model uncertainties increases as the process dimensionality increases (Featherstone and Braatz, 1997; Braatz, 1997); hence developing algorithms for computing robustness margins for large scale systems is of immense practical importance. This explains why researchers have spent many man-centuries working to derive efficient numerical algorithms for computing robustness margins (a *man-century* refers to one man working 40 hours per week for one century) (Bartlett *et al.*, 1989; Beck and Doyle, 1992; Boyd *et al.*, 1994; Chiang and Safonov, 1992; Daniel *et al.*, 1986; de Gaston and Safonov, 1988; Fan *et al.*, 1991; Kharitonov, 1978; Kouvaritakis and Latchman, 1985a, b; Latchman, 1988; Newlin and Young, 1992; Pena and Sideris, 1990; Sideris, 1990; Sideris and Pena, 1989; Šiljak, 1989; Tierno and Young, 1992; Young and Doyle, 1990).

Computational complexity theory provides an approach for determining the level of accuracy and computational speed that are obtainable by algorithms for computing robustness margins, and as to which classes of algorithms may provide practical

* Author to whom all correspondence should be addressed.

robustness margin computation for large scale systems (Braatz *et al.*, 1993). One of the main goals of computational complexity theory is to classify computational problems as being *polynomial-time* or NP-*hard*. A computational problem is *polynomial-time* if the time needed to solve the problem is bounded by a fixed function that is polynomial in the quantity of data needed to define the problem. Examples of polynomial-time problems are linear programs, matrix inversion, and gain margin computation. Although the exact consequences of a problem being NP-*hard* is still a fundamental open question in the theory of computational complexity (Garey and Johnson, 1983; Papadimitriou and Steiglitz, 1982), it is generally accepted that a problem being NP-hard means that it *cannot* be solved in polynomial time in the worst case.

Braatz *et al.* (1993) have shown that exact μ-computation is NP-hard. Braatz (1996) has shown that even approximating μ within an a priori chosen tolerance is an NP-hard problem. The μ computation scales very poorly as the size of the problem increases. This motivates the development of polynomial-time algorithms for reducing the dimension of large scale uncertain systems, before applying μ-computation to the system to analyze its robustness. Algorithms are developed that can reduce the dimension of uncertain systems by orders of magnitude, which allows robustness margin computation to be practically performed for uncertain systems with much larger dimensionality. The theoretical implications of the algorithms are explored, and the algorithms are applied to the computation of robustness margins for some large scale systems.

2. Mathematical nomenclature

Doyle (1982) and Safonov (1982) derived the structured singular value, μ, to test for the robustness of uncertain systems. To calculate μ the uncertainty (the set of possible plants) must be modeled as norm bounded perturbations Δ_i on the nominal system M (see Fig. 1). Through weights each perturbation is normalized to be of size one

$$\|\Delta_i\|_\infty \equiv \sup_\omega \bar{\sigma}(\Delta_i) \leq 1, \tag{1}$$

where $\bar{\sigma}(\Delta_i)$ is the maximum singular value of matrix Δ_i. The perturbation Δ_i is complex for representing unmodeled dynamics, and real for representing parametric uncertainty. The set of real numbers will be denoted by $\mathscr{R}$, the set of complex numbers by $\mathscr{C}$, the $r \times r$ identity matrix by I_r, and the $r \times r$ matrix of zeros by 0_r. Given a vector $(r_1, \ldots, r_l)$ with $\sum_{i=1}^{l} r_i = n$, the perturbations which may occur at different locations in the system are collected in the block-diagonal matrix Δ

$$\Delta \equiv \{\text{diag}\{\delta_1^r I_{r_1}, \ldots, \delta_k^r I_{r_k}, \delta_{k+1}^c I_{r_{k+1}}, \ldots, \delta_m^c I_{r_m}, \Delta_{m+1}, \ldots, \Delta_l\} \,|\, \delta_i^r \in \mathscr{R}, \delta_i^c \in \mathscr{C}, \Delta_i \in \mathscr{C}^{r_i \times r_i}\}, \tag{2}$$

and the system is arranged to match the diagram in Fig. 1. The *block structure* refers to the vector $(r_1, \ldots, r_l)$. The nominal system M in Fig. 1 is determined by the nominal model, the size and nature of the uncertainty, the performance specifications, and the controller. Off-the-shelf programs construct M and the structure of Δ directly from the system description (Balas *et al.*, 1992; Chiang and Safonov, 1992; Russell and Braatz, 1996a, b).

The structured singular value $\mu_\Delta(M)$ is defined as

$$\mu_\Delta(M) = \begin{cases} 0 & \text{if there does not exist } \Delta \in \mathbf{\Delta} \text{ such that } \det(I - M\Delta) = 0, \\ \left(\min_{\Delta \in \mathbf{\Delta}} \{\bar{\sigma}(\Delta) | \det(I - M\Delta) = 0\} \right)^{-1} & \text{otherwise.} \end{cases} \tag{3}$$

Without loss of generality we have taken M and each subblock of Δ to be square. The time required to either approximately or exactly compute μ grows rapidly as the dimensions of M and Δ increase. The following section presents an exact algorithm that reduces the dimension of the M and Δ matrices in a manner that does not affect the value of μ.

3. Exact model reduction algorithm

The exact model reduction (EMR) algorithm is listed in Table 1. The following theorem states that the reduced model produced by the EMR algorithm has the same robustness margin as the original model (proof in appendix).

Theorem 1 (Exact model reduction). *Let $M \in \mathscr{C}^{n \times n}$ and $\Delta \in \mathbf{\Delta}$ where $\mathbf{\Delta}$ is defined in equation (2). Then $\mu_\Delta(M) = \mu_{\bar{\Delta}}(\bar{M})$, where $\bar{\mathbf{\Delta}}$ corresponds to the block-structure $\bar{r} = (\bar{r}_1, \ldots, \bar{r}_l)$ and $\bar{M}$ and $\bar{r}$ are obtained by performing the EMR algorithm (in Table 1).*

Theorem 1 implies that a method for computing the robustness margin μ for a large scale uncertain system described by M and Δ is to first apply the EMR

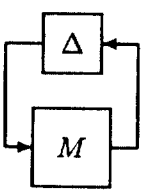

Fig. 1. System arrangement for calculating $\mu_\Delta(M)$

Table 1. Exact model reduction (EMR) algorithm

Step 1: Perform a singular value decomposition on M to obtain

$$M = \sum_{i=1}^{n} u_i \sigma_i v_i^H. \qquad (*)$$

Eliminate the terms of equation (*) corresponding to $\sigma_i = 0$ to obtain

$$M = \begin{bmatrix} U_1 \\ \vdots \\ U_l \end{bmatrix} \Sigma \begin{bmatrix} V_1 \\ \vdots \\ V_l \end{bmatrix}^H = U \Sigma V^H \qquad (**)$$

where U_i and V_i are $r_i \times r$ matrices and $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_r)$ where r is the rank of M.

Step 2: Let $W_i^U = (U^{U_i})^H$ where U^{U_i} is the unitary outer rotation matrix of the singular value decomposition of U_i, and let $W_i^V = (U^{V_i})^H$ where U^{V_i} is the unitary outer rotation matrix of the singular value decomposition of V_i. Row reduce each of the U_i and V_i matrices so that their last $r_i - r_i^U$ and $r_i - r_i^V$ rows contain only zeros by premultiplying U_i and V_i by W_i^U and W_i^V, respectively. Now, construct $W^U = \text{blockdiag}(W_1^U, \ldots, W_l^U)$ and $W^V = \text{blockdiag}(W_1^V, \ldots, W_l^V)$.

Step 3: Let $\bar{r}_i = \max\{r_i^U, r_i^V\}$ and $\bar{U}_i$ and $\bar{V}_i$ be the $\bar{r}_i \times r$ matrices consisting of the first $\bar{r}_i$ rows $W_i^U U_i$ and $W_i^V V_i$, respectively. Let

$$\bar{U} = \begin{bmatrix} \bar{U}_1 \\ \vdots \\ \bar{U}_l \end{bmatrix} \quad \text{and} \quad \bar{V} = \begin{bmatrix} \bar{V}_1 \\ \vdots \\ \bar{V}_l \end{bmatrix}.$$

Step 4: Let $\bar{W}_i^U$ and $\bar{W}_i^V$ be the $\bar{r}_i \times r$ matrices consisting of the first $\bar{r}_i$ rows of W_i^U and W_i^V, respectively.

Step 5: Define $\bar{W}^U = \text{blockdiag}\ (\bar{W}_1^U, \ldots, \bar{W}_{\bar{r}_m}^U, I_{\bar{r}_{m+1}}, \ldots, I_{\bar{r}_l})$ and $\bar{W}^V = \text{blockdiag}(\bar{W}_1^V, \ldots, \bar{W}_{\bar{r}_m}^V, I_{\bar{r}_{m+1}}, \ldots, I_{\bar{r}_l})$. Let $\bar{M} = \bar{U} \Sigma \bar{V}^H \bar{W}^V (\bar{W}^U)^H$ which is a $\bar{n} \times \bar{n}$ matrix where $\bar{n} = \sum_{i=1}^{l} \bar{r}_i$.

algorithm to compute an equivalent uncertain system with lower dimension matrices ($\bar{M}$ and $\bar{\Delta}$), and then compute μ for the lower dimension uncertain system. The EMR algorithm is stated and proved as an exact algorithm because the theoretical results in Section 5 will require this. Due to roundoff errors, the robustness margins for the original and reduced dimension uncertain systems will not be exactly equal. However, each step of the algorithm is very well conditioned, so that the difference between the robustness margins for the original and reduced dimension systems is negligible. This is supported by theoretical results in Section 4 and has been tested extensively using random matrices (some typical results are presented in Example 2 of Section 7).

It is sometimes of interest to compute perturbations of minimum magnitude that destabilize the uncertain system (i.e., a Δ that achieves the minimum in equation (3)). A control engineer may use this information to judge the likelihood that such parameter values would occur in practice. Off-the-shelf software applied to the computation of μ for the uncertain system of reduced dimension $\bar{M}$ provides as output a destabilizing perturbation of minimum magnitude $\bar{\Delta}$. To construct a destabilizing perturbation of minimum magnitude Δ for the original system M, sequentially add row and column vectors of zeros to $\bar{\Delta}$ corresponding to the row numbers removed in $W^U U$ and $W^V V$ in Step 3 of the EMR algorithm.

Theoretical bounds derived in Section 5 indicate that applying the EMR algorithm before performing μ-computation can substantially reduce the computational expense associated with robustness margin computation. The speedup provided by the EMR algorithm will be illustrated by application to a large scale paper machine control system in Example 1 of Section 7. Another important consideration in robustness margin computation that was discussed above concerns accuracy. Since smaller matrices result in better numerical conditioning during μ-computation (this was demonstrated using extensive simulations in Chapter 6 of Young (1993)), tighter bounds on the μ-values can be obtained by performing the EMR algorithm. This potential for improved accuracy in robustness margin computation is illustrated in Example 2 of Section 7.

In the EMR algorithm, rank deficiency of the matrix M and row rank deficiecies of U_i and V_i allow the dimensions of the μ-problem to be reduced. If there are no rank deficiencies in both Steps 1 and 3, no reduction in dimension will occur. In the instances where the EMR does not reduce the dimension sufficiently, the algorithm described in the next section may provide an uncertain system of even lower dimension that has approximately the same robustness margin as the original system.

4. Approximate model reduction algorithm

The strategy behind the EMR algorithm is to delete subspaces of M, U_i, and V_i that do not affect the value of the robustness margin. The approximate model

reduction (AMR) algorithm in Table 2 deletes subspaces of M, U_i, and V_i that have a very small effect on the value of the determinant given in the definition of μ (see equation (3)).

Given the computational complexity in computing μ exactly, it is just as difficult to compute the exact difference between the actual μ-values and the μ-values obtained by performing the AMR algorithm. However, this difference is reduced by removing only subspaces associated with singular values that are nearly zero.

In practice, μ-values are not computed directly, but rather the upper and lower bounds are computed. When the perturbations are complex, many of the norm and eigenvalue perturbation theorems (Golub and van Loan, 1983; Stewart and Sun, 1990) can be extended to provide estimates of the error between the actual μ-values and the μ-values obtained by performing the AMR algorithm. The extensions to these theorems are more conservative, due to the fact that optimizations are used in the computation of the lower and upper bounds for μ. For brevity, only the most useful of these conditions will be provided here.

Theorem 2 bounds the error caused by arbitrarily perturbing an M matrix (this is stated as a separate result as it may be of independent interest), while Corollary 1 applies Theorem 2 to place an upper bound on the error induced by the AMR algorithm (proofs in appendix).

Theorem 2. (μ-Upper bound perturbation theorem). *Let M, $E \in \mathbb{C}^{n\times n}$ and $\Delta \in \mathbf{\Delta}$, where $\mathbf{\Delta}$ contains the block-structure for complex blocks only ($r = (r_{k+1}, \ldots, r_l)$) as defined in equation (2). Then, the upper bound for $\mu_{\mathbf{\Delta}}(M)$ is calculated and defined by*

$$\bar{\mu}_{\mathbf{\Delta}}(M) \equiv \inf_{D\in\mathscr{D}} \bar{\sigma}(DMD^{-1}) \tag{4}$$

where the set $\mathscr{D}$ has the structure

$$\mathscr{D} \equiv \{\mathrm{diag}\{D_{k+1}, \ldots, D_m, d_{m+1} I_{r_{m+1}}, \ldots, d_l I_{r_l}\} \mid d_i \in \mathbb{R}; d_i > 0; D_i \in \mathbb{C}^{r_i \times r_i}; D_i = D_i^H > 0\}. \tag{5}$$

Assume the infima in equation (4) is achieved for $\bar{\mu}_{\mathbf{\Delta}}(M)$ and $\bar{\mu}_{\mathbf{\Delta}}(M+E)$. Then the folowing inequalities apply

$$\bar{\mu}_{\mathbf{\Delta}}(M+E) - \bar{\sigma}((D^{**})E(D^{**})^{-1}) \le \bar{\mu}_{\mathbf{\Delta}}(M) \le \bar{\mu}_{\mathbf{\Delta}}(M+E) + \bar{\sigma}((D^{*})E(D^{*})^{-1}), \tag{6}$$

$$\bar{\mu}_{\mathbf{\Delta}}(M+E) - \kappa(D^{**})\bar{\sigma}(E) \le \bar{\mu}_{\mathbf{\Delta}}(M) \le \bar{\mu}_{\mathbf{\Delta}}(M+E) + \kappa(D^{*})\bar{\sigma}(E), \tag{7}$$

where $\kappa(D) \equiv \bar{\sigma}(D)\bar{\sigma}(D^{-1})$, $D^* \equiv \arg\min_{D\in\mathscr{D}} \bar{\sigma}(D(M+E)D^{-1})$, and $D^{**} \equiv \arg\min_{D\in\mathscr{D}} \bar{\sigma}(DMD^{-1})$.

Corollary 1 (Approximate model reduction). *Define M, $\mathbf{\Delta}$, r, $\bar{\mu}$, and $\mathscr{D}$ as in Theorem 2, with the additional assumption that all of the uncertainties in $\mathbf{\Delta}$ are full block. Let the SVD of $M = U_+\Sigma_+V_+^H + U_-\Sigma_-V_-^H$, where Σ_- contains only those singular values set equal to zero by Modification 1 of the AMR algorithm, the matrices U_+ and V_+ contain only the singular vectors corresponding to Σ_+, and the matrices U_- and V_- contain only the singular vectors corresponding to Σ_-. Let $W^U U_+ = \hat{U}_+ + \check{U}_+$ and $V_+^H(W^V)^H = \hat{V}_+^H + \check{V}_+^H$ where $\check{U}_+$ and $\check{V}_+^H$ are the parts of $W^U U_+$ and $V_+^H(W^V)^H$, respectively, removed by Modification 2 of the AMR algorithm.*

Let $\bar{M}$ and $\bar{\mathbf{\Delta}}$ be the reduced matrices computed from the AMR algorithm, and $\bar{\mathscr{D}}$ be compatible with $\bar{\mathbf{\Delta}}$. Assume the infima in equation (4) are achieved for $\bar{\mu}_{\mathbf{\Delta}}(M)$ and $\bar{\mu}_{\bar{\mathbf{\Delta}}}(\bar{M})$, and define $D^{**} \equiv \arg\min_{D\in\mathscr{D}} \bar{\sigma}(DMD^{-1})$ and $\bar{D}^* \equiv \arg\min_{D\in\bar{\mathscr{D}}} \bar{\sigma}(\bar{D}\bar{M}\bar{D}^{-1})$. Then

$$\bar{\mu}_{\bar{\mathbf{\Delta}}}(\bar{M}) - \bar{\sigma}((D^{**})E(D^{**})^{-1}) \le \bar{\mu}_{\mathbf{\Delta}}(M) \le \bar{\mu}_{\bar{\mathbf{\Delta}}}(\bar{M}) - \bar{\sigma}((D^{*})E(D^{*})^{-1}), \tag{8}$$

$$\bar{\mu}_{\bar{\mathbf{\Delta}}}(\bar{M}) - \kappa(D^{**})\bar{\sigma}(E) \le \bar{\mu}_{\mathbf{\Delta}}(M) \le \bar{\mu}_{\bar{\mathbf{\Delta}}}(\bar{M}) + \kappa(\bar{D}^{*})\bar{\sigma}(E), \tag{9}$$

apply with E given by

$$E = -(W^U)^H(\hat{U}_+\Sigma_+\check{V}_+^H + \check{U}_+\Sigma_+\hat{V}_+^H + \check{U}_+\Sigma_+\check{V}_+^H)W^V - U_-\Sigma_-V_-^H, \tag{10}$$

and D^* corresponding to the block structure of $\mathbf{\Delta}$ with the scalar values $(d_{m+1}, \ldots, d_l)$ of $\bar{D}^*$.

Furthermore.

$$\bar{\mu}_{\bar{\mathbf{\Delta}}}(\bar{M}) - \kappa(D^{**})(3\bar{\sigma}(M) + \bar{\sigma}(\Sigma_-)) \le \bar{\mu}_{\mathbf{\Delta}}(M) \le \bar{\mu}_{\bar{\mathbf{\Delta}}}(\bar{M}) + \kappa(\bar{D}^{*})(3\bar{\sigma}(M) + \bar{\sigma}(\Sigma_-)). \tag{11}$$

The objective of the AMR algorithm is to reduce the size of the μ-problem while minimizing the inaccuracies in the μ-value. The tolerances in the AMR algorithm provide the trade-off between these two

Table 2. Approximate model reduction (AMR) algorithm

Perform the EMR algorithm with the following two modifications:

Modification 1: In Step 1 of the EMR algorithm, set equal to zero the singular values of M that are below some predefined tolerance.

Modification 2: In Step 2 of the EMR algorithm, set equal to zero the singular values of U_i and V_i that are below some predefined tolerance and decrease r_i^U and r_i^V accordingly.

objectives, and Corollary 1 can be useful for choosing the tolerance without solving the original μ-problem. For instance, equation (11) provides an upper bound on $\bar{\mu}_{\Delta}(M)$ in terms of $\bar{D}^*$, which is obtained directly from the output of the upper bound μ-computation for the reduced M using the μ-toolbox. Also note that, when the infima are achieved, equations (8) and (9) prove that the AMR algorithm is numerically robust (that is, the μ-value for the reduced problem approaches the μ-value for the original problem as E goes to zero). The conservatism of the bounds in Corollary 1 are illustrated in Example 2 of Section 7.

There will always be dynamics in the real system at very high frequencies that is not captured by the nominal model. Real parameter uncertainties do not completely capture such unmodeled dynamics, since the use of real parameter uncertainties requires an upper bound be placed on the number of states, and this upper bound is never known in practice (for example, how many states are precisely needed to represent stiction in a valve?). This implies that uncertain system representations should always include at least one complex perturbation that enters nontrivially in the closed loop system. For such systems, Packard and Pandey (1993) have shown that the robustness margin is a continuous function of the elements of M. Deleting subspaces corresponding to small singular values (as done in the EMR and AMR algorithms) has a small effect on the elements of M, and for problems where μ is a continuous function of M, has a small effect on the value of the robustness margin. This argument, which can be stated rigorously in terms of $\varepsilon - \delta$ statements (not given here for brevity), implies that the EMR and AMR algorithms are well-behaved model reduction techniques. This conclusion has been confirmed by extensive testing on both random matrices and descriptions of real systems (see Section 7 for examples).

5. Theoretical implications

Braatz *et al.* (1993) have shown that μ computation is NP-hard as a function of the dimension of the uncertain system matrix M. For practical computational reasons, it is natural to wonder whether there is a more natural parameter for which the computational complexity is a function of. For example, recent results regarding the μ-computation for systems with rank-one M matrices (Chen *et al.*, 1991; Young, 1993) led John Doyle at Caltech to recently hypothesize that the NP-hardness of μ-computation may be more directly related to the rank of M than the dimension of M. Although we will not show that the NP-hardness of μ-computation is a function only of the rank of M, we will show that the NP-hardness of μ-computation scales as a function of the rank of M multiplied by the number of perturbations.

Theorem 3 characterizes the computational complexity of the EMR algorithm.

Theorem 3 (Computational complexity of the EMR algorithm). Let $M \in \mathscr{C}^{n\times n}$ *and* $\Delta \in \mathbf{\Delta}$ *where* $\mathbf{\Delta}$ *is defined in equation* (2). *The number of flops required by the EMR algorithm is bounded by* $(26l + 12)n^3 - (2l + 4)n^2$.

The off-the-shelf software commonly used for robustness margin computation computes upper and lower bounds on μ (Balas *et al.*, 1992; Chiang and Safonov, 1992). The tightest polynomial-time-computable upper bound on μ of $M \in \mathscr{C}^{n\times n}$ can be formulated as a linear matrix inequality, whose computation grows as $\mathcal{O}(n^4 \ln(n))$ (Boyd *et al.*, 1994). To the authors' knowledge, there is no tighter bound on μ that can be computed with lower computational expense as a function of the dimension of M. Theorem 3 indicates that the EMR algorithm is bounded by a third-degree polynomial in the dimension of M (first order in the number of uncertainty blocks). Thus the overall computational expense of computing the tightest polynomial-time upper bound on μ can be reduced by performing the EMR algorithm before robustness margin computation. Typical speedups are illustrated on several examples in Section 7.

Theorem 4 provides an upper bound on the dimension of the reduced system resulting from application of the EMR algorithm.

Theorem 4 (Upper bound on dimension of reduced system). *The row (= column) dimension of* $\bar{M}$ *resulting from applying the EMR algorithm to* $M \in \mathscr{C}^{n\times n}$ *and* $\Delta \in \mathbf{\Delta}$ (*where* $\mathbf{\Delta}$ *is defined in equation* (2)) *satisfies the following inquality*:

$$\text{dimension}(\bar{M}) = \sum_{i=1}^{l} \bar{r}_i \le l \,\text{rank}(M). \qquad (12)$$

Theorems 3 and 4 imply that the EMR algorithm provides a polynomial-time transformation of a given μ-computation problem into an equivalent μ-computation problem of dimension less than or equal to the number of perturbation blocks multiplied by the rank of M. This strongly suggests that μ-computation problems for uncertain systems described by M matrices of low rank can be considered as being "easier" than for systems with M matrices of full rank.

6. Comparison with previous work

Rivera and Morari (1987, 1990) developed methods to reduce the number of plant or controller states, with the overall objective being to aid in the design of a low order robust controller. The objective of this paper, to efficiently compute robustness margins for systems with large numbers of inputs and outputs, is completely different.

The only existing algorithm for the model reduction of large scale systems with linear time-invariant uncertainties was proposed by Fan and Tits (1986). Their algorithm only applies to systems with complex perturbations; whereas the new algorithm holds for

both real and complex, and repeated and full block uncertainties. The new algorithm reduces to Fan and Tits' when only non-repeated complex uncertainties are considered (the weight in Table 1 $\bar{W}^V(\bar{W}^U)^H = I$ when there are only full block complex uncertainties), while having significant modifications for real and repeated complex uncertainties. Our method of proof (described in the appendix) is different, and is more intuitive than that used by Fan and Tits who used the numerical range formulation of the robustness margin (Fan and Tits, 1986).

Beck *et al.* (Beck, 1994; Beck and Doyle, 1995; Wang *et al.*, 1991) have developed model reduction techniques that hold for general noncommutative operators. It is the opinion of the authors that their work is seminal. However, it is expected that the extent of dimensionality reduction achievable by their techniques to systems with linear time-invariant uncertainties (which are commutative) is limited. Also, their algorithms are based on linear matrix inequalities, whose solution is significantly more computationally expensive than for our algorithms which only requires a limited number of SVD calculations.

7. Examples

7.1. Example 1: Paper machine control system

Most paper machine models used for cross directional controller design have the form (see (Laughlin *et al.*, 1993; Braatz, 1997; Braatz *et al.*, 1996) for details)

$$P(s) = p(s)P_{CD}, \tag{13}$$

where $p(s)$ is a transfer function representing scalar dynamics, and P_{CD} is a static interaction matrix. Most reported interaction matrices are Toeplitz symmetric

$$P_{CD} = \underbrace{\begin{pmatrix} p_1 & p_2 & \cdots & p_m & 0 & \cdots & \cdots & 0 \\ p_2 & p_1 & p_2 & \cdots & p_m & \ddots & \ddots & \vdots \\ \vdots & p_2 & p_1 & p_2 & \cdots & \ddots & \ddots & \vdots \\ p_m & \vdots & p_2 & \ddots & \ddots & \vdots & p_m & 0 \\ 0 & p_m & \vdots & \ddots & \ddots & p_2 & \vdots & p_m \\ \vdots & \ddots & \ddots & \cdots & p_2 & p_1 & p_2 & \vdots \\ \vdots & \ddots & \ddots & p_m & \cdots & p_2 & p_1 & p_2 \\ 0 & \cdots & \cdots & 0 & p_m & \cdots & p_2 & p_1 \end{pmatrix}}_{n \times n}, \tag{14}$$

where m specifies the spatial extent of coupling across the paper machine, and n is the number of actuators and sensor lanes (most modern paper machines have edge effects, and more sensor lanes than actuators, but a further discussion of paper machine models would be outside of the scope of this paper). For our nominal model we will use the $m = 10$ interaction parameters for a paper board machine model (Karlsson and Haglund, 1983):

$$p_1 = 1; \quad p_2 = 0.9; \quad p_3 = 0.7; \quad p_4 = 0.8;$$

$$p_5 = 1; \quad p_6 = 0.6; \quad p_7 = -0.5;$$

$$p_8 = -0.4; \quad p_9 = -0.2; \quad p_{10} = -0.2. \tag{15}$$

It will be assumed that there are 101 actuators ($n = 101$). The nominal scalar dynamics are assumed similar to those given by Laughlin *et al.* (1993)

$$p(s) = \frac{e^{-\theta s}}{\tau s + 1}, \tag{16}$$

where the time delay $\theta = 1$ and the open loop time constant $\tau = 1$. Due to a low quantity of high quality experimental data (Featherstone and Braatz, 1995), nominal paper machine models always have a significant amount of uncertainty associated with them. We will model this uncertainty as being associated with both the input and output of the process, i.e. the true process is assumed to be an element of the set

$$(I + W_O\Delta_O)P(s)(I + W_I\Delta_I), \tag{17}$$

where the uncertainty weights are given by

$$W_O = W_I = \frac{0.5s + 0.1}{0.5s + 1} I_n. \tag{18}$$

Each uncertainty weight allows for steady state inaccuracies of up to 10% and high frequency uncertainties of 100%, with crossover frequencies of $\omega_l = 0.2$ and $\omega_h = 2$.

It is well known to experts in paper machine controls that the poor conditioning of paper machine makes it impossible to control all n possible disturbance directions (Featherstone and Braatz, 1995; Kjaer *et al.*, 1994; Kristinsson and Dumont, 1996). Researchers have proposed various methods to select which directions to control, including methods based on Gram polynomials (Kjaer *et al.*, 1994; Kristinsson and Dumont, 1996), splines (Halouskova *et al.*, 1993), and columns of the input rotation matrix of a singular value decomposition of the interactions matrix (Braatz and Featherstone, 1995; Featherstone and Braatz, 1995, 1997). The robust controller design procedure of Braatz and VanAntwerp (Braatz and VanAntwerp, 1996) results in the controller

$$K(s) = k(s)K_{CD}, \tag{19}$$

where $k(s)$ represent scalar dynamics given by

$$k(s) = \frac{\tau s + 1}{\lambda s}, \tag{20}$$

with the desired closed loop time constant $\lambda = 2$, and K_{CD} is a static decoupling matrix constructed from the singular value decomposition of $P_{CD} = U\Sigma V^T$. More specifically,

$$K_{CD} = V\begin{bmatrix} \hat{\Sigma}^{-1} & 0 \\ 0 & 0 \end{bmatrix} U^T, \tag{21}$$

where $\hat{\Sigma}$ is equal to the upper left submatrix of Σ with fifteen rows and columns (these rows and columns correspond to the singular values of P_{CD} with the largest magnitude).

Now we will apply the EMR algorithm for computing robust stability margins for the uncertain system described by the nominal plant (14)–(16), uncertainty set (17)–(18), and controller (19)–(21). Constructing the M matrix by hand as shown in textbooks and monographs (Morari and Zafiriou, 1989; Skogestad and Postlethwaite, 1996) or using off-the-shelf programs (Balas *et al.*, 1992; Chiang and Safonov, 1992; Russell, 1996; Russell and Braatz, 1996a, b) results in

$$M = \begin{bmatrix} -W_O P(I+KP)^{-1}K & W_O P(I+KP)^{-1} \\ -W_I K(I+PK)^{-1} & -W_I KP(I+KP)^{-1} \end{bmatrix}, \tag{22}$$

where the elements of M are a function of frequency ($s = j\omega$).

Using the MATLAB μ-tools toolbox (Balas *et al.*, 1992), lower bounds via the power iteration and upper bounds for the robustness margins of M were computed for various uncertainty structures (structure of Δ_O/structure of Δ_I): (1) repeated real/full complex, (2) repeated complex/full complex, and (3) full complex/full complex. The EMR algorithm reduced the block structure from (101, 101) to (30, 30) for all frequencies between [0.1, 10] for each of the uncertainty descriptions. The upper and lower bounds for the robustness margins calculated before and after the application of the EMR algorithm for the repeated complex/full complex and full complex/full complex uncertainty structures are the same to five significant figures for the entire frequency range (see Fig. 2). For the repeated real/full complex case (Fig. 3), the lower bound for μ using the EMR algorithm is somewhat better than the lower bound before using the EMR algorithm. The largest improvement of the lower bound for μ by implementing the EMR algorithms is 0.29 at frequency $\omega = 1.26$, while the worst case degradation of the lower bound by implementing the EMR algorithm is 0.077 at frequency $\omega = 0.95$.

Figures 4–6 indicate that the robustness margin computations using the EMR algorithm is less computationally expensive than the computations without the EMR algorithm. For the repeated real/full complex uncertainty structure, when the bounds are not tight there is roughly an order of magnitude difference as can be concluded from Figs. 3 and 6.

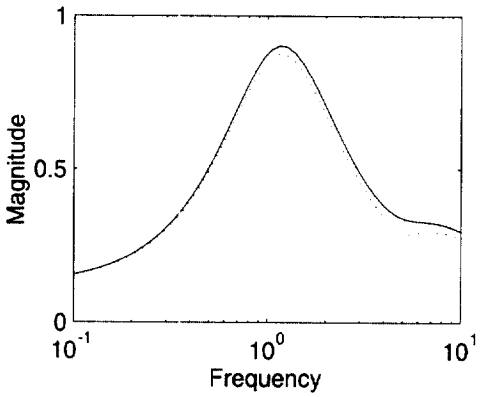

Fig. 2. Plots of $\mu(\omega)$ for M (the upper and lower bounds for $\mu(\omega)$ were within 5 significant digits). Solid curve is for the full complex/full complex uncertainty structure, and the dotted curve is for the repeated complex/full complex uncertainty description.

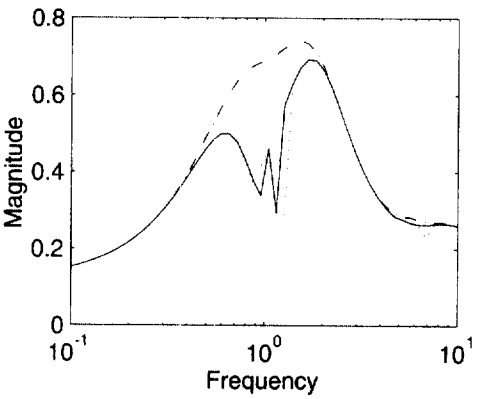

Fig. 3. Plots of $\mu(\omega)$ bounds for M with the repeated real/full complex uncertainty structure. Dotted curve is the lower bound using M. Solid curve is the lower bound using the EMR algorithm on M. Dash-dotted curve is the upper bound using either M or using the EMR algorithm on M.

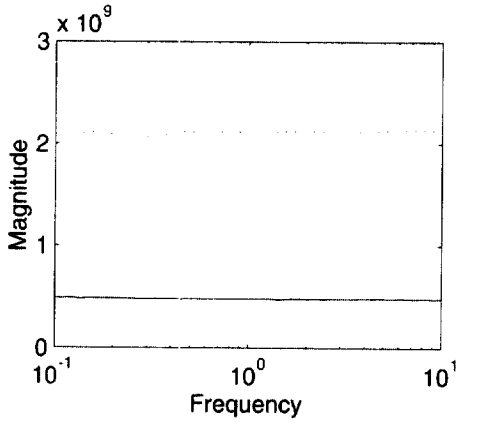

Fig. 4. Flops required to compute the lower and upper $\mu(\omega)$ bounds for the repeated complex/full complex uncertainty structure. Dotted curve represents the flops using M. Solid curve represents the total flops required to reduce M using the EMR algorithm and to compute the $\mu(\omega)$ bounds.

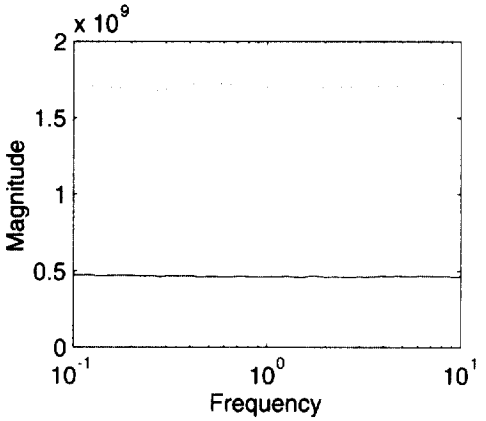

Fig. 5. Flops required to compute the lower and upper $\mu(\omega)$ bounds for the full complex/full complex uncertainty structure. Dotted curve represents the flops using M. Solid curve represents the total flops required to reduce M using the EMR algorithm and to compute the $\mu(\omega)$ bounds.

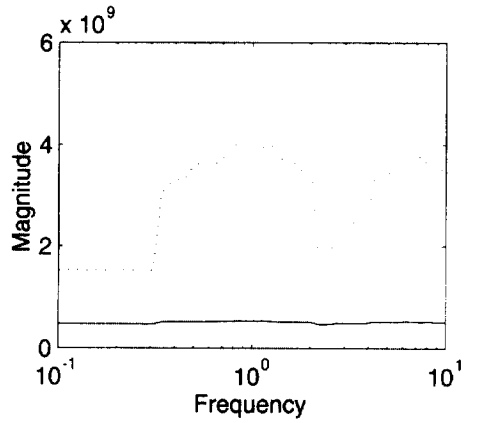

Fig. 6. Flops required to compute the lower and upper $\mu(\omega)$ bounds for the repeated real/full complex uncertainty structure. Dotted curve represents the flops using M. Solid curve represents the total flops required to reduce M using the EMR algorithm and to compute the $\mu(\omega)$ bounds.

7.2. Example 2: Random matrices

The EMR and AMR algorithms have been applied to numerous random matrices to test their performance. In this section, the EMR and AMR algorithms are tested on random matrices. The performance of the EMR algorithm on a rank 2 matrix is shown in Table 3. For all of the uncertainty descriptions listed in the table, μ-computations utilizing the EMR algorithm result in tighter bounds than μ-computations without the EMR algorithm. Also, the required number of flops is reduced by over an order of magnitude by performing the EMR algorithm before using standard software to compute μ.

The AMR algorithm is tested on a full rank matrix in Table 4. Table 4 shows that the combination of the AMR algorithm with a standard μ algorithm produces tighter bounds and requires less computational expense than the application of μ software alone. Extensive applications to random matrices have demonstrated that the EMR/AMR algorithms are numerically robust, and produce similar results as shown in Tables 3 and 4.

Table 5 illustrates the application of Corollary 1. It shows that practical error bounds for the upper bound computation of μ are obtained by using tolerances [0.001, 0.1]. The computed upper bounds with and without the AMR algorithm are identical up to five significant digits, indicating that the AMR algorithm performs much better than the bounds provided by Corollary 1.

8. Conclusions

Exact and approximate polynomial-time algorithms were developed for reducing the dimension of large scale uncertain systems. The polynomial-time algorithms rely primarily on singular value decompositions, which can be computed accurately and effectively using well-conditioned public-domain computer algorithms. The exact algorithm is shown to have direct implications for understanding the inherent

Table 3. Comparison of μ-computations with and without the EMR algorithm for the 100×100 rank 2 matrix $M = UV$, with randomly-generated U and V given in equations (44) and (45)

Block structure before EMR	Block structure after EMR	μ-bounds without EMR	μ-bounds with EMR	Flops without EMR	Flops[a] with EMR
$(25^r, 25^r, 25^c, 25^f)$	$(2^r, 2^r, 2^c, 2^f)$	0.584 0.471	0.579 0.567	1.28×10^9	5.46×10^7
$(25^r, 25^c, 25^c, 25^f)$	$(2^r, 2^c, 2^c, 2^f)$	0.794 0.778	0.794 0.793	1.18×10^9	5.44×10^7
$(50^r, 25^c, 25^f)$	$(2^r, 2^c, 2^f)$	0.567 0.473	0.567 0.565	1.09×10^9	5.48×10^7

[a] The total number of flops required to reduce M and to compute μ.

Note: Superscripts r, c and f stand for repeated real, repeated complex and full complex uncertainties, respectively.

Table 4. Comparison of μ-computations with and without the AMR algorithm for the full rank matrix $\hat{M} = UV + \hat{U}\hat{V}^H$, where $\hat{U}$ and $\hat{V}$ are random 100×98 matrices with the real and complex part of each element bounded between 0 and 1 (this matrix is available from http://brahms.scs.uiuc.edu). In the application of the AMR algorithm (see Table 2), the tolerance in Modification 1 is $0.1\sigma_{max}$ where σ_{max} is the maximum singular value of $\hat{M}$, and the tolerance in Modification 2 is 0.1

Block structure before AMR	Block structure after AMR	μ-bounds without AMR	μ-bounds with AMR	Flops without AMR	Flops[a] with AMR
$(25^r, 25^r, 25^c, 25^f)$	$(1^r, 1^r, 1^c, 1^f)$	0.820 0.675	0.836 0.797	1.58×10^9	4.16×10^7
$(25^r, 25^c, 25^c, 25^f)$	$(1^r, 1^c, 1^c, 1^f)$	0.115 0.112	0.115 0.115	1.21×10^9	4.15×10^7
$(50^r, 25^c, 25^f)$	$(1^r, 1^c, 1^f)$	0.802 0.675	0.800 0.797	1.11×10^9	4.22×10^7

[a] The total number of flops required to reduce $\hat{M}$ and to compute μ.

Note: Superscripts r, c and f stand for repeated real, repeated complex and full complex uncertainties, respectively.

Table 5. Application of Corollary 1 (using equation (10) in equation (6)) to be the upper bound μ-computation of the full rank matrix $\hat{M} = UV + \hat{U}\hat{V}^H$, where $\hat{U}$ and $\hat{V}$ are random 100×98 matrices with the real and complex part of each element bounded between 0 and 1 (this matrix is available from http://brahms.scs.uiuc.edu). The block structure of the uncertainty is the full-block structure (25, 25, 25, 25). The tolerance value (t) indicates that the tolerance in Modification 1 is $t\sigma_{max}$ where σ_{max} is the maximum singular value of $\hat{M}$, and the tolerance in Modification 2 is t.

Tolerance t	Block structure after AMR	Upper μ Bound without AMR	Upper μ bound with AMR	Error bounds
0.001	(25, 25, 25, 25)	1.664	1.664	[1.662, 1.666]
0.01	(3, 3, 3, 3)	1.664	1.664	[1.660, 1.668]
0.1	(1, 1, 1, 1)	1.664	1.664	[1.506, 1.822]

computational complexity of robustness margin computation. Theoretical bounds and numerical examples indicate that the algorithms allow robustness margin computation to be practically performed on uncertain systems of very high dimensionality. Application to a realistic description of a large scale paper machine control system demonstrates the utility of the algorithms.

References

Balas, G.J., Doyle, J.C., Glover, K., Packard, A.K. and Smith, R.S.R. (1992) *μ-Analysis and Synthesis Toolbox (μ-Tools): Matlab Functions for the Analysis and Design of Robust Control Systems*. The MathWorks, Natick, Massachusetts.

Bartlett, A.C., Hollot, C.V. and Lin, H. (1989) Root locations of an entire polytope of polynomals: it suffices to check the edges. *Math. Control Signals Systems* **1**, 61–71.

Beck, C. (1994) Minimality of uncertain systems and IQCs. In *Proc. IEEE Conf. on Decision and Control*, Lake Buena Vista, Florida, pp. 3068–3073.

Beck, C. and Doyle, J. (1995) Realizations of uncertain systems and formal power series. In *Proc. American Control Conf.*, Seattle, Washington, pp. 1481–1485.

Beck, C. and Doyle, J.C. (1992) Mixed μ upper bound computation using LMI computation. In *Proc. IEEE Conf. on Decision and Control*, pp. 3187–3192.

Boyd, S., El Ghaoui, L., Feron, E. and Balakrishnan, V. (1994) *Linear Matrix Inequalities in System and Control Theory, Studies in Applied Mathematics*, vol. 15. SIAM, Philadelphia, PA.

Braatz, R.D. (1996) Robustness margin computation for large scale systems. In *AIChE Annual Meeting*, Chicago, Illinois, Paper 141d.

Braatz, R.D. (1997) The current status of sheet and film process control. In *Chemical Process Control*, AIChE Symp. Series. AIChE, New York (in press).

Braatz, R.D. and Featherstone, A.P. (1995) Identification and control of large scale paper machines. In *Weyerhaeuser Workshop on Modeling and Control of Kamyr Digesters and Paper Machines*, Tacoma, Washington.

Braatz, R.D., Ogunnaike, B.A. and Featherstone, A.P. (1996) Identification, estimation, and control of sheet and film processes. In *Proc. IFAC World Congress*, San Francisco, California, pp. 319–324.

Braatz, R.D. and VanAntwerp, J.G. (1996) Robust cross-directional control of large scale paper machines. In *Proc. IEEE International Conf. on Control Applications*, Dearborn, Michigan, pp. 155–160.

Braatz, R.D., Young, P.M., Doyle, J.C. and Morari, M. (1993) Computational complexity of μ calculation. In *Proc. American Control Conf.*, San Francisco, California, pp. 1682–1683.

Brosilow, C.B. (1979) The structure and design of Smith predictors from the view point of inferential control. In *Proc. Joint Automation and Control Conf.*, p. 288.

Chen, J., Fan, M.K.H. and Nett, C.N. (1991) The structured singular value and stability of uncertain polynomials: a missing link. *Control of Systems wth Inexact Dynamic Models, ASME*, pp. 15–23.

Chiang, R.Y. and Safonov, M.G. (1992) *Robust Control Toolbox: For Use with MATLAB*. The MathWorks, Natick, Massachusetts.

Daniel, R.W., Kouvaritakis, B. and Latchman, H. (1986) Principal direction alignment: A geometric framework for the complete solution to the μ-problem. *IEE Proc. Pt. D* **133**, 45–56.

de Gaston, R.R.E. and Safonov, M.G. (1988) Exact calculation of the multiloop stability margin. *IEEE Trans. Automat. Control* **33**, 156–171.

Doyle, J.C. (1982) Analysis of feedback systems with structured uncertainties. *IEE Proc. Pt. D* **129**, 242–250.

Fan, M.K.H. and Tits, A.L. (1986) Characterization and efficient computation of the structured singular value. *IEEE Trans. Automat. Control* **31**, 734–743.

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

Featherstone, A.P. and Braatz, R.D. (1995) Control relevant identification of sheet and film processes. In *Proc. American Control Conf.* Seattle, Washington, pp. 2692–2696.

Featherstone, A.P. and Braatz, R.D. (1997) Model-oriented identification of sheet and film processes. *A.I.Ch.E. J.* **43**, 1989–2001.

Garey, M.R. and Johnson, D.S. (1983) *Computers and Intractability: A Guide to NP-Completeness*. Freeman, New York.

Golub, G.H. and van Loan, C.F. (1983) *Matrix Computations*. Johns Hopkins University. Press, Baltimore, Maryland.

Halouskova, A., Karny, M. and Nagy, I. (1993) Adaptive cross-direction control of paper basis weight. *Automatica* **29**, 425–429.

Karlsson, H. and Haglund, L. (1983) Optimal cross-direction basis weight and moisture profile control on paper machines. In *Proc. 3rd Int. Pulp and Paper Process Control Symp.*, Vancouver, BC, Canada.

Kharitonov, V.L. (1978) Asymptotic stability of an equilibrium position of a family of systems of linear differential equations. *Differential'nye Uraveniya* **14**, 1483–1485.

Kjaer, A.P., Heath, W.P. and Wellstead, P.E. (1994) Identification of cross-directional behaviour in web production: techniques and experience. In *Proc. IFAC Symp. on System Identification*, Copenhagen, Denmark.

Kouvaritakis, B. and Latchman, H. (1985a) Necessary and sufficient stability condition for systems with structured uncertainties: The major principal direction alignment principle. *Int. J. Control* **42**, 575–598.

Kouvaritakis, B. and Latchman, H. (1985b) Singular-value and eigenvalue techniques in the analysis of systems with structured perturbations. *Int. J. Control* **41**, 1381–1412.

Kristinsson, K. and Dumont, G.A. (1996) Paper machine cross directional basis weight using Gram polynomials. *Automatica* **32**, 533–548.

Latchman, H.A. (1988) Analysis of systems with element-by-element bounded uncertainties. In *Proc. American Control Conf.*, pp. 378–383.

Laughlin, D., Morari, M. and Braatz, R.D. (1993) Robust performance of cross-directional basis-weight control in paper machines. *Automatica* **29**, 1395–1410.

Morari, M. and Zafiriou, E. (1989) *Robust Process Control*. Prentice-Hall, Englewood Cliffs, New Jersey.

Newlin, M.P. and Young, M.P. (1992) Mixed μ problems and branch and bound techniques. In *Proc. IEEE Conf. on Decision and Control*, pp. 3175–3180.

Ogunnaike, B.A. and Ray, W.H. (1994) *Process Dynamics, Modeling, and Control*. Oxford University Press, New York.

Packard, A. and Pandey, P. (1993) Continuity properties of the real/complex structured singular value. *IEEE Trans. Automat. Control* **38**, 415–428.

Palmor, Z.J. and Shinnar, R. (1981) Design of advanced process controllers. *AI.Ch.E. J.* **27**, 793–805.

Papadimitriou, C.H. and Steiglitz, K. (1982) *Combinatorial Optimization: Algorithms and Complexity*. Prentic-Hall, Englewood Cliffs, New Jersey.

Pena, R.S.S. and Sideris, A. (1990) Robustness with real parametric and structured complex uncertainty, *Int. J. Control* **52**, 753–765.

Rivera, D.E. and Morari, M. (1987) Control-relevant model reduction problems for *SISO* H_2, H_∞, and μ-controller synthesis. *Int. J. Control* **46**, 505–527.

Rivera, D.E. and Morari, M. (1990) Plant and controller reduction problems for closed-loop performance. *Automatica* **26**, 361–369.

Russell, E.L. (1996). Multidimensional realization and model reduction of large scale uncertain systems. M.S. thesis, University of Illinois, Urbana, Illinos.

Russell, E.L. and Braatz, R.D. (1996a) Analysis of large scale systetms with model uncertainty, actuator and state constraints, and time delays. In *AIChE Annual Meeting*, Chicago, Illinois. Paper 45a.

Russell, E.L. and Braatz, R.D. (1996b) MDR – A code for multidimensional realization and model reduction. University of Illinois, Urbana, elr@bach.scs.uiuc.edu. computer software.

Safonov, M.G. (1982) Stability margins of diagonally perturbed multivariable feedback systems. *IEE Proc. Pt. D* **129**, 251–256.

Seborg, D.E., Edgar, T.F. and Mellichamp, D.A. (1989) *Process Dynamics and Control*. Wiley, New York.

Sideris, A. (1990) Elimination of frequency search from robustness tests. In *Proc. IEEE Conf. on Decision and Control*, pp. 41–45.

Sideris, A. and Peaa, R.S. (1989) Fast computation of the multivariable stability margin for real interrelated uncertain parameters. *IEEE Trans. Automat. Control* **34**, 1272–1276.

Skogestad, S. and Postlethwaite, I. (1996) *Multivariable Feedback Control: Analysis and Design*. Wiley, New York.

Stephanopoulos, G. (1983) Synthesis of control systems for chemical plants: A challenge for creativity. *Comput. Chem. Engng* **7**, 331–365.

Stewart, G.W. and Sun, J.G. (1990) *Matrix Perturbation Theory*. Academic Press, Boston, Massachusetts.

Tierno, J.E. and Young, P.M. (1992) An improved μ lower bound via adaptive power iteration. In *Proc. IEEE Conf. on Decision and Control*, pp. 3181–3186.

Šiljak, D.D. (1989) Parameter space methods for robust control design: a guided tour. *IEEE Trans. Automat. Control* **34**, 674–688.

Wang, W., Doyle, J., Beck, C., and Glover, K. (1991) Model reduction of LFT systems. In *Proc. IEEE Conf. on Decision and Control*, Brighton, U.K.

Young, P.M. (1993) Robustness with parametric and dynamic uncertainties. PhD thesis, California Institute of Technology, Pasadena.

Young, P.M. and Doyle, J.C. (1990) Computation of the structured singular value with real and complex uncertanities. In *Proc. IEEE Conf. on Decision and Control* pp. 1230–1235.

Appendix A

A.1. Proof of Theorem 1

Let us consider $\mu_\Delta(M)$ when there exists a $\Delta \in \mathbf{\Delta}$ such that $\det(I - M\Delta) = 0$. Then

$$\mu_\Delta(M) = \left(\min_{\Delta \in \mathbf{\Delta}} \{\bar{\sigma}(\Delta) | \det(I - M\Delta) = 0\} \right)^{-1}. \quad (23)$$

By using the singular value decomposition as defined in equation (**) of Table 1 with the unitary matrices W^U and W^V as defined in the algorithm,

$$\mu_\Delta(M) = \mu_\Delta((W^U)^H W^U U \Sigma V^H (W^V)^H W^V). \quad (24)$$

Let $W^U_{rc} = \text{blockdiag}(W^U_1, \ldots, W^U_m, I_{r_{m+1}}, \ldots, I_{r_l})$ and $W^U_\Delta = \text{blockdiag}(I_{r_1}, \ldots, I_{r_m}, W^U_{m+1}, \ldots, W^U_l)$. Let W^V_{rc} and W^V_Δ be defined similarly. By using the fact that $\mu(M) = \mu(MU) = \mu(UM)$ for any unitary matrix U with the same structure as Δ (Young, 1993),

$$\begin{aligned} \mu_\Delta(M) &= \mu_\Delta((W^U_\Delta)^H (W^U_{rc})^H W^U U \Sigma V^H (W^V)^H W^V_{rc} W^V_\Delta) \\ &= \mu_\Delta((W^U_{rc})^H W^U U \Sigma V^H (W^V)^H W^V_{rc}). \end{aligned} \quad (25)$$

Now with $\bar{U}_i$ and $\bar{V}_j$ defined as in the algorithm, we have

$$[W^U U \Sigma V^H (W^V)^H]_{ij} = \begin{bmatrix} \bar{U}_i \Sigma \bar{V}_j^H & 0 \\ 0 & 0 \end{bmatrix}_{ij}, \quad (26)$$

where the ijth submatrix has dimensions $r_i \times r_j$. Then, the sequence

$$\mathscr{S} = \bigcup_{j=1}^{l} \begin{cases} \left\{ \bar{r}_j + 1 + \sum_{i=1}^{j-1} r_i, \ldots, r_j + \sum_{i=1}^{j-1} r_i \right\} & \text{if } r_j > \bar{r}_j, \\ \{\emptyset\} & \text{if } r_j = \bar{r}_j, \end{cases} \quad (27)$$

lists row and column numbers of $W^U U \Sigma V^H (W^V)^H$ which contain only zeros. Define the matrix $\tilde{W}^V$ equal to W^V_{rc} with the corresponding rows in $\mathscr{S}$ equal to zero, and the matrix $(\tilde{W}^U_{rc})^H$ equal to $(W^U_{rc})^H$ with the corresponding columns in $\mathscr{S}$ up to $\sum_{i=1}^{m} r_i$ equal to zero. Then, $(W^U_{rc})^H W^U U \Sigma V^H (W^V)^H W^V_{rc} = (\tilde{W}^U_{rc})^H W^U U \Sigma V^H (W^V)^H \tilde{W}^V$, and since $(\tilde{W}^U_{rc})^H$ commutes with Δ

$$\begin{aligned} \mu_\Delta(M) &= \left(\min_{\Delta \in \mathbf{\Delta}} \{\bar{\sigma}(\Delta) | \det(I - (\tilde{W}^U_{rc})^H W^U U \Sigma V^H \right. \\ &\qquad \left. \times (W^V)^H \tilde{W}^V \Delta) = 0\} \right)^{-1} \\ &= \left(\min_{\Delta \in \mathbf{\Delta}} \{\bar{\sigma}(\Delta) | \det(I - W^U U \Sigma V^H (W^V)^H \right. \\ &\qquad \left. \times \tilde{W}^V (\tilde{W}^U_{rc})^H \Delta) = 0\} \right)^{-1} \\ &= \left(\min_{\Delta \in \mathbf{\Delta}} \{\bar{\sigma}(\Delta) | \det(I - W^U U \Sigma V^H (W^V)^H \right. \\ &\qquad \left. \times \tilde{W}^V (\tilde{W}^U)^H \Delta) = 0\} \right)^{-1}, \end{aligned} \quad (28)$$

where $(\tilde{W}^U)^H$ is equal to $(W^U_{rc})^H$ with the columns in $\mathscr{S}$ equal to zero. Note that $W^U U \Sigma V^H (W^V)^H \tilde{W}^V (\tilde{W}^U)^H$ has the same structure as the matrices given in equation (26). Define $\hat{\Delta}$ equal to Δ with rows in $\mathscr{S}$ equal to zeros, and $\tilde{\Delta}$ equal to $\hat{\Delta}$ with columns in $\mathscr{S}$ equal to zeros. Then,

$$\begin{aligned} &\det(I - W^U U \Sigma V^H (W^V)^H \tilde{W}^V (\tilde{W}^U)^H \Delta) \\ &= \det(I - W^U U \Sigma V^H (W^V)^H \tilde{W}^V (\tilde{W}^U)^H \hat{\Delta}) \\ &= \det(I - \hat{\Delta} W^U U \Sigma V^H (W^V)^H \tilde{W}^V (\tilde{W}^U)^H) \\ &= \det(I - \tilde{\Delta} W^U U \Sigma V^H (W^V)^H \tilde{W}^V (\tilde{W}^U)^H) \\ &= \det(I - W^U U \Sigma V^H (W^V)^H \tilde{W}^V (\tilde{W}^U)^H \tilde{\Delta}). \end{aligned} \quad (29)$$

Immediate from equations (28) and (29),

$$\begin{aligned} \mu_\Delta(M) = \left(\min_{\tilde{\Delta} \in \tilde{\mathbf{\Delta}}} \{\bar{\sigma}(\tilde{\Delta}) | \det(I - W^U U \Sigma V^H \right. \\ \left. \times (W^V)^H \tilde{W}^V (\tilde{W}^U)^H \tilde{\Delta}) = 0\} \right)^{-1}, \end{aligned} \quad (30)$$

where $\tilde{\Delta}$ corresponds to the original block structure $(r_1, \ldots, r_l)$ with the rows and columns in $\mathscr{S}$ replaced by zeros.

Note that when $i = j$

$$[I - W^U U \Sigma V^H (W^V)^H \tilde{W}^V (\tilde{W}^U)^H \tilde{\Delta}]_{ij} = \begin{cases} \begin{bmatrix} I_{r_i} - \bar{U}_i \Sigma \bar{V}_j^H \bar{W}_j^V (\bar{W}_j^U)^H \delta_j^r I_{\bar{r}_j} & 0 \\ 0 & I_{r_i - \bar{r}_i} \end{bmatrix}_{ij} & \forall j \le k, \\ \begin{bmatrix} I_{r_i} - \bar{U}_i \Sigma \bar{V}_j^H \bar{W}_j^V (\bar{W}_j^U)^H \delta_j^c I_{\bar{r}_j} & 0 \\ 0 & I_{r_i - \bar{r}_i} \end{bmatrix}_{ij} & \forall k+1 \le j \le m, \\ \begin{bmatrix} I_{r_i} - \bar{U}_i \Sigma \bar{V}_j^H \bar{W}_j^V (\bar{W}_j^U)^H \tilde{\Delta}_j I_{\bar{r}_j} & 0 \\ 0 & I_{r_i - \bar{r}_i} \end{bmatrix}_{ij} & \forall m+1 \le j \le l \end{cases} \tag{31}$$

and when $i \neq j$

$$[I - W^U U \Sigma V^H (W^V)^H \tilde{W}^V (\tilde{W}^U)^H \tilde{\Delta}]_{ij} = \begin{cases} \begin{bmatrix} -\bar{U}_i \Sigma \bar{V}_j^H \bar{W}_j^V (\bar{W}_j^U)^H \delta_j^r I_{\bar{r}_j} & 0 \\ 0 & 0 \end{bmatrix}_{ij} & \forall j \le k, \\ \begin{bmatrix} -\bar{U}_i \Sigma \bar{V}_j^H \bar{W}_j^V (\bar{W}_j^U)^H \delta_j^c I_{\bar{r}_j} & 0 \\ 0 & 0 \end{bmatrix}_{ij} & \forall k+1 \le j \le m, \\ \begin{bmatrix} -\bar{U}_i \Sigma \bar{V}_j^H \bar{W}_j^V (\bar{W}_j^U)^H \tilde{\Delta}_j I_{\bar{r}_j} & 0 \\ 0 & 0 \end{bmatrix}_{ij} & \forall m+1 \le j \le l, \end{cases} \tag{32}$$

where $\bar{W}_i^U$ and $\bar{W}_i^V$ are defined in the algorithm, $\tilde{\Delta}_j$ corresponds to $\tilde{\Delta}$ as Δ_j corresponds to Δ, and the ijth submatrix has dimensions $r_i \times r_j$. Therefore, by performing a Laplace expansion of $[I - W^U U \Sigma V^H (W^V)^H \tilde{W}^V (\tilde{W}^U)^H \tilde{\Delta}$ about the rows which have a single nonzero element of value one, it is concluded that $\det(I - W^U U \Sigma V^H (W^V)^H \tilde{W}^V (\tilde{W}^U)^H \tilde{\Delta}) = \det(I - \bar{U} \Sigma \bar{V}^H \bar{W}^V (\bar{W}^U)^H \bar{\Delta})$ where $\bar{U}$, $\bar{V}$, $\bar{W}^V$, and $(\bar{W}^U)^H$ are defined in the algorithm and $\bar{\Delta}$ is equal to $\tilde{\Delta}$ with the rows and columns in $\mathscr{S}$ eliminated. Since $\tilde{\Delta} \mapsto \bar{\Delta}$ is one-to-one and $\bar{\sigma}(\tilde{\Delta}) = \bar{\sigma}(\bar{\Delta})$,

$$\mu_{\Delta}(M) = (\min \{\bar{\sigma}(\Delta) | \det(I - \bar{U} \Sigma \bar{V}^H \bar{W}^V \times (\bar{W}^U)^H \bar{\Delta}) = 0\})^{-1} = \mu_{\bar{\Delta}}(\bar{m}). \tag{33}$$

Now, let us consider $\mu_{\Delta}(M)$ when there does not exist a $\Delta \in \mathbf{\Delta}$ such that $\det(I - M\Delta) = 0$. Then by reviewing the previous steps of this proof, it is concluded that there does not exist and $\bar{\Delta} \in \bar{\Delta}$ such that $\det(\mathbf{I} - \bar{M}\bar{\Delta}) = 0$. Therefore, for the case when $\mu_{\Delta}(M) = 0$, $\mu_{\bar{\Delta}}(\bar{M}) = 0$. □

A.2. Proof of Theorem 2

Given the definitions provided in the theorem statement,

$$\min_{D \in \mathscr{D}} \bar{\sigma}(DMD^{-1}) \le \bar{\sigma}(D^* M (D^*)^{-1}) \tag{34}$$

$$= \bar{\sigma}(D^*(M + E - E)(D^*)^{-1}) \tag{35}$$

$$\le \bar{\sigma}(D^*(M + E)(D^*)^{-1}) + \bar{\sigma}(D^* E (D^*)^{-1}) \tag{36}$$

$$\le \bar{\sigma}(D^*(M + E)(D^*)^{-1}) + \bar{\sigma}(D^*)\bar{\sigma}(E)\bar{\sigma}((D^*)^{-1}) \tag{37}$$

$$= \bar{\sigma}(D^*(M + E)(D^*)^{-1}) + \kappa(D^*)\bar{\sigma}(E) \tag{38}$$

which are the right-hand-side inequalities in equations (6) and (7). A similar argument gives the left-hand-side inequalities in equations (6) and (7). □

A.3. Proof of Corollary 1

Given the definitions provided in the corollary statement,

$$\begin{aligned} M &= U_+ \Sigma_+ V_+^H + U_- \Sigma_- V_-^H \\ &= (W^U)^H W^U U_+ \Sigma_+ V_+^H (W^V)^H W^V + U_- \Sigma_- V_-^H \\ &= (W^U)^H (\hat{U}_+ + \check{U}_+) \Sigma_+ (\hat{V}_+ + \check{V}_+)^H W^V + U_- \Sigma_- V_-^H \\ &= \underbrace{(W^U)^H \hat{U}_+ \Sigma_+ \hat{V}_+^H W^V}_{M+E} + \underbrace{(W^U)^H (\hat{U}_+ \Sigma_+ \check{V}_+^H + \check{U}_+ \Sigma_+ \hat{V}_+^H + \check{U}_+ \Sigma_+ \check{V}_+^H) W^V + U_- \Sigma_- V_-^H}_{-E}, \end{aligned} \tag{39}$$

where W^U and W^V are defined in Table 1. This derived the expression in equation (10).

The following equalities shows that $\bar{\mu}_{\Delta}(M+E) = \bar{\mu}_{\bar{\Delta}}(\bar{M})$ when all the uncertainties are full block:

$$\begin{aligned}
&\bar{\sigma}(D(M+E)D^{-1}) \\
&\quad = \bar{\sigma}(W^U D(W^U)^H \hat{U}_+ \Sigma_+ \hat{V}_+^H W^V D^{-1}(W^V)^H) \\
&\quad = \bar{\sigma}(D\hat{U}_+ \Sigma_+ \hat{V}_+^H D^{-1}) \\
&\quad = \bar{\sigma}(D\hat{U}_+ \Sigma_+ \hat{V}_+^H \tilde{W}^V (\tilde{W}^U)^H D^{-1}) \\
&\quad = \bar{\sigma}(\bar{D}\bar{U}\Sigma\bar{V}^H \tilde{W}^V (\tilde{W}^U)^H \bar{D}^{-1}) \\
&\quad = \bar{\sigma}(\bar{D}\bar{M}\bar{D}^{-1}).
\end{aligned} \tag{40}$$

The first equality follows from equation (39) and the fact that W^U and $(W^V)^H$ are unitary matrices. The second equality results since D and D^{-1} commute with W^U and $(W^V)^H$, respectively. The third and fourth equalities follow from the fact $\hat{U}_+ \Sigma_+ \hat{V}_+^H$ has the form of equation (26) where $\tilde{W}^V$ and $\tilde{W}^U$ are defined in Theorem 1. The last equality follows by definition. Applying Theorem 2, $\kappa(\bar{D}^*) = \kappa(D^*)$, and equation (40) proves equations (8) and (9).

To prove equation (11), apply equation (9) and the inequalities:

$$\begin{aligned}
\bar{\sigma}(E) &\le \bar{\sigma}((W^U)^H(\hat{U}_+ \Sigma_+ \check{V}_+^H + \check{U}_- \Sigma_+ \hat{V}_+^H \\
&\qquad + \check{U}_+ \Sigma_+ \check{V}_+^H) W^V) + \bar{\sigma}(U_- \Sigma_- V_-^H) \\
&\le \bar{\sigma}(\hat{U}_+ \Sigma_+ \check{V}_+^H + \check{U}_+ \Sigma_+ \hat{V}_+^H + \check{U}_+ \Sigma_+ \check{V}_+^H) \\
&\qquad + \bar{\sigma}(U_- \Sigma_- V_-^H) \\
&\le \bar{\sigma}(\hat{U}_+ \Sigma_+ \check{V}_+^H) + \bar{\sigma}(\check{U}_+ \Sigma_+ \hat{V}_+^H) \\
&\qquad + \bar{\sigma}(\check{U}_+ \Sigma_+ \check{V}_+^H) + \bar{\sigma}(U_- \Sigma_- V_-^H) \\
&\le \bar{\sigma}(\Sigma_+) + \bar{\sigma}(\Sigma_+) + \bar{\sigma}(\Sigma_+) + \bar{\sigma}(\Sigma_-) \\
&= 3\bar{\sigma}(M) + \bar{\sigma}(\Sigma_-).
\end{aligned} \tag{41}$$

□

A.4. Proof of Theorem 3

The number of flops (floating point operations) needed to compute the SVD for the first step of the algorithm is bounded by $13n^3$ where $M \in \mathbb{C}^{n \times n}$ (Golub and van Loan, 1983). In the second step, the number of flops required to row reduce all the inner and outer rotation matrices, U_i and V_i, corresponding to the block structures r_i is bounded by $(26l + 4)n^3 - 2ln^2$, where r_i and l are defined in equation (2). This bound can be calculated by considering the SVD and matrix multiplication individually. First, the flops needed to calculate the SVDs of every U_i and V_i is limited to $26ln^3$. Secondly, the bound on the flops needed to perform the matrix multiplications for the second step is

$$\begin{aligned}
2\sum_{i=1}^{l} nr_i(2r_i - 1) &\le 2n^2\left(\sum_{i=1}^{l} 2r_i - 1\right) \\
&= 2n^2(2n - l) = 4n^3 - 2ln^2.
\end{aligned} \tag{42}$$

By summing the two parts of the second step, the bound on the flops is calculated to be $(26l + 4)n^3 - 2ln^2$.

The third and fourth steps of the algorithm do not require any flops. The fifth step consists entirely of matrix multiplication, and the necessary number of flops is $n(2n-1)\bar{n} + 2(2n-1)\bar{n}^2 + n(2\bar{n}-1)\bar{n}$. Since $n \ge \bar{n}$, the number of flops for the fifth step is bounded by the polynomial $8n^3 - 4n^2$. Therefore, the flops performed for the entire model reduction algorithm is bounded by $(26l + 12)n^3 - (2l + 4)n^2$. □

A.5. Proof of Theorem 4

Since $\text{rank}(U_i)$ and $\text{rank}(V_i)$ from Step 2 of Table 1 must be less than or equal to $\text{rank}(M)$, by the definition of $\bar{r}_i$ in Step 3 of Table 1, each element of the resulting block structure vector of the EMR algorithm satisfies

$$\bar{r}_i \le rank(M). \tag{43}$$

Theorem 4 follows by summing both sides of this inequality over the number of perturbation blocks.

A.6. Matrices for Example 2

$$U = \frac{1}{100}\begin{bmatrix}
8 & 5 & 6 & 7 & 7 & 9 & 10 & 8 & 6 & 8 & 8 & 8 & 4 & 9 & 8 & 6 & 1 & 4 & 10 & 6 & 0 & 7 & 8 & 5 & 8 \\
10 & 7 & 10 & 3 & 5 & 2 & 1 & 0 & 1 & 3 & 2 & 5 & 4 & 7 & 4 & 0 & 5 & 5 & 9 & 7 & 2 & 7 & 10 & 9 & 5 \\
9 & 3 & 6 & 8 & 2 & 9 & 3 & 10 & 8 & 6 & 9 & 10 & 6 & 5 & 7 & 6 & 4 & 4 & 10 & 1 & 4 & 3 & 9 & 7 & 4 \\
3 & 0 & 5 & 9 & 9 & 1 & 6 & 4 & 5 & 5 & 3 & 2 & 0 & 9 & 1 & 2 & 9 & 3 & 8 & 9 & 7 & 7 & 4 & 10 & 9 \\
10 & 5 & 8 & 1 & 3 & 4 & 6 & 3 & 5 & 9 & 5 & 1 & 8 & 10 & 5 & 3 & 7 & 6 & 1 & 8 & 10 & 6 & 3 & 10 & 10 \\
9 & 2 & 2 & 0 & 8 & 8 & 10 & 8 & 5 & 2 & 1 & 9 & 6 & 9 & 9 & 9 & 9 & 0 & 2 & 6 & 0 & 5 & 4 & 5 & 8 \\
5 & 6 & 3 & 6 & 6 & 0 & 4 & 4 & 7 & 5 & 1 & 5 & 2 & 7 & 6 & 0 & 8 & 7 & 8 & 7 & 2 & 2 & 9 & 10 & 1 \\
4 & 10 & 0 & 9 & 8 & 9 & 7 & 1 & 10 & 4 & 4 & 7 & 9 & 1 & 2 & 2 & 4 & 8 & 6 & 3 & 3 & 9 & 3 & 3 & 6
\end{bmatrix}^T$$

$$+ \frac{j}{100}\begin{bmatrix}
9 & 6 & 4 & 3 & 4 & 2 & 3 & 6 & 8 & 3 & 7 & 7 & 6 & 9 & 8 & 0 & 1 & 2 & 0 & 1 & 5 & 8 & 0 & 2 & \\
4 & 8 & 5 & 9 & 7 & 2 & 6 & 1 & 5 & 3 & 6 & 5 & 8 & 4 & 10 & 3 & 5 & 6 & 4 & 1 & 10 & 5 & 5 & 6 & \\
8 & 4 & 5 & 8 & 10 & 8 & 1 & 0 & 7 & 10 & 6 & 10 & 1 & 5 & 8 & 4 & 1 & 8 & 3 & 6 & 7 & 4 & 1 & 2 & 8 & 1 \\
8 & 9 & 5 & 0 & 6 & 2 & 2 & 3 & 5 & 1 & 6 & 1 & 4 & 3 & 1 & 2 & 3 & 4 & 2 & 2 & 9 & 5 & 5 & 3 & 1 & 5 \\
0 & 4 & 8 & 5 & 0 & 5 & 8 & 6 & 1 & 3 & 5 & 9 & 8 & 9 & 10 & 7 & 5 & 2 & 6 & 0 & 6 & 9 & 10 & 2 & 4 & \\
2 & 4 & 7 & 10 & 10 & 2 & 7 & 5 & 9 & 6 & 6 & 2 & 5 & 10 & 3 & 5 & 2 & 9 & 7 & 3 & 0 & 0 & 8 & 5 & 3 & \\
6 & 6 & 0 & 4 & 3 & 1 & 4 & 5 & 2 & 7 & 9 & 4 & 1 & 1 & 7 & 8 & 1 & 4 & 10 & 6 & 7 & 6 & 6 & 5 & 2 & \\
10 & 4 & 8 & 6 & 5 & 10 & 3 & 1 & 4 & 0 & 10 & 10 & 1 & 4 & 9 & 10 & 8 & 6 & 9 & 5 & 9 & 6 & 2 & 3 & 4 &
\end{bmatrix}^T \tag{44}$$

$$V = \frac{1}{100}\begin{bmatrix}
9 & 2 & 7 & 8 & 1 & 5 & 7 & 0 & 9 & 5 & 0 & 7 & 5 & 9 & 6 & 6 & 3 & 10 & 7 & 6 & 9 & 6 & 0 & 10 & 4 \\
10 & 5 & 7 & 8 & 2 & 3 & 4 & 4 & 5 & 0 & 5 & 8 & 10 & 10 & 3 & 8 & 1 & 7 & 2 & 5 & 9 & 6 & 8 & 8 & 0 \\
2 & 3 & 9 & 5 & 6 & 1 & 2 & 0 & 8 & 2 & 9 & 7 & 4 & 2 & 10 & 7 & 6 & 4 & 8 & 4 & 8 & 4 & 1 & 2 & 2 \\
4 & 4 & 6 & 9 & 7 & 1 & 3 & 0 & 0 & 4 & 1 & 4 & 6 & 7 & 2 & 1 & 5 & 4 & 3 & 1 & 4 & 6 & 2 & 7 & 9 \\
8 & 1 & 7 & 7 & 4 & 9 & 2 & 1 & 0 & 6 & 1 & 7 & 9 & 4 & 5 & 10 & 7 & 2 & 2 & 4 & 10 & 6 & 0 & 4 & 8 \\
0 & 1 & 5 & 2 & 4 & 8 & 4 & 4 & 2 & 1 & 8 & 5 & 3 & 5 & 7 & 6 & 8 & 4 & 7 & 9 & 9 & 1 & 7 & 6 & 1 \\
3 & 6 & 1 & 9 & 8 & 8 & 4 & 2 & 5 & 2 & 9 & 4 & 10 & 3 & 1 & 5 & 1 & 9 & 9 & 7 & 1 & 6 & 9 & 2 & 8 \\
1 & 7 & 4 & 8 & 8 & 5 & 2 & 1 & 9 & 8 & 2 & 7 & 2 & 6 & 10 & 1 & 4 & 6 & 8 & 10 & 8 & 7 & 4 & 8 & 7
\end{bmatrix}$$

$$+ \frac{j}{100}\begin{bmatrix}
7 & 7 & 8 & 8 & 10 & 6 & 3 & 10 & 6 & 4 & 5 & 3 & 3 & 8 & 8 & 1 & 5 & 2 & 7 & 8 & 3 & 1 & 9 & 2 & & & \\
10 & 4 & 5 & 5 & 7 & 3 & 3 & 4 & 8 & 8 & 8 & 4 & 3 & 7 & 5 & 1 & 1 & 2 & 3 & 7 & 8 & 6 & 6 & 8 & & & \\
2 & 3 & 6 & 3 & 3 & 8 & 1 & 9 & 3 & 5 & 1 & 9 & 5 & 8 & 5 & 4 & 3 & 8 & 7 & 8 & 7 & 1 & 5 & 10 & 5 & & \\
0 & 0 & 1 & 8 & 0 & 3 & 0 & 8 & 8 & 3 & 0 & 1 & 9 & 6 & 6 & 1 & 4 & 5 & 8 & 1 & 10 & 4 & 6 & 5 & 5 & & \\
1 & 4 & 9 & 7 & 3 & 7 & 4 & 7 & 8 & 2 & 6 & 5 & 0 & 4 & 3 & 2 & 0 & 7 & 4 & 7 & 1 & 1 & 5 & 3 & 7 & 3 & 5 \\
5 & 1 & 7 & 9 & 3 & 4 & 1 & 4 & 6 & 7 & 5 & 8 & 0 & 5 & 7 & 0 & 5 & 9 & 7 & 10 & 6 & 6 & 7 & 5 & 10 & 1 & 7 \\
1 & 3 & 5 & 5 & 9 & 0 & 6 & 3 & 6 & 3 & 4 & 5 & 5 & 2 & 3 & 7 & 9 & 3 & 8 & 5 & 7 & 1 & 1 & 7 & 5 & & \\
8 & 3 & 5 & 4 & 7 & 6 & 1 & 10 & 4 & 7 & 8 & 1 & 6 & 7 & 2 & 5 & 4 & 7 & 4 & 1 & 9 & 4 & 2 & 3 & 9 & &
\end{bmatrix} \tag{45}$$