JB 82-02

Paper Entitled

"Algorithms and Computational Aspects
Pertaining to Block Diagonal Methods for
Design of Decentralized Feedback Compensation"

From the Proceedings of

21st IEEE Conference on Decision and Control
pp. 479-484

Orlando, Florida
December 1982

ALGORITHMS AND COMPUTATIONAL ASPECTS PERTAINING TO BLOCK DIAGONAL DOMINANCE METHODS FOR DESIGN OF DECENTRALIZED FEEDBACK COMPENSATION

W. H. Bennett

Aerospace Systems Division
Naval Research Laboratory
Washington, D.C. 20375

and

J. S. Baras

Electrical Engineering Department
University of Maryland
College Park, MD. 20742

ABSTRACT

In a previous paper the notion of a block diagonally dominant transfer function matrix is used to provide criteria for the design of decentralized feedback control. In this paper we develop computer implementable algorithms for determining the block diagonal dominance condition. The algorithms are developed with consideration for numerical stability. Implementation of the algorithms in an interactive computing environment is suggested.

1.0 Introduction

In [3] the present authors described a new design methodology for decentralized feedback control based on a frequency dependent notion of low-interacting subsystems. Using a generalized Gershgorin theorem for partitioned matrices we provided a definition of block diagonal dominance of a matrix rational transfer function on a contour in the complex s plane. This criterion was used to establish conditions under which a plant can be stabilized by constant gain decentralized linear feedback. Moreover the dominance margins (analogous to the radii of the Gershgorin discs in the standard case) are shown to be useful in determining robustness of the closed loop decentralized structure to modeling uncertainty.

In this paper we concentrate on the computational problem of determining whether a matrix transfer function is block diagonally dominant (BDD) along a significant portion of the imaginary axis. The flexibility inherent in the method in terms of choice of matrix norms and partitioning leaves us with several computational problems which do not arise in the standard case [1]. The structure of the paper is as follows. In section 2.0 we give some notation, definitions, and describe the fundamental computational problem. In section 3.0 we discuss the development of computer implementable algorithms for determining BDD over a closed interval of the imaginary axis.

2.0 The Computational Problem of Determining BDD

2.1 Notation/Definitions

Throughout this paper we will use the following notational conventions. An nxn matrix, A, will often be considered in an mxm block partitioning. The A_{ij} will denote the ijth submatrix of dimension $k_i x k_j$ where $\Sigma_{i=1}^{m} k_i = n$.

Given a vector space norm $|\cdot|$, in addition to the usual induced matrix norm,

$$||A|| = \sup_{x \neq 0} \frac{|Ax|}{|x|}, \qquad (1)$$

we can define a matrix infimum or reciprocal norm as

$$\square A \square = \inf_{x \neq 0} \frac{|Ax|}{|x|}. \qquad (2)$$

Clearly for A nonsingular $\square A \square = ||A^{-1}||^{-1}$. We will denote the set of singular values of a matrix A as sing(A). Let

$$\sigma_{min}(A) \overset{\Delta}{=} \min_{\sigma \varepsilon sing(A)} \sigma .$$

$$\sigma_{max}(A) \overset{\Delta}{=} \max_{\sigma \varepsilon sing(A)} \sigma .$$

Thus using the Euclidean vector norm $\sigma_{min}(A) = \square A \square_2$ and $\sigma_{max}(A) = ||A||_2$.

Definition: An nxn matrix rational function A(s) partitioned into mxm subblocks is said to be block diagonally dominant (BDD) on a contour Γ in the complex plane if

$$d_{A,\Gamma} = \min_{s \varepsilon \Gamma} \{ \max[\min_{i \varepsilon \underline{m}} d_i(s), \min_{i \varepsilon \underline{m}} d_i'(s)] \} > 0,$$

for all s on Γ where

$$d_i(s) = \square A_{ii}(s) \square - \sum_{k \neq i}^{m} ||A_{ik}(s)||$$

$$d_i'(s) = \square A_{ii}(s) \square - \sum_{k \neq i}^{m} ||A_{ki}(s)||$$

and $\underline{m} = \{1,2,\ldots,m\}$.

2.2 The Computational Problem

As discussed in [3] we are concerned with the design of decentralized feedback compensation which can be expressed formally via the equations,

$$\begin{aligned} y(s) &= G(s)\ u(s) \\ u(s) &= Fy(s) + v(s) \end{aligned} \qquad (3)$$

where G(s) is the nxn plant rational transfer function matrix and F is an nxn block diagonal (decentralized) constant feedback compensator. We are concerned with conditions which guarantee the regularity of the matrix return difference, $R(s) \overset{\Delta}{=} I + FG(s)$, for s along a significant portion of the jω axis. The relationship between BDD and regularity of R(s) suggests that the quantity $\square R(s) \square$, thought of as a measure of regularity for R(s) at s, will play an important role.

The relationship between matrix regularity conditions on $R(s)$ and inclusion regions for its spectrum (as provided in this case by the generalized Gershgorin theorem) is intrinsic since $R(s) - \lambda I$ is singular if and only if λ is an eigenvalue for $R(s)$. Considerable freedom is available in defining $\Box\bullet\Box$ (since we can choose any vector norm), but for computational reasons there is an obvious natural choice. Similarly, the quantity $||H_m(s)||$ provides a measure of closed loop robustness in the face of modeling uncertainties, where $H_m(s) = G_m[I + F(s)G(s)]^{-1}$. This follows simply by considering uncertainty as an additive perturbation $\Delta G(j\omega)$ to the true plant response $G_t(j\omega)$; i.e., $G_m(j\omega) = G_t(j\omega) + \Delta G(j\omega)$, with known bound $\ell(\omega) > ||\Delta G(j\omega)||$. If $||H_m(j\omega)||<\ell(\omega)$ then the closed loop structure remains stable for all perturbations satisfying this bound [10].

The computational problem considered here is to determine for some desired partitioning of the plant transfer function matrix, $G(s)$, if $G(s)$ is BDD along a closed interval, $\Omega = [-j\omega_o, j\omega_o]$, of the imaginary axis. This criterion can be expressed as

$$d_i^* = \min_{\omega\in\Omega} \{\max[d_i(\omega),\ d_i'(\omega)]\} > 0 \qquad (4)$$

for each $i \in \underline{m}$, where

$$\begin{aligned} d_i(\omega) &= \Box G_{ii}(j\omega)\Box - \sum_{k\neq i}^{m} ||G_{ik}(j\omega)||, \\ d_i'(\omega) &= \Box G_{ii}(j\omega)\Box - \sum_{k\neq i}^{m} ||G_{ki}(j\omega)||, \end{aligned} \qquad (5)$$

and $\underline{m} = \{1,2,\ldots,m\}$. Futhermore, to establish certain design objectives of closed loop robustness we may need to compute estimates for the dominance margins, d_i^* for $i\varepsilon\underline{m}$.

3.0 Development of Computer Implementable Algorithms

3.1 Computational Considerations for the Choice of Norms

The matrix norms $||A||_\infty$, the matrix absolute row-sum, and $||A||_1$, the matrix absolute column-sum, offer obvious computational advantages over $||A||_2$, the maximum singular value of A. However, computation of the associated infima, $\Box A\Box_1$ and $\Box A\Box_\infty$, is more or less equivalent to computing the matrix A^{-1} first, and then computing the norm of A^{-1}. By construction of the closed contour Γ, we know that $G_{ii}^{-1}(s)$ will exist on Γ [1]. Inversion of $A_{ii}(s)$ at a finite number of frequencies ω, $0\leq\omega\leq\omega_0$, can be implemented using standard algorithms for the solution of systems of linear equations; e.g., an LU or QR factorization method may be used, both of which are available in LINPACK [6]. The computational cost for this approach will be dominated by approximately k_i^3 multiplications required to compute these factorizations for each ω. (Typically a QR method requires twice as many multiplications as the LU method but is potentially more stable numerically.) However there is no guarantee that the matrix condition number with respect to inversion, $\kappa(A_{ii}(s)) = ||A_{ii}(s)||/\Box A_{ii}(s)\Box$, will not be large for some s on Γ. If this is true it indicates that the calculation of $A_{ii}^{-1}(s)$ and therefore $\Box A_{ii}(s)\Box = ||A_{ii}^{-1}(s)||^{-1}$ may be largely dominated by roundoff errors. From (4) and (5) we see that this may occur when the accuracy of the calculation is most crucial for BDD considerations. This suggests that independently of the norm used, estimates of the condition number, $\kappa(A_{ii}(s))$, should be also computed. Alternately, $A_{ii}^{-1}(s)$ can be computed over the field of rational functions via a modified Faddeev algorithm [5]. However, this algorithm may be numerically unstable in practice [9].

Let sing(A) denote the set of singular values of A. Then the maximum and minimum singular values of A(s) are:

$$||A||_2 = \sigma_{\max}\{A\} \equiv \max_{\sigma\ \varepsilon\ \mathrm{sing}(A)} \sigma$$

$$\Box A\Box_2 = \sigma_{\min}\{A\} \equiv \min_{\sigma\ \varepsilon\ \mathrm{sing}(A)} \sigma\ .$$

We can therefore exploit standard, numerically-stable, algorithms for determining the singular values of A(s) for a finite number of values of s on Γ. It is our opinion that the use of the Euclidean vector norm on all subspaces is most appropriate for general application of our results, primarily because of the availability of standard software which has been well tested [6]. In section 3.3 we discuss the application of this standard software to the problem of finding $\Box A(j\omega)\Box_2$ and propose an algorithm for doing this.

The price we pay for using the 2-norm is the computational cost. If standard software is used to find the full singular value decomposition (SVD) of the submatrix $A_{ij}(s)$ then typically four times as many multiplications are required as for either an LU or QR factorization method at every value s on Γ. We wish to point out, in defference to the former conclusion, that in special cases there may be other (perhaps more gross) estimates of the dominance margins involving other matrix norms which will suffice for our applications. For instance, by recognizing the equivalence of norms on finite dimensional spaces we can state that a sufficient condition for A, an mxm matrix partitioned as above, to be BDD using the ∞-norm on all subspaces and having margin of dominance d_i for $i\varepsilon\underline{m}$ in that norm;

$$\text{i.e.,}\quad \Box A_{ii}\Box_\infty - \sum_{j\neq i}^{m} ||A_{ij}||_\infty \geq d_i \quad \text{for } i\varepsilon\underline{m}$$

is that

$$k_i^{-1/2}\Box A_{ii}\Box_2 > \sum_{j\neq i}^{m} ||A_{ij}||_\infty + d_i \quad \text{for } i\varepsilon\underline{m}.$$

This follows from the well known fact that for $B \varepsilon C^{mxn}$ $\sqrt{n}||B||_2 \geq ||B||_\infty$. Or equivalently that $\sqrt{n}\Box B\Box_\infty \geq \Box B\Box_2$ (cf.[14]).

Thus we can calculate an estimate (lower bound) for the row (or column) dominance margin, $d_i(s)$ (or $d_i'(s)$), measured in the ∞-norm as

$$\hat{d}_i = k_i^{-1/2}\ \Box A_{ii}\Box_2 - \sum_{j\neq i}^{m} ||A_{ij}||_\infty$$

which is tight for k_i a small integer. In such cases we can take advantage of the computational efficiency of the matrix ∞-norm for the off-diagonal blocks. However in many cases (e.g., k_i>>1) this may be a conservative estimate of the actual dominance margin. (In fact you may not be able to determine if a matrix is BDD from such estimates at all!) We wish to emphasize that such computationally efficient estimates for the margin of dominance offer significant advantages in the determination of partitions "natural" to the problem at hand.

3.2 Determination of Block Diagonal Dominance: A Line Search Problem

Clearly the computational problem of solving equation (4) over the closed interval Ω of the imaginary axis may involve a very costly line search unless conditions are available which can characterize a finite point subset of Ω within which the minimum must lie. Usually this would be done by finding the roots of the derivative (as long as it exists) of the function being minimized. Except in simple cases where these roots can be found algebraically, we would usually employ a recursive root finding method (e.g., Newton's method) whose convergence would depend on either:

(i) an assumption of differentiability of the function everywhere on Ω,

or (ii) an assumption that the function is unimodal on Ω.

In the case of (4) neither of these assumptions holds on Ω. Given a rational matrix function, $A(\omega)$, $\omega\varepsilon R$ and any vector norm the associated matrix norm $||A(\omega)||$ and matrix infimum, $\square A(\omega)\square$, are each continuous functions of ω which are not continuously differentiable on Ω. Thus clearly $d_i(\omega)$, $d_i'(\omega)$, and $\psi(\omega)\overset{\Delta}{=}\max[d_i(\omega),d_i'(\omega)]$ are each continuous on $\omega\varepsilon\Omega$. In such cases the best one can hope for is to solve (4) approximately by recursively refining a finite grid of points, $\Omega_k \subseteq \Omega$; i.e., $\Omega_k \subseteq \Omega_{k+1} \subseteq \cdots \subseteq \Omega$. The resulting sequence, $\{d_i^k\}_{k=1}^{\infty}$, converges to d_i^* in the sense that $d_i \leq d_i^{k+1} \leq d_i^k$.

From a practical standpoint we would like to make use of some additional problem structure so as to more intelligently direct recursive refinement of the grids, Ω_k. For this reason and because of the discussion in 3.1 we will consider only the Euclidean vector norm in defining $d_i(\omega)$, and $d_i'(\omega)$. Thus we rewrite (4) and (5) as

$$d_i^* = \min_{\omega\varepsilon\Omega}\{\psi(\omega)\} > 0$$
$$\psi_i(\omega) = \max\lfloor d_i(\omega),\ d_i'(\omega)\rfloor \tag{4'}$$

for $i\varepsilon\underline{m}$ and with

$$d_i(\omega) = \sigma_{\min}\{G_{ii}(j\omega)\} - \sum_{k\neq i}^{m} \sigma_{\max}\{G_{ik}(j\omega)\}$$
$$d_i'(\omega) = \sigma_{\min}\{G_{ii}(j\omega)\} - \sum_{k\neq i}^{m} \sigma_{\max}\{G_{ki}(j\omega)\}. \tag{5'}$$

This puts us in contact with the spectral properties of the singular values of matrices. Let $A(\omega)$ be an ℓxp complex valued rational matrix function of the real scalar ω. ($A(\omega)$ will be used to represent generically the submatrices $G_{ik}(j\omega)$.) Without loss of generality, take $\ell<p$. We prefer to define the ℓ nonzero singular values for $A(\omega)$ as $\sigma_i(\omega)$ $i=1,\ldots,\ell$ which satisfy the symultaneous equations

$$A(\omega)\ y^i(\omega) = \sigma_i(\omega)\ x^i(\omega)$$
$$A^*(\omega)\ x^i(\omega) = \sigma_i(\omega)\ y^i(\omega) \tag{6}$$

for the unit length vectors $x^i(\omega)\ \varepsilon\ C^{\ell}$ and $y^i(\omega)\ \varepsilon\ C^{p}$ called the left (resp. right) singular vectors associated with the i^{th} singular value, $\sigma_i(\omega)$.

Proposition 1:

The derivative of some $\sigma_i(\omega)\ \varepsilon$ sing $\{A(\omega)\}$,

$$\frac{d\sigma_i(\omega)}{d\omega}\ \overset{\Delta}{=}\ \lim_{h\to 0}\ \frac{\lfloor\sigma_i(\omega+h)-\sigma_i(\omega)\rfloor}{h},$$

exists and is continuous for all $\omega\varepsilon\Omega$ where $\sigma_i(\omega)$ is distinct; i.e., $\sigma_i(\omega) \neq \sigma_j(\omega)$ for $j\ \varepsilon\ \underline{\ell}$, $j\neq i$.

Proof: (cf. [16]).

Indeed when $\sigma_i(\omega)$ is distinct we can calculate the derivative as

$$\frac{d\sigma_i(\omega)}{d\omega} = y^{i*}(\omega)\ \frac{dA(\omega)}{d\omega}\ x^i(\omega) + x^{i*}(\omega)\left[\frac{dA(\omega)}{d\omega}\right]^*\ y^i(\omega) \tag{7}$$

where $x^i(\omega)$ and $y^i(\omega)$ are the right (resp. left) singular vectors for $A(\omega)$ associated with $\sigma_i(\omega)$. This formula is easy to see by recognizing that (6) can be written

$$\left\{\begin{pmatrix}0 & A(\omega)\\ A^*(\omega) & 0\end{pmatrix} - \sigma_i(\omega)I\right\}\left\{\begin{matrix}x^i(\omega)\\ y^i(\omega)\end{matrix}\right\} = 0. \tag{8}$$

Then differentiation of (8) leads to (7).

Next we observe that the following assumptions will hold in most design situations.

Assumption 1: The rational matrix transfer function $G(s)$ will be analytic for $s = j\omega\ \omega\varepsilon\Omega$.

Assumption 2: Let $\psi_i(\omega) = \max[d_i(\omega),d_i'(\omega)]$ for $i\varepsilon\underline{m}$. Then the set $\hat{\Omega} \subsetneq \Omega$ of points in Ω at which $d\psi_i/d\omega$ does not exist is a finite countable subset of Ω with the property that the points $\hat{\omega}\ \varepsilon\ \underset{\underline{m}}{\cup}\ \hat{\Omega}_i$ are isolated in Ω; i.e., for any two points $\hat{\omega}_1,\hat{\omega}_2\ \varepsilon\ \underset{\underline{m}}{\cup}\ \hat{\Omega}_i$ there exists an open neighborhood $B\varepsilon(\hat{\omega}_1)$ of ω_1 which does not contain $\hat{\omega}_2$.

Remark: The inclusion of a point $\hat{\omega}\ \varepsilon\ \Omega$ in the set $\underset{\underline{m}}{\cup}\ \Omega_i$ is associated with one of three possible events:

(i) $\sigma_{\min}\{G_{ii}(\hat{\omega})\}$ is not distinct for some $i\varepsilon\underline{m}$

or (ii) $\sigma_{\max}\{G_{ik}(\hat{\omega})\}$ is not distinct for some $i,k\ \varepsilon\ \underline{m}$, $i\neq k$

(iii) $d_i(\hat{\omega}) = d_i'(\hat{\omega})$ for some $i\varepsilon\underline{m}$.

For the above class of functions $\psi_i(\omega)$ over Ω we can define uniquely upper and lower limiting values for $\frac{d\psi_i(\hat{\omega})}{d\omega}$ at $\hat{\omega}_i\varepsilon\hat{\Omega}$ in the following sense. Take $\hat{\omega}_i\ \varepsilon\ \hat{\Omega}$ and choose two sequences $\{\underline{\omega}_k\}$ and $\{\bar{\omega}_k\}$ converging to $\hat{\omega}$ from below (resp. above); i.e., $\underline{\omega}_1<\underline{\omega}_2<\ldots<\underline{\omega}_n<\hat{\omega}$ and $\hat{\omega}<\bar{\omega}_n<\bar{\omega}_{n-1}<\ldots<\bar{\omega}_1$). Then

$$\lim_{k\to\infty}\ \frac{d\psi_i(\underline{\omega}_k)}{d\omega} = \underline{\psi}'(\hat{\omega})\quad,\quad \lim_{k\to\infty}\frac{d\psi_i(\bar{\omega}_k)}{d\omega} = \bar{\psi}'(\hat{\omega}).$$

Clearly if $\underline{\psi}_i'(\hat{\omega})\ \bar{\psi}_i'(\hat{\omega}) < 0$ then $\psi_i(\hat{\omega})$ is a local extremum for $\psi_i(\hat{\omega})$.

With the above framework for the class of functions $\psi_i(\omega)$, we will propose a algorithm for generating a sequence of refined grids $\Omega_0 \subseteq \Omega_1 \subseteq \Omega_2 \subseteq \cdots \subseteq \Omega$ based on the idea of bisection in root finding. To do this we need the following assumption.

Assumption 3: We can choose an initial grid $\Omega_0 \subseteq \Omega$ with the property that for any two adjacent points ω_1, ω_2, $\varepsilon\ \Omega_0$ there exists no more than one point $\omega\ \varepsilon\ [\omega_1, \omega_2]$ at which $\psi(\omega)$ has a local extremum.

Thus we offer the following algorithm.

Algorithm for Recursive Grid Refinement :

Purpose: To compute $\psi_i^*(\omega) = \min\limits_{\omega\varepsilon\Omega} \psi_i(\omega)$ (cf. eqns. (4') and (5'))

Given: $\Omega_0 \subseteq \Omega$, a finite point set ("grid") on Ω
Ω_0 is ordered s.t. $\omega_{n+1} > \omega_n$
for $n = 1,\ldots, N_0-1$
N_0 = cardinality of Ω_0
and ε is a miminum desired grid point resolution

Step 1: Compute $\psi_i(\omega)$ for $\omega\ \varepsilon\ \Omega_k$ (cf., section 3.3 for details)

Step 2: Compute for $\omega\ \varepsilon\ \Omega_k$

$$\frac{d\psi_i(\omega)}{d\omega} = \begin{cases} \dfrac{d\,[d_i(\omega)]}{d\omega}, \text{if } d_i(\omega) > d_i'(\omega) \\ \\ \dfrac{d\,[d_i'(\omega)]}{d\omega}, \text{if } d_i'(\omega) > d_i(\omega) \end{cases}$$

(cf., eqn.(7))

Step 3: Set $\psi_i^* = \min\limits_{\omega\varepsilon\Omega_k} \{\psi_i(\omega)\}$

$\omega^* = \arg\ \psi_i^*$

Step 4:

$$\Omega_{k+1} = \Omega_k \cup \left\{\omega = \omega_n + \frac{\omega_{n+1} - \omega_n}{2} \right| \quad \omega_n, \omega_{n+1}\ \varepsilon\ \Omega_k$$

for $n = 1,2,\ldots,N_k-1$,

$$\frac{d\psi}{d\omega}i(\omega_{n+1}) < 0 < \frac{d\psi}{d\omega}i(\omega_n), \quad |\omega_{n+1} - \omega_n| > \varepsilon\}$$

Set N_{k+1} = cardinality of Ω_{k+1}

Step 5: If $\Omega_{k+1} = \Omega_k$ return ψ_i^* and ω^* and stop.

Else return to step 1.

The computations required in step 1 and step 2 above involve finding the minimum (or maximum) singular value and associated right and left singular vectors for certain matrix subblocks of G(s) (see equation (7)). Although this information is available from a full singular value decomposition we propose an algorithm in section 3.3 which concentrates on finding the minimum amount of information necessary in an effort to decrease computational cost.

We wish to point out that construction of an initial grid, Ω_0, which meets assumption 3 may be difficult for large multivariable problems even when a transfer function model is known. However via an appropriate partitioning of G(s) for decentralized control we hope that this construction may be more physically motivated by looking at the subsystem and interconnection dynamics separately. Clearly the algorithm will be most useful in an interactive computing environment where the user can apply engineering insight.

3.3 An Algorithm for Computing $\|A(j\omega)\|_2$ Along ω in Ω

Several standard procedures are available for calculating the singular value decomposition (SVD) of a matrix (see [6]-[8]). These procedures are numerically stable and well tested. Since A(s) is rational in s we see that $\sigma_{\max}\{A(s)\}$ and $\sigma_{\min}\{A(s)\}$ are both continuous real-valued functions of s for s on Γ. Thus if we evaluate $\sigma_{\max}\{A(s)\}$ (or $\sigma_{\min}\{A(s)\}$) for a finite set of values of s equally spaced along Γ then for adjacent values of s the corresponding maximum (or minimum) singular values will be close. In this section we propose an algorithm which determines the minimum (or maximum) singular value of A(s) for s along a significant portion of the $j\omega$ axis.

To see how this might be done consider an inverse iteration on A ,i.e.; solve recursively

$$A\underline{x}^{(k+1)} = \underline{x}^{(k)} \qquad k=1,2,\ldots$$

starting with some judiciously chosen $\underline{x}^{(0)}$. From the spectral decomposition of A it can be seen that the sequence of vectors $\{\underline{x}^{(k)}\}$ will convergence to the space spanned by the eigenvectors associated with the eigenvalues of A of smallest modulus. If the vectors are re-normalized at each iteration so that $||\underline{x}^{(k)}||_2=1$ then for k large enough a good estimate of the smallest eigenvalue is

$$\lambda_{\min} \simeq \underline{x}^{(k)*}A\underline{x}^{(k)}$$

In practice this works well [13] in the important case when $|\lambda_{\min}| << |\lambda_{\max}|$; i.e., A is ill-conditioned with respect to inversion.

Now recall that if $\sigma\ \varepsilon$ sing(A) then there exist vectors $\underline{x}$ and $\underline{y}$ with appropriate dimensions so that

$$A\ \underline{x} = \sigma\underline{y} \tag{3.1}$$

$$A^*\underline{y} = \sigma\underline{x}$$

where A* is the complex conjugate transpose of A. Here $\underline{x}$ and $\underline{y}$ are called the right and left singular vectors of A associated with the singular value σ, respectively. The right singular vector, $\underline{x}$ is also an eigenvector associated with the eigenvalue $\lambda=\sigma^2$ of A*A and $\underline{y}$ is an eigenvector for the same eigenvalue but associated with AA*.

An inverse iteration applied to the pair of equations

$$\begin{aligned} A(j\omega)\underline{x}^{(k+1)} &= \underline{y}^{(k)} \\ A^*(j\omega)\underline{y}^{(k+1)} &= \underline{x}^{(k)} \end{aligned} \tag{3.2}$$

is the basis for the algorithm which follows.

Algorithm for finding $\|A(j\omega)\|_2$:

Given: $A(j\omega)$, a pxm matrix rational in $j\omega$ with $p \geq m$, $\Delta\omega, \omega_0$, the frequency increment and bandwidth respectively, and $\varepsilon << 1$, a tolerance based on computer precission and required accuracy.

Begin:

Step 1: Set N := $\omega_0/\Delta\omega$ and find the full SVD of

$A(j0)$ by a standard procedure [6].

$$A(j0) = W\Sigma Z^*$$

where $WW^* = I_p$, $ZZ^* = I_m$, and Σ is a pxm matrix with non-zero diagonal elements $\{\sigma_1, \sigma_2, \ldots, \sigma_m\}$ ordered such that $\sigma_1 \geq \ldots \geq \sigma_m \geq 0$.

Save the right/left singular vectors associated with σ_m in

$$\underline{z} := \underline{z}_m \quad , \quad \underline{w} := \underline{w}_m$$

<u>For i := 1 until N,Do:</u>

<u>Step 2</u>: Set $\omega_i := i\ \Delta\omega$ and factor

$$A(j\omega_i) = \tilde{Q}_i\ \tilde{R}_i = [Q_i\ C]\ \tilde{R}_i = Q_i \begin{pmatrix} \tilde{R}_i \\ 0 \end{pmatrix}$$

where $\tilde{Q}_i^*\tilde{Q}_i = I_p$, $Q_i^*Q_i = I_m$, and R_i is right triangular matrix which is the first m rows of $\tilde{R}_i$. Note that Q_i is the first m columns of $\tilde{Q}_i$. Remark: Standard routines are available to perform the QR factorization above which require approximately m^3 operations [15].

<u>Step 3</u>: Set

$$\underline{x}^{(0)} := \underline{z}$$
$$\underline{v}^{(0)} := Q_i^*\ \underline{w}$$

Note that $\underline{x}^{(i)}$, $\underline{v}^{(i)}$, $\underline{z}$ are m-vectors while $\underline{w}$ is a p-vector.

<u>Step 4</u>: Set k := 1

<u>Step 4.1</u>: Solve sequentially

$$R_i^*\ \underline{v}^{(k+1)} = \underline{x}^{(k)}$$

and

$$R_i\ \underline{x}^{(k+1)} = \underline{v}^{(k)}$$

by forward and backward substitution. This requires approximately m^2 operations.

<u>Step 4.2</u>: Normalize the new vectors

$$d_x := \left[\underline{x}^{*(k+1)}\ \underline{x}^{(k+1)}\right]^{1/2}$$
$$d_v := \left[\underline{v}^{*(k+1)}\ \underline{v}^{(k+1)}\right]^{1/2}$$
$$\underline{x}^{(k+1)} := \underline{x}^{(k+1)} / d_x$$
$$\underline{v}^{(k+1)} := \underline{v}^{(k+1)} / d_v$$

<u>Step 4.3</u>: Compare the new vectors with the last pair

<u>If</u> $|\underline{x}^{*(k)}\underline{x}^{(k+1)}+\underline{v}^{*(k)}\underline{v}^{(k+1)}-1| < \varepsilon$
<u>Then</u> Go To Step 4
<u>Else</u> set k:=k+1 and return to step 4.1

<u>Step 5</u>: Compute the minimum singular value of $A(j\omega_i)$ as

$$\sigma_{min}^{(i)} := \mathrm{Re}\ \{\ \underline{v}^{*(k+1)}\ R_i\ \underline{x}^{(k+1)}\ \}$$

<u>Step 6</u>: Save $\underline{z} := \underline{x}^{(k+1)}$ and $\underline{w} := Q_i\ \underline{v}^{(k+1)}$

<u>While</u> i<N Increment i:=i+1 and Return to step 2
<u>Else</u> return the sequence of singular values of $A(j\omega), \{\sigma_{min}^{(i)}\}_{i=1}^{N}$.

This type of iterative algorithm offers a computational advantage over the standard methods for finding the full singular value decomposition <u>only if</u> the iteration (i.e., steps 4.1-4.3) converges rapidly to the desired accuracy. This is why we initialize the algorithm with a full SVD at ω=0 and use the appropriate singular vectors to initialize the next iteration. Whenever convergence is slow we recommend re-initializing by computing another full SVD. This should provide reasonable savings over certain regions of the $j\omega$ axis.

Since for the intended application the computations will be applied to submatrices of the overall system, it is reasonable to assume that these matrices will be of relatively low order. The above algorithm should then have both the efficiency and numerical stability nescessary for the relatively large amount of computations needed.

We mention that a related method for finding singular values that would require the formation of $A^T(-j\omega)A(j\omega)$. A naive approach here would require approximately m^2 floating point operations to form the Hermitian matrix at each ω. However when $A(j\omega)$ is ill-conditioned with respect to inversion this product could easily be dominated by roundoff error. A more sensible approach would involve again obtaining the QR factors of $A(j\omega)$ where the Hermittian form $A^T(-j\omega)A(j\omega) = R^*R$ reveals the Cholesky factors [13,pg.158]. If we then apply inverse iteration to the Hermitian form we would find an algorithm similar to that above except that we have masked the use of the left singular vector of $A(j\omega)$ in the iteration. This could have a potentially detrimental effect on the convergence of the algorithm [14].

Another related method would be to solve for the singular vectors by inverse iteration on the matix $A(j\omega)-I\sigma_{last}$. Here we use the last singular value found at say ω_1, the last data point, as an estimate for the next singular value at $\omega_2=\Delta\omega+\omega_1$. However if the estimate is not close enough we may instead of finding $\sigma_{min}\{A(j\omega_2)\}$ find some other singular value for $A(j\omega_2)$ which is closer to σ_{last} than is $\sigma_{min}\{A(j\omega_2)\}$. This can happen due to too large a step, $\Delta\omega$, or due to clustering of the singular values of $A(j\omega_2)$.

To change the algorithm to find the 2-norm, $||A(j\omega)||_2 = \sigma_{max} A(j\omega)$ merely exchange the indexing of the vectors $\underline{v}$ and $\underline{x}$ in step 3.1 so as to implement the usual power method iteration.

<u>3.4 Considerations Relative to Form of Plant Model Data</u>

We have considered here problems in which the plant model is available as a rational matrix expression in the Laplace variable s. The algorithm above also works well for the case where we obtain real data on the system frequency response from measurements on the system. In this case we have a series of matrices $\{G(j\omega_i)\}_{i=1}^{N}$. In the case that we have a state space realization, $G(j\omega)=C[j\omega I-A]^{-1}B$ it appears that more work must be done. However efficient algorithms exist for calculating $G(j\omega)$ for a large number of $\omega\epsilon\Omega$ [9].

<u>4.0 Conclusions</u>

The advantages associated with decentralized feedback control for large scale multivariable plants are well known. For many practical large scale engineering problems the modeling problem can be greatly simplified with decentralized control since useful physical insights can be brought to bear when modeling subsystem and interaction dynamics separately.

Notions of closed loop robustness and parameter insensitivity are perhaps best established from frequency response models (cf.[10]-[12]). By basing design of decentralized feedback control on certain frequency response measures of subsystem interaction one can obtain measures of closed loop robustness which include robustness to model uncertainties in interaction dynamics. [3]

In the INA method developed by Rosenbrock and his colleagues the decentralized control viewpoint is not emphasized primarily because of a desire to reduce subsystem interaction through the use of series multivariable compensators to the point that single loop (classical) feedback design techniques can be applied. When decentralized control is the objective we prefer (rather than attempt to adjust subsystem interaction) to investigate and exploit "natural" system partitionings and frequency response measures. The flexibility available to exploit these system partitionings for decentralized feedback control based on block diagonal dominance is discussed more fully in [3].

Our purpose in this paper has been to show that the determination of the generalized notion of block diagonal dominance of a partitioned transfer function matrix requires a sizeable-but manageable-burden of computation. In this paper we address the problem of determining whether for a given partitioning the open loop transfer function matrix is BDD. Thus we postpone the design problem of choosing decentralized feedback compensation until BDD has been established for the open loop plant (cf. [3]). We have emphasized two aspects of the computational problem.

First, the design of numerically stable algorithms for computing the frequency dependent matrix norms and dominance margins. Here we emphasize the use of singular values in computing the matrix norms primarily because their spectral properties permit the design of the algorithm in section 3.3. This algorithm is based on a modified inverse power method iteration with well known convergence properties (cf.[15]).

Second, the design of a search algorithm with known convergence properties for evaluating the BDD condition of (4) and (5). Convergence of the proposed grid refinement algorithm is based on assumption 3 which may be difficult to guarantee in practice without applying additional engineering or problem specific insight. Thus we conclude that the natural environment in which to implement these algorithms is an interactive computer-based code. In such an environment the engineer or designer can apply the above algorithms together with engineering insight and problem specific knowledge towards the goal of assessing feasibility of decentralized feedback control for a variety of system partitionings.

These algorithms have been coded on a VAX-1170 computer in the Ratfor language at the Naval Research Laboratory. They are currently being integrated into an interactive package which will be used to assess feasibility of some large space structure decentralized control schemes. Results of these studies will be reported elsewhere.

Acknowledgements

We wish to acknowledge helpful conversations with Prof. D.P. O'Leary, Computer Science Dept., University of Maryland leading to the development of numerical algorithms and conversations with Prof. A. Tits, E.E. Dept., University of Maryland leading to development of the grid refinement algorithm.

References

[1] Rosenbrock, H. H., Computer-Aided Control System Design, Academic Press, Inc., London, 1974

[2] Rosenbrock, H.H., "Design of multivariable control systems using the Inverse Nyquist Array", Proc. IEEE Vol. 116, No. 11, Nov. 1969, pp.1929-1936

[3] Bennett,W.H. & Baras,J.S. "Block Diagonal Dominance and the Design of Decentralized Feedback Compensation", to appear (see also Proc. of Conf. on Information Sciences & Systems, Johns Hopkins Univ., 1979)

[5] N. Munro & V. Zakian, "Inversion of Rational Polynomial Matrices", Electronic Letters, Sept. 1970, Vol. 6, No. 19, pp. 628

[6] J. J. Dongarra, C. B. Moler, J. R. Bunch, G. W. Stewart, LINPACK Users Guide, SIAM Press, 1979

[7] G. Golub and W. Kahan, "Calculations the Singular Values and Pseudo-Inverse of a Matrix", J. SIAM Numer Anal., Vol. 2, No. 2, 1965, pp. 205-224

[8] G. Golub and C. Reinsch, "Singular Value Decomposition and Least Squares Solutions," Numer. Math., Vol. 4, 1970, pp. 403-420

[9] A. Laub, "Efficient Multivariable Frequency Response Computations", Proc. of 19th IEEE CDC, December 1980, pp.39

[10] J.C. Doyle and G. Stein, "Multivariable Feedback Design: Concepts for a Classical/Modern Synthesis", IEEE Trans. on Automatic Control: Special Issue on Linear Multivariable Control Systems, Vol. AC-20. No. 1, pp. 4-16

[11] I. Postlethwaite, J. M. Edmonds, and A. G. J. MacFarlane, "Principal Gains and Principal Phases in the Analysis of Linear Multivariable Feedback Systems", IEEE Trans. on Automatic Control: Special Issue on Linear Multivariable Control Systems, Vol. AC-20. No. 1, pp. 32-46

[12] M. G. Safonov, A. J. Laub, and G. L. Hartman, "Feedback Properties of Multivariable Systems: The Role and Use of the Return Difference Matrix", IEEE Trans. on Automatic Control: Special Issue on Linear Multivariable Control Systems, Vol. AC-20. No. 1, pp.47-65

[13] G. Dahlquist and A. Bjorck Numerical Methods, Prentice Hall, Inc. 1974

[14] A.K. Cline,C.B. Moler,G.W. Stewart,& J.H. Wilkinson, "An Estimate for the Condition Number of a Matrix", SIAM Num. Anal., No.16, 1979, pp.368-375

[15] G. W. Stewart, Introduction to Matrix Computations, Academic Press, 1973

[16] E. Polak and D. Q. Mayne, On The Solution of Singular Value Inequalities Over a Continuum of Frequencies, report UCB/ERL M80/8, Univ. of Calif., Berkeley, CA, 1980.