Technical Notes and Correspondence

Projection and Deflation Methods for Partial Pole Assignment in Linear State Feedback

YOUCEF SAAD

Abstract—**Two projection methods are proposed for partial pole placement in linear control systems. These methods are of interest in the common situation where the system is very large and only a few of its poles must be assigned. The first method is based on computing an orthonormal basis of the left invariant subspace associated with the eigenvalues to be assigned and then solving a small inverse eigenvalue problem resulting from projecting the initial problem into that subspace. The second method can be regarded as a variant of the Wielandt deflation technique used in eigenvalue methods.**

I. Introduction

This note is concerned with the problem of assigning the poles of the single-input continuous-time linear control system

$$\dot{x} = Ax + bu \tag{1.1}$$

where A is a constant $N \times N$ matrix, b is a vector, and u is a scalar function of t. For simplicity, we start with the single input case although our results can readily be extended to multiple input control systems as will be seen in Section V. We will interchangeably speak of the eigenvalues of the matrix A of the system (1.1) and the poles of its corresponding transfer function $H(s) \equiv (sI - A)^{-1}b$. In the single input situation, the pole assignment problem consists of finding a feedback vector f, such that the closed-loop system

$$\dot{x} = (A - bf^T)x + bv$$

has desired poles. In other words the linear algebra problem is to find a vector f so that the perturbed matrix $A - bf^T$ has the desired eigenvalues. This *state feedback* technique constitutes one of the most popular ways of modifying dynamical behaviors of time-invariant linear control systems [26].

Several methods to solve this problem are available for the case when the matrix A is small, the one generally preferred being an analog of the QR method developed separately by Miminis and Paige [9] and Petkov [16].

However, in many realistic situations the matrix A is so large that the use of a QR-like algorithm becomes impractical. This arises, for example, in large space structure control [2] and in the control of electrical networks [8]. Using these techniques to solve such large pole assignment problems is not only uneconomical, but also untrustworthy. A general consensus seems to be that assigning the poles of a system of order, say, 50 may be the upper limit of what can be reliably achieved with standard double precision and these classical numerical methods.

What is often desired in practice is to modify the eigenvalues of the original system to make it stable. For the continuous-time problem (1.1), this means that the eigenvalues with positive real parts must be shifted to the left-half plane. In reasonable practical situations one can expect that the original system is nearly stable in the sense that only a few of its eigenvalues are located in the right-half plane. The problem is then to place only those unstable eigenvalues in the left-half plane and leave the others unchanged.

The main contribution of this note is to provide two numerical methods to solve this problem which are suitable for large scale systems. The first one, described in Section II, is a projection method in which the system is projected onto the left invariant subspace associated with the unstable poles. The second, described in Section III, is based on the Wielandt deflation process well known in numerical linear algebra and consists of assigning the various unstable poles one at a time.

Concerning notation, the norm $\|\cdot\|$ represents the Euclidean norm in C^N. Given two vectors x and y in C^N, their complex inner product will often be represented by (x, y), and the acute angle between them by $\theta(x, y)$. The transpose of a matrix X will be denoted by X^T and the transpose of its complex conjugate by X^H. We will denote by $\Lambda(X)$ the spectrum of a square matrix X.

II. An Algorithm for Partial Pole Assignment by State Feedback

A. The Projection Process

Let A be an $N \times N$ real nonsymmetric matrix whose eigenvalues are

$$\lambda_1, \lambda_2, \cdots, \lambda_k, \lambda_{k+1}, \cdots, \lambda_N.$$

Let b be a given real vector on which we will make some additional mild assumptions later. The problem considered is to find a vector f so that the modified matrix

$$M = A - bf^T \tag{2.1}$$

has the given eigenvalues

$$\mu_1, \mu_2, \cdots, \mu_k, \lambda_{k+1}, \cdots, \lambda_N.$$

In other words, we would like to assign the eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_k$ of A into $\mu_1, \mu_2, \cdots, \mu_k$, while leaving the rest of the spectrum of A unchanged, and this with the rank one perturbation $-bf^T$. We refer to this as *the partial pole assignment problem.* We assume that both sets of eigenvalues $\{\lambda_1, \lambda_2, \cdots, \lambda_k\}$ and $\{\mu_1, \mu_2, \cdots, \mu_k\}$ are symmetric with respect to the real axis, i.e., if a complex value is in one of the sets then so is its conjugate. Moreover, if a multiple eigenvalue belongs to the set $\{\lambda_1, \lambda_2, \cdots, \lambda_k\}$ we assume that it is represented several times in the set according to its algebraic multiplicity.

In order to solve this problem we will need an orthonormal basis of the *left* invariant subspace of A associated with the eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_k$. Let us assume that we have computed the partial Schur factorization for A^T

$$A^TQ = QR \tag{2.2}$$

where Q is an $N \times k$ matrix whose columns form an orthonormal basis of the left invariant subspace associated with λ_i, $i = 1, \cdots, k$ and R is a $k \times k$ upper quasi-triangular matrix. The column vectors q_i, $i = 1, 2, \cdots, k$ of Q will be referred to as the left Schur vectors of A. We will seek a solution f in the form

$$f = Qg. \tag{2.3}$$

Manuscript received January 23, 1986; revised April 20, 1987. Paper recommended by Past Associate Editor, J. B. Pearson. This work was supported in part by the Office of Naval Research under Contract N00014-82-K-0184, by the Department of Energy under Contract DEAC02-81ER0996, and by the Army Research Office under Contract DAAG-29-83-0177.

The author is with the University of Illinois, Urbana, IL 61801-2932.

IEEE Log Number 8716991.

0018-9286/88/0300-0290$01.00 © 1988 IEEE

Consider the matrix M^TQ

$$M^TQ=[A^T-fb^T]Q=QR-Qgb^TQ \equiv Q[R-gs^T]$$

which can be rewritten as

$$M^TQ=Q[R^T-sg^T]^T \equiv QC_k^T. \tag{2.4}$$

The above equation means that the choice (2.3) makes the subspace spanned by Q also invariant under M^T. Moreover, the eigenvalues of the matrix M associated with this invariant subspace are the eigenvalues of the $k \times k$ matrix $C_k = R^T - sg^T$. Under certain simple conditions, which will be clarified shortly, the eigenvalues of this small matrix can be assigned to be μ_i, $i = 1, \cdots, k$, by an appropriate choice of the vector g.

Therefore, let g be chosen so that

$$\Lambda\{C_k\} \equiv \Lambda\{R^T-sg^T\}=\{\mu_i, i=1, \cdots, k\}. \tag{2.5}$$

We will refer to the above problem as *the projected problem*. Clearly, if k is small, the projected problem can be solved by standard pole assignment techniques. As was seen above this choice of g transforms the eigenvalues λ_i, $i = 1, \cdots, k$ of A into μ_i, $i = 1, \cdots, k$. What is interesting is that, in addition, the remaining eigenvalues of A are unchanged as is stated in the following theorem.

Theorem 2.1: Let g be determined so as to solve the projected $k \times k$ problem (2.5) and let f be given by (2.3). Then the matrix $M = A - bf^T$ has the eigenvalues $\mu_1, \mu_2, \cdots, \mu_k, \lambda_{k+1}, \cdots, \lambda_N$.

Proof: Let $W = [w_1, w_2, \cdots, w_{N-k}]$ be the remaining Schur vectors of A^T associated with the remaining eigenvalues λ_i, $i > k$. In other words, W is an orthonormal basis of the orthogonal complement of Span$\{Q\}$ and is such that the matrix $X \equiv [Q, W]$ is orthogonal. Using (2.4) we have

$$Q^TM^TQ=C_k^T; \quad W^TM^TQ=0.$$

From (2.3) we get,

$$Q^TM^TW=Q^T(A^T-Qgb^T)W \equiv Q^TA^TW+E^T;$$

$$W^TM^TW=W^T(A^T-Qgb^T)W=W^TA^TW.$$

Hence,

$$X^TMX=\begin{pmatrix} C_k & O \\ W^TAQ+E & W^TAW \end{pmatrix}. \tag{2.6}$$

The proof follows immediately. ■

Based on the above theorem, an algorithm for solving the partial pole assignment problem is the following.

Algorithm 1: Partial Pole Assignment by Projection

1) Compute the k eigenvalues to be assigned and the associated partial Schur decomposition (2.2).

2) Compute $s = Q^Tb$ and solve the projected inverse eigenvalue problem, i.e., find g such that $\Lambda\{R^T - sg^T\} = \{\mu_1, \mu_2, \cdots, \mu_k\}$.

3) Form the feedback solution vector $f = Qg$.

We should mention that in contrast to the Schur method presented by Varga [24], the above algorithm requires only a partial Schur factorization. Another important point to note is that the partial Schur decomposition (2.2) is by no means indispensible. All we need in order to use Algorithm 1 is an orthonormal basis Q of the invariant subspace associated with $\lambda_1, \cdots, \lambda_k$. The matrix R which represents the linear operator A^T in that basis is then not necessarily quasi-triangular.

The cost of Algorithm 1 is dominated by the first step which basically consists of computing a k-dimensional invariant subspace associated with the k eigenvalues to be assigned. Numerical methods for this purpose and related matters are briefly discussed next.

B. Computing Invariant Subspaces of Nonsymmetric Matrices

Work on numerical methods for computing invariant subspaces of general large matrices is still in the experimental stage; see the survey by Parlett [13]. These methods should be used with caution because eigenproblems can be so sensitive to variations in the original data, namely the matrix A, that any procedure to approximate eigenvalues or eigenvectors of A will encounter serious difficulties. The sensitivity of a given eigenvalue λ_i of A to perturbations is usually measured by its condition number which is defined as the inverse of the cosine of the angle between its corresponding right and left eigenvectors y_i and z_i, i.e.,

$$c(\lambda_i)=\frac{\|z_i\| \, \|y_i\|}{|(z_i, y_i)|}. \tag{2.7}$$

In practice, this means that a small perturbation to A, of norm ϵ, may disturb the eigenvalue λ_i by as much as $c(\lambda_i)\epsilon$.

Similarly, the condition number for the eigenvector y_i involves the reduced resolvent $S(\lambda_i)$, defined as the inverse of the restriction of $A - \lambda_i I$ to $\{z_i\}^\perp$, the subspace orthogonal to the left eigenspace associated with λ_i; see [3, p. 17]

$$c(y_i)=\|S(\lambda_i)\|. \tag{2.8}$$

Although not apparent from the definition, the condition number for the eigenvector is implicitly related to that for the eigenvalues of A; see Wilkinson [25]. Moreover, it is easy to show from the definition that [3]

$$c(y_i) \geq \max_{j \neq i} \frac{1}{|\lambda_j - \lambda_i|}.$$

While there may be practical situations with disastrously large condition numbers for nonsymmetric problems, this cannot happen in the symmetric case since the condition number of an eigenvalue is always unity. Thus, when dealing with large nonsymmetric eigenvalue problems we should expect the task to be much more arduous than that of a symmetric problem of the same size.

In the simple situation where the eigenvalues λ_i, $i = 1, \cdots, k$ are real and simple, then the Schur vectors q_i, $i = 1, 2, \cdots, k$ can be obtained from a Gram–Schmidt orthogonalization of the eigenvectors z_i, $i = 1, 2, \cdots, k$ of A^T associated with λ_i, $i = 1, \cdots, k$ in this order. In this case the Schur vectors are unique apart from a sign, whenever the order of the eigenvalues is fixed. If an eigenvalue λ_i is complex, one can take the real part of z_i and orthogonalize it against the previous Schur vector q_j, $j = 1, \cdots, j - 1$, and then do the same with its imaginary part, to get the Schur vectors q_i and q_{i+1}. In case an eigenvalue is multiple but not defective, then the eigenvalues can again be orthogonalized against the previous q_j's to obtain the next Schur vectors. However, in these cases the Schur vectors are not uniquely determined.

Standard methods for computing eigenvalues/eigenvectors of matrices, such as the well-known QR algorithm, become too costly as the size of the matrix increases. Moreover, these methods do not exploit the fact that we may be seeking for a very small number of eigenvalues, nor do they take advantage of any sparsity pattern of A. Several special methods for computing a few eigenvalues and eigenvectors of large sparse matrices have been developed in the literature. We will not give a detailed account of the various techniques available, but we can mention some of the basic principles.

An idea that is basic to sparce eigenvalue calculations is that of projection processes [19]. Given a subspace spanned by a system of m orthonormal vectors $V = [v_1, \cdots, v_m]$, a projection process onto $K \equiv$ *span*$\{v\}$ consists of approximating some of the eigenvalues of A by those of the $m \times m$ matrix $C = V^TAV$. The corresponding approximate eigenvectors are the vectors Vy_i where y_i are the eigenvectors of C. Similarly, the approximate Schur vectors are the vector columns of VU, where $U = [u_1, u_2, \cdots, u_m\}$ are the Schur vectors of C, i.e., U^HCU is quasi-upper triangular.

Based on this, one of the simplest methods for computing invariant subspaces is the so-called subspace iteration methods well known to structural engineers [5], [23]. In its simplest form, the subspace iteration can be described as follows.

1) Choose an initial orthonormal system $V_0 \equiv [v_1, v_2, \cdots, v_m]$ and an integer k.

2) Compute $X = A^kV_0$ and orthonormalize X to get V.

3) Perform a projection process with V.

4) Test for convergence. If satisfied then exit else continue.

5) Take $V_0 = VU$, the set of approximate Schur vectors (alternatively take $V_0 = VY$, the set of approximate eigenvectors), choose a new k and go to 2.

The above algorithm utilizes the matrix A only to compute successive matrix by vector products $w = Av$, so sparsity can be exploited. However, it faces two drawbacks. The first is that it is known to be a slow method, much slower than some of the alternatives to be described next. The other is that it computes the eigenvalues of *largest modulus* of A not those with largest real parts. In some cases the eigenvalues with largest real parts may also be those with largest modulus in which case the method can be used. But a subspace iteration algorithm in the simple form given above cannot be used as a general purpose technique. A more satisfactory alternative is to use a Chebyshev–subspace iteration: step 2) is replaced by $X = t_k(A)V_0$, where t_k is obtained from the Chebyshev polynomial of the first kind of degree k, by a linear change of variables. The three-term recurrence of the Chebyshev polynomial allows us to compute a vector $w = t_k(A)v$ at almost the same cost as $A^k v$. Moreover, it is then possible to compute the desired right-most (or left-most) eigenvalues of A. Also, performance is improved as this is the usual primary reason for using the Chebyshev iteration. Details on implementation can be found in [20].

A second method used in the literature is the Arnoldi process [1], [22] which is a projection process onto $K_m = span\{v_1, Av_1, \cdots, A^{m-1}v_1\}$. The algorithm starts with some nonzero vector v_1 and generates the sequence of vectors v_i from the recurrence

$$v_{j+1} = \frac{1}{h_{j+1,j}}\left(Av_j - \sum_{i=1}^{j} h_{ij}v_i\right)$$

where the scalars $h_{i,j}$, $i = 1, \cdots, j+1$ are chosen so that v_{j+1} is orthogonal to all previous v_i's and is of norm unity.

It is straightforward to show that the system $V_m \equiv \{v_1, v_2, \cdots, v_m\}$ forms an orthonormal basis of the subspace K_m. We can then use V_m for a projection process onto K_m and it turns out that the matrix $V_m^T A V_m$ which is needed for this purpose is nothing but the upper Hessenberg matrix H_m whose nonzero elements are the scalars $h_{i,j}$ generated by the algorithm.

As m increases, the eigenvalues of H_m that are located in the outmost part of the spectrum start converging towards corresponding eigenvalues of A. The difficulty with the above algorithm is that as m increases, cost and storage increase rapidly. One solution is to use the method iteratively: m is fixed and the initial vector v_1 is taken at each new iteration as a linear combination of some of the approximate eigenvectors. Moreover, there are several ways of accelerating convergence by preprocessing v_1 by a Chebyshev iteration before restarting, i.e., by taking $v_1 = t_k(A)z$ where z is again a linear combination of eigenvectors.

A technique related to Arnoldi's method is the nonsymmetric Lanczos algorithm [15], [4] which delivers a nonsymmetric tridiagonal matrix instead of a Hessenberg matrix. Unlike Arnoldi's process, this method requires multiplications by both A and A^T at every step. On the other hand, it has the big advantage of requiring little storage (five vectors). Although no comparisons of the performances of the Lanczos and the Arnoldi type algorithms have been made, the Lanczos methods are usually recommended whenever the number of eigenvalues to be computed is large.

Finally, if the matrix is banded, an efficient solution is the shift and invert strategy which consists of using one of the above iterative methods (subspace iteration, Arnoldi, or Lanczos) for the matrix $(A - \sigma I)^{-1}$, where σ is some shift chosen say at the center of some small region of the complex plane where eigenvalues are sought. The matrix $(A - \sigma I)^{-1}$, need not be explicitly computed: all we need is to factor $(A - \sigma I)$ into LU and subsequently at each step of the iterative method solve two triangular systems, one with L and the other with U. Thus, band structure can be fully exploited. In [14] several implementations of the shift and invert strategy are considered and the problem of avoiding complex arithmetic when A is real is addressed.

For Algorithm 1 to be practically feasible the number k of unstable modes must be relatively small. An acknowledged weakness of our projection procedure is that it may not be known *a priori* how many eigenvalues must be assigned until they are computed. Ideally, we would like all the eigenvalues with positive real parts and only those to be assigned values in the complex left-half plane.

When A is symmetric a well-known property, referred to as Sylvester's inertia theorem, states that if we factor A into LU, then the number of positive eigenvalues of A is identical to the number of positive diagonal elements in U [12], [25]. As a result in the course of a shift and invert based algorithm, where matrices of the form $(A - \sigma I)$ are factored, this property can be efficiently used to determine the number of unstable eigenvalues [25] at the same time as the eigenvalue calculation proceeds. In fact, this technique is now widely used in standard codes for structural engineering [18]. Unfortunately, its applicability is limited to symmetric matrices.

An alternative way in which the difficulty can be handled is to keep computing eigenvalues with positive real parts until no more are found or until the user decides that there are too many eigenvalues to be assigned and takes some appropriate measure. Therefore, a computational code based on this approach must incorporate a parameter $k_{\max}$ which is the maximum allowable number of poles to be placed.

C. Existence of a Solution

A look at the proof of Theorem 2.1 reveals that a solution of the form $f = Qg$ to the partial pole placement exists if and only if a solution to the projected problem exists. As a consequence of a well-known result a solution exists *for any set* $\mu_1, \cdots, \mu_k$ if and only if the system (R^T, s) is controllable [26]. We will refer to this as the partial controllability condition for (A, b) in the subspace Span$\{Q\}$. As is well known, (R^T, s) is controllable iff

$$\text{rank }\{s, R^Ts, \cdots, (R^T)^{k-1}s\} = k. \tag{2.9}$$

Denoting by Π_k the orthogonal projector onto the subspace spanned by Q, it is clear that R^T is a matrix representation of the linear operator $\Pi_k A \Pi_k$ in the basis Q while QR^TQ^T is a matrix representation in the original (canonical) basis. Here, we have abused the notation by using the same symbol A for the matrix and the linear operator that it represents. The matrix representation of Π_k in the original basis is QQ^T. An important observation is that

$$\Pi_k A \Pi_k = \Pi_k A \tag{2.10}$$

because the subspace Span$\{Q\}$ is invariant under A^T and, therefore, $\Pi_k A^T \Pi_k = A^T \Pi_k$, which yields the result by transposition. Notice that we also have

$$\Pi_k A^j \Pi_k = \Pi_k A^j, \qquad \forall j \tag{2.11}$$

since Span$\{Q\}$ is invariant under $(A^T)^j$. We can now formulate a condition for the existence of a solution.

Proposition 2.1: The system (1.1) is partially controllable in the subspace Span(Q), i.e., the partial pole assignment problem has a solution of the form (2.3) for any set $\mu_1, \cdots, \mu_k$, if and only if the following condition holds:

$$\dim [\Pi_k \text{ span } \{b, Ab, \cdots, A^{k-1}b\}] = k.$$

Proof: Multiplying each vector of the system (2.9) by the matrix Q and replacing QQ^T by Π_k, (2.9) becomes rank $\{\Pi_k b, \Pi_k A \Pi_k b, \cdots, (\Pi_k A \Pi_k)^{k-1} b\} = k$. Then using (2.10) and (2.11) we get the condition rank $\{\Pi_k b, \Pi_k A \Pi_k b, \cdots, \Pi_k A^{k-1} \Pi_k b\} = k$; and by a final application of (2.11) we obtain the desired result. ■

The proposition can be interpreted as follows. The system is partially controllable in the sense of the proposition iff the so-called Krylov subspace $K_k \equiv$ Span $\{b, Ab, \cdots, A^{k-1}b\}$ is of full rank and contains no vector orthogonal to the left invariant subspace associated with the unstable eigenvalues. For example, it is clear that when $\Pi_k b = 0$ the system is not partially controllable. As a result the size of $\|\Pi_k b\|$ as compared to the size of $\|b\|$ can be a good first measure of how controllability has deteriorated by restricting the problem into the invariant subspace. To make this statement more accurate, it is helpful to use a measure of controllability introduced by Paige [11] which is defined as the distance from the original system (A, b) to the closest uncontrollable system, or

$$\mu(A, b) = \text{Min } \|(\delta A, \delta b)\| \text{ s.t. } (A+\delta A, b+\delta b) \text{ is uncontrollable.}$$

Our situation is a particular case since we restrict the solution candidates f to lie in Span$\{Q\}$. If we allowed δA in the above definition to be arbitrary and nonzero, then Q will change and the notion of near partial controllability in Span$\{Q\}$ will be difficult to define. For convenience, one may restrict the above definition by taking $\delta A = 0$. The corresponding measure, which was also hinted at in [11], defines a natural notion of partial controllability in Span$\{Q\}$ as follows:

$$\mu_Q(A, b) = \text{Min } \|\delta b\| \text{ s.t. } (A, b+\delta b) \text{ is uncontrollable in Span } \{Q\}.$$

Observing that the vector $b - \Pi_k b$ is orthogonal to Span$\{Q\}$, we note that the system $(A, b - \Pi_k b)$ is not controllable in Span$\{Q\}$. Therefore,

$$\mu_Q(A, b) \le \|\Pi_k b\|$$

which provides an upper bound for the distance of the closest Span$\{Q\}$ − uncontrollable system from (A, b). Intuitively, the situation when $\|\Pi_k b\|$ is small will be difficult to treat because the problem is nearly *unsolvable*. This is observed in the numerical experiments and will be further discussed shortly.

Note that the problem of assigning only a few *bad* eigenvalues, instead of all of them as is traditionally done, is not new and some theory on the existence of a solution in a general context is developed by Wonham [26]. Our context is limited by the fact that we look for a solution of a particular form, namely the form (2.3).

III. Use of Deflation Methods

A well-known technique in eigenvalue methods is the so-called Wielandt deflation; see Wilkinson [25, pp. 596-599]. Suppose that we have computed the eigenvalue λ_1 of largest modulus and its corresponding eigenvector y_1 of a given matrix A by some algorithm such as, in the simplest case, the power method. Assume further for simplicity that λ_1 is real. A common problem is to compute the next dominant eigenvalue λ_2 of A. An old artifice for achieving this is to use a deflation procedure: a rank one modification of the original matrix is performed so as to displace the eigenvalue λ_1 to the origin, while keeping all other eigenvalues unchanged. Thus, the eigenvalue λ_2 becomes the dominant eigenvalue of the modified matrix and, therefore, the power method can subsequently be applied to this matrix to compute the next dominant pair λ_2, y_2. In contrast to other deflation techniques, Wielandt's deflation requires only the knowledge of the right eigenvector. The deflated matrix is of the form

$$A_1 = A - \frac{\sigma}{b^T y_1} y_1 b^T \tag{3.1}$$

where b is an arbitrary vector not orthogonal to y_1 and σ is an appropriate shift. The following proposition relates the eigenvalues and eigenvectors of A to those of A_1.

Proposition 3.1: Let A have eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_N$, with associated right eigenvectors $y_1, y_2, \cdots, y_N$ and left eigenvectors $z_1, z_2, \cdots, z_N$ and let A_1 be defined by (3.1). Then the eigenvalues of A_1 are $\lambda_1 - \sigma, \lambda_2, \cdots, \lambda_N$. The corresponding right eigenvectors are

$$y_1, y_2 - \alpha_2 y_1, \cdots, y_i - \alpha_i y_1, \cdots, y_N - \alpha_N y_1$$

in which

$$\alpha_i = \frac{\sigma b^T y_i}{b^T y_1 (\lambda_i - \lambda_1 + \sigma)} \tag{3.2}$$

and the corresponding left eigenvectors are $\hat{z}_1, z_2, z_3, \cdots, z_N$, where

$$\hat{z}_1 = \frac{1}{\|b\|} (A - (\lambda_1 - \sigma) I)^{-H} b.$$

Here X^{-H} stands for the inverse of X^H and the normalization by $\|b\|$ in $\hat{z}_1$ is for convenience. The proof of the proposition consists of a simple verification. Note that the formula (3.2) breaks down whenever $\sigma = \lambda_1 - \lambda_i$, i.e., whenever the eigenvalue λ_1 is moved to any other eigenvalue λ_i of A. In this case if λ_i is simple before the deflation, it becomes a double eigenvalue of A_1 and, since y_1 is its only associated eigenvector, λ_i is also defective.

In the particular case when the direction of the vector b coincides with that of the left eigenvector z_1, then $\alpha_i = 0$, $i = 2, \cdots, N$ and all left and right eigenvectors of A are preserved. The particular deflation technique corresponding to this choice of b is known as Hotelling's deflation. One of the difficulties with this form of deflation is that when the eigenvalue λ_1 is poorly conditioned then this results in large modifications to the matrix A because of the division by $b^T y_1$ in (3.1), and this may result in numerical difficulties.

Going back to the pole assignment problem, let us assume that $k = 1$, i.e., that there is only one eigenvalue to assign. From (2.3) the solution f is of the form $f = \gamma_1 y_1$ where here γ_1 is a scalar. It becomes clear that the technique described in the previous section is nothing but a Wielandt deflation technique for the matrix A^T since the transpose of the modified matrix has the form

$$A^T - \gamma_1 y_1 b^T.$$

To shift the eigenvalue λ_1 to μ_1 we must take $\gamma_1 = (\lambda_1 - \mu_1) b^T y_1$.

According to the above proposition, this deflation technique has the property of changing the eigenvalue λ_1 into μ_1 while leaving the others unchanged. Moreover, the *right* eigenvector of A^T associated with the eigenvalue λ_1 is preserved and the *left* eigenvectors associated with the *remaining* eigenvalues are unchanged. However, the left eigenvector of A^T associated with λ_1 (i.e., the right eigenvector of A associated with λ_1) does change except in the particular case when the vector b happens to be exactly the left eigenvector of A^T associated with λ_1 (Hotelling's deflation).

This suggests an alternative to Algorithm 1 which consists of moving one pole at a time. For simplicity, we present the new algorithm only for the case where the eigenvalues $\lambda_1, \cdots, \lambda_k$ are real and simple. The idea of this approach is to construct the solution f progressively by accumulating multiples of left eigenvectors of the deflated matrices. Thus, in the first step we deflate A into $A_1 = A - \gamma_1 b y_1^T$ where γ_1 is chosen so as to shift the eigenvalue λ_1 of A into μ_1. The eigenvalues $\lambda_j, j > 1$ are preserved as well as the corresponding right eigenvectors z_j. We then compute the right-most eigenvalue λ_2 together with its left eigenvector y_2 by one of the algorithms described in Section II-B. Then A_1 is deflated in turn into $A_2 = A_1 - \gamma_2 b y_2^T$, where γ_2 is again appropriately chosen. The eigenvalue λ_2 is moved into μ_2 while the other eigenvalues namely μ_1 and $\lambda_3, \cdots, \lambda_N$ are unchanged. This procedure is continued until exhaustive of all unstable modes. In the process the ith deflated matrix A_i will have eigenvalues $\mu_1, \mu_2, \cdots, \mu_i, \lambda_{i+1}, \cdots, \lambda_N$.

Algorithm 2: Partial Pole Assignment by Successive Deflations

Start: Choose $k_{\max}$ the maximum number of poles to place. Set $f_0 := 0$, $A_0 := A$, $i := 1$.

Iterate:

1) Compute the eigenvalue λ_i with largest real part of A_{i-1} together with its left eigenvector y_i.

2) If Re$(\lambda_i) < 0$ then exit [No more unstable eigenvalues]

3) Else compute

$$f_i := f_{i-1} + \gamma_i y_i, \qquad \text{where} \quad \gamma_i = \frac{\lambda_i - \mu_i}{b^T y_i} \tag{3.3}$$

and define $A_i = A - b f_i^T = A_{i-1} - \gamma_i b y_i^T$.

4) If $i < k_{\max}$ then set $i := i + 1$ and go to 1, else exit [too many eigenvalues to assign].

Obviously, the matrix A_i is never computed explicitly in step 3), since all that is required by most of the methods described in Section II-B for computing eigenvalues and eigenvalues of large matrices, are matrix by vector multiplications $x \to A_i x$. These operations can be performed by storing A and the vectors b and f_i. The final vector f_i provided by the above algorithm is a linear combination of the eigenvectors $y_1, y_2, \cdots, y_i$, and it is rather straightforward to show that these vectors are in turn linear combinations of the left Schur vectors $q_1, \cdots, q_i$ of A. As a result, f_i is the desired solution to the partial pole placement problem, i.e., it is equal to the solution provided by Algorithm 1.

If a computed eigenvalue λ_i is complex, complex arithmetic can be avoided by replacing step 3) by $f_i = f_{i-2} + \gamma_i \operatorname{Re}(y_i) + \gamma_{i-1} \operatorname{Im}(y_i)$ in which γ_{i-1} and γ_i are obtained by solving a projected 2×2 inverse eigenvalue problem that transforms λ_i, $\bar{\lambda}_i$ into μ_i, $\bar{\mu}_i$. If a computed eigenvalue is multiple, then the algorithm is still usable even if the eigenvalue is defective [just replace *the eigenvalue* by *any one of the eigenvalues* in step 1)]

A small advantage of this algorithm over Algorithm 1 is that there is no need to solve a $k \times k$ inverse eigenvalue problem. However, k is usually small relative to N and so the cost of solving the projected problem is negligible. A disadvantage of the new algorithm is that if we encountered numerical difficulties with some eigenvalue λ_i then *all* the following matrices $A_i, A_{i+1}, A_{i+2}, \cdots$ may be badly perturbed or may lead to badly conditioned eigenvalues. Moreover, since eigenvalues and eigenvectors are obtained one at a time, it is not possible to permute these eigenvectors so as to minimize the effect of a bad vector. In other words, the algorithm is sensitive to the order of the eigenvalues. In contrast, Algorithm 1 only requires an orthonormal basis of the invariant subspace. It is perfectly possible to select this basis in some optimal way, for example so as to minimize the effect of numerical cancellation in the modified Gram–Schmidt algorithm in a subspace iteration type method.

An important question to ask is how do the conditionings of the eigenvalues of the modified problem change under the deflation process (3.1). In fact, a more general question is how do the sensitivities vary under the more general transformation of Algorithm 1. A comparison of the expressions (2.6) and the analogous expression (2.6) and the analogous expression for A

$$X^TAX = \begin{pmatrix} R^T & O \\ W^TAQ & W^TAW \end{pmatrix}$$

indicates that a crude measure of the variation of the sensitivity is the norm of $E = W^Tbg^T$, which vanishes when b is in the right invariant subspace of A, associated with the k unstable eigenvalues. We do not pursue this question on the sensitivity of the modified matrix in its general form here but we mention that a detailed analysis has been developed by Nichols [10]. However, we would like to consider the question in some detail in the simplest case of the transformation (3.1). As was seen above the condition numbers of the closed-loop system will change for a general vector b while it will be unchanged in the very particular case when b belongs to the left eigenspace of A^T (i.e., to the right eigenspace of A). The following proposition results from the expressions of the eigenvectors of A_1 given by Proposition 2.2.

Proposition 3.2: If we assume that the eigenvalues of A are simple and call c_i the condition number of the eigenvalue number i in the modified problem then, using the notation of Proposition 2.2,

$$c_1 = |\sigma| \frac{\|\hat{z}\|}{\cos \theta(b, y_1)} \tag{3.4}$$

and

$$c_i = \frac{\|y_i - \alpha_i y_1\|}{\|y_i\|} c(\lambda_i), \qquad i = 2, \cdots, N. \tag{3.5}$$

Proof: We have

$$c_1 = \frac{\|\hat{z}_1\| \, \|y_1\|}{|(\hat{z}_1, y_1)|} = \frac{\|\hat{z}_1\| \, \|b\| \, \|y_1\|}{|((A - (\lambda_1 - \sigma)I)^{-H} b, y_1)|}$$

$$= \|\hat{z}_1\| \frac{\|b\| \, \|y_1\|}{|(b, (A - (\lambda_1 - \sigma)I)^{-1} y_1)|} \cdot$$

Since y_1 is an eigenvector of A associated with λ_1 this becomes

$$c_i = |\sigma| \, \|\hat{z}_1\| \frac{\|b\| \, \|y_1\|}{|(b, y_1)|}$$

which proves the desired result for c_1. For $i > 1$ we have

$$c_i = \frac{\|z_i\| \, \|y_i - \alpha_i y_1\|}{|(z_i, y_i - \alpha_i y_1)|} \cdot$$

Since we assume that the eigenvalues of A are simple, the inner product of z_i and y_1 vanishes and the above expression simplifies into

$$c_i = \frac{\|z_i\| \, \|y_i - \alpha_i y_1\|}{|(z_i, y_i)|} = \frac{\|z_i\| \, \|y_i\|}{|(z_i, y_i)|} \cdot \frac{\|y_i - \alpha_i y_1\|}{\|y_i\|}$$

which yields the result. ■

The above expressions yield an interesting analysis. The condition of the first eigenvalue depends essentially on three factors. The first is the multiplicative factor $|\sigma|$. The second is the norm of $\hat{z}_1$ which depends in part on where the eigenvalue μ_1 is placed: if it is placed on another eigenvalue, then the condition number of μ_1 is *infinite* which is due to the fact that the eigenvalue becomes defective as was seen earlier. The third factor is the acute angle between b and y_1; the larger this angle the worse the condition number of μ_1. This angle has also been shown to be related to the measure of partial controllability.

For the other eigenvalues, it is preferable for the discussion to normalize the vectors y_i and y_1 and call $\hat{\alpha}_i$ the corresponding new α_i in (3.2). Then the relation (3.5) indicates that the original condition number is not amplified only when

$$|\hat{\alpha}_i| \leq 2 \cos \theta(y_1, y_i).$$

Unfortunately, the expression of $\hat{\alpha}_i$ does not give a simple interpretation of this condition. However, it is obvious from the expression (3.2) that we always have

$$c_i \leq (1 + |\hat{\alpha}_i|) c(\lambda_i).$$

Based on this upper bound, and the expression for α_i we see that there are again three factors that determine the condition number c_i. The first is again the multiplicative factor $|\sigma|$. The second is the ratio b^Ty_i/b^Ty_1 which can be large if b is nearly orthogonal to y_1 but not to y_i. The third is the denominator $\lambda_i - \lambda_1 + \sigma$ which can vanish if μ_1 is any of the other eigenvalues of A.

An interesting aspect of Algorithm 2 is the explicitness of the numerical complications that we may encounter. The inner product b^Ty_i in the denominator of the expression for γ_i in (3.3) shows that if b is nearly orthogonal to the left invariant subspace Span$\{Q\}$, then we can expect to have numerical difficulties. Moreover, for multiple input problems there are ways of enhancing the stability of the feedback solution by exploiting the extra freedom provided by the inputs [7]. This can easily be implemented for the method of Section II but it is not possible with the progressive method of this section which is defined for single input problems only. However, block generalizations of Wielandt deflation methods can then be used and this will be discussed in Section V.

IV. Conditioning of Partial Pole Assignment

In this section we would like to study how the solution f to the partial pole placement problem varies under perturbations $(\delta A, \delta b)$ to the pair (A, b). The situation of partial pole placement differs from that of standard pole placement because of the constraint that we put on the solution to belong to a particular invariant subspace of A. This will be reflected in our analysis. Unfortunately, it is rather difficult to describe the most general situation and we must consider simplified cases although we suspect that much of the analysis extends. First, we will separate the influence of variations of A and variations of b. In other words, we will consider only perturbations of the form $(0, \delta b)$ or $(\delta A, 0)$. Second, we restrict ourselves to the problem of assigning only one eigenvalue, say the eigenvalue λ_1 of largest real part. For convenience the subscripts on eigenvalues and eigenvectors will be dropped. Then the solution to the problem takes the very simple form

$$f = \frac{\sigma}{(y, b)} y, \tag{4.1}$$

where σ is the shift $\lambda - \mu$. Under perturbations of the form $(0, \delta b)$, i.e., when A is fixed and b is changed into $b + \delta b$, the new solution is $f + \delta f$ where δf is given by

$$\delta f = -\frac{\sigma}{(y, b)} y \left[\frac{(y, \delta b)/(y, b)}{1 + (y, \delta b)/(y, b)} \right].$$

As a result, up to second-order terms, we have the simple formula

$$\frac{\|\delta f\|}{\|f\|} \approx |(y, \delta b)/(y, b)| = \frac{\cos \theta(y, \delta b)}{\cos \theta(y, b)} \frac{\|\delta b\|}{\|b\|}. \tag{4.2}$$

Thus, the sensitivity of the partial pole placement with respect to variations in b can be measured by the condition number

$$c_b(\lambda) = \frac{1}{\cos \theta(b, y)}.$$

Note that the angle between the vector b and the left eigenvector plays once again a crucial role.

When the matrix A is perturbed by δA, then both the eigenvalue λ and the left eigenvector y will change. Let us assume that the eigenvector is perturbed by δy. The size of δy is determined from classical perturbation analysis and can be bounded in terms of the condition number for the eigenvector defined in Section II-B. The perturbed vector f will have the form

$$f + \delta f = \frac{\tilde{\sigma}(y + \delta y)}{(b, y + \delta y)} = \frac{\tilde{\sigma} y}{(b, y)(1 + t)} + \frac{\tilde{\sigma} \delta y}{(b, y)(1 + t)}$$

where we have set $t \equiv (b, \delta y)/(b, y)$ and $\tilde{\sigma} = \mu - (\lambda + \delta\lambda)$. Therefore,

$$\delta f = \frac{-\tilde{\sigma}}{(b, y)(1 + t)} y + \frac{\tilde{\sigma} \delta y}{(b, y)(1 + t)} = \frac{\tilde{\sigma}}{(b, y)(1 + t)} [\delta y - ty].$$

As a result, within first-order approximation,

$$\frac{\|\delta f\|}{\|f\|} \approx \|\delta y - ty\|$$

assuming that $\|y\| = 1$. Interestingly, the vector $\delta y - ty$ involved in the above expression is orthogonal to b and represents the difference between δy and its (oblique) projection onto y and orthogonally to b. If we denote by P the oblique projector onto y orthogonal to b then

$$\frac{\|\delta f\|}{\|f\|} \approx \|(I - P)\delta y\| \leq \|I - P\| \, \|\delta y\|.$$

Moreover, it is easy to prove that the norm of $I - P$ is given by[1]

$$\|I - P\| = \frac{\|b\|}{|(b, y)|} = \frac{1}{\cos \theta(b, y)}.$$

This tells us that the condition number with respect to variations in the matrix A can be defined as the product of the condition number of the eigenvector y and the inverse of the cosine of the angle between b and y,

$$c_A(\lambda) = \frac{c(y)}{\cos \theta(b, y)}.$$

In other words the sensitivity of the partial pole placement with respect to variations in A is the product of the sensitivity of the problem to variations in b and the sensitivity of the eigenvector y.

An interesting conclusion to be drawn from this very simple analysis is that the condition of the partial pole placement problem depends critically on the condition of the underlying partial eigenproblem. In contrast, the condition of a classical full pole placement problem seems independent of that of the eigenproblem of A as is strongly suggested by a numerical example in [9]. In practice, a badly conditioned eigenproblem can be detected by estimating the corresponding condition numbers $c(\lambda_i)$. A related difficulty that remains in nonsymmetric eigenvalue problems is that there are no simple *a posteriori* error bounds for approximations of eigenvalues; see for example [6]. Thus, there is always the possibility of taking the wrong decision in step 2) of Algorithm 2 because an approximation to an eigenvalue is negative while in fact the exact eigenvalue is positive.

V. The Multiple Input Case

As was mentioned earlier in Section I, Algorithm 1 extends readily to multiple input problems. Given an $N \times N$ matrix A with eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_N$, and an $N \times m$ matrix B, the problem is to find an $m \times N$ matrix F such that the matrix $A - BF$ has the eigenvalues $\mu_1, \mu_2, \cdots, \mu_k, \lambda_{k+1}, \lambda_{k+2}, \cdots, \lambda_N$. We must assume that B is of full rank and require that the number of inputs m be no larger than k, the number of eigenvalues to be assigned. As before, the procedure starts with the computation of a partial Schur factorization (2.2) of A^T. Then we will look for a solution of the form

$$F = QG^T$$

where G is a $m \times k$ matrix to be computed. Proceeding as in Section II and denoting by S the $k \times m$ matrix $S = Q^T B$, we find that the wanted matrix G is the solution to the reduced problem

$$\Lambda\{R^T - SG\} = \{\mu_1, \mu_2, \cdots, \mu_k\}.$$

The generalization of Theorem 2.1 is straightforward.

In order to derive an analog of Algorithm 2, we start by pointing out that there are block generalizations of the usual Wielandt deflation technique, that use invariant subspaces of single eigenvectors; see Wilkinson [25, p. 599]. The following description is a slight variation of the one given in [25].

Assume that we have the partial Schur factorization

$$A^T Q_1 = Q_1 C_1 \tag{5.1}$$

where Q_1 is an $N \times m$ unitary matrix and where C_1 is $m \times m$ quasi-upper-triangular with eigenvalues $\lambda_1, \cdots, \lambda_m$. Let H be any $N \times m$ matrix such that

$$H^T Q_1 = C_1 - \Theta \tag{5.2}$$

where Θ is any $m \times m$ matrix having eigenvalues $\lambda_1 - \sigma_1, \lambda_2 - \sigma_2, \cdots, \lambda_m - \sigma_m$. Then one way of defining the block Wielandt deflation of A^T is

$$A_1^T = A^T - Q_1 H^T. \tag{5.3}$$

Clearly, the invariant subspace Span$\{Q_1\}$ is still invariant under A_1^T since

$$A_1^T Q_1 = (A^T - Q_1 H^T) Q_1 = Q_1 C_1 - Q_1 (C_1 - \Theta) = Q_1 \Theta.$$

Thus, the eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_m$ of A have been moved into $\lambda_1 - \sigma_1, \lambda_2 - \sigma_2, \cdots, \lambda_m - \sigma_m$. Moreover, it can be shown [25] that the other eigenvalues are unchanged and there are formulas similar to those of Proposition 2.2, that show in particular that the left eigenvectors associated with the eigenvalues $\lambda_{m+1}, \cdots, \lambda_N$ do not change. The new right eigenvalues can also be derived.

The remaining question is how to select the matrix H. We would like to select H such that the transpose of (5.3) is of the desired form $A - BF$. This dictates that $H = BK$, where K is an $m \times m$ matrix and the condition (5.2) results in

$$K^T = (C_1 - \Theta)[B^T Q_1]^{-1}. \tag{5.4}$$

Concerning the matrix Θ the simplest choice is to take a diagonal matrix with entries $\lambda_i - \sigma_i$ or block diagonal matrix with blocks of size two by two to avoid complex arithmetic when an eigenvalue is complex. We have opted for the Schur factorization in (5.1) but any factorization of the form $A^T Q_1 = Q_1 C_1$ can be used; see [25]. The above describes how to assign m eigenvalues only but it is straightforward to derive a

[1] This can be shown from the simple result that for a matrix of the form $I - uv^T$, in which $(u, v) = 1$ the maximum singular value is the product of the norms of u and v. Incidentally, the norm of P is equal to the norm of $I - P$.

progressive analog of Algorithm 2 which would move m eigenvalues at each iteration.

An important aspect in standard multiple input problems is that the conditioning of the closed-loop system matrix M can be optimized in some sense, in order to produce robust pole assignment. This technique was introduced by Kautsky *et al.* [7] who show how to exploit the extra degrees of freedom provided by larger ranks of B to optimize the condition numbers of eigenvalues of the closed-loop system matrix M. Basically, the idea is to write the conditions that an $N \times N$ matrix X must satisfy in order for it to constitute a matrix of eigenvectors of the closed-loop system matrix M. Then the technique is to find one such matrix that is as close as possible to being orthogonal. Once this matrix of eigenvectors has been selected, the solution F is easily derived. The ideal case is when X can be chosen to be orthogonal which can be achieved in the particular case where B is of rank N. We believe that a similar technique can be developed for partial pole assignment. If we set the goal of minimizing the condition numbers of the displaced eigenvalues only, then much of the theory developed in [7] can be generalized. Given the importance of robust pole assignment we feel that this is worth further study.

VI. Numerical Tests

As an application, we consider the following partial differential equation

$$\frac{\partial u}{\partial t}=\Delta u+\beta\frac{\partial u}{\partial x}+\gamma u+F(x, y, t)$$

on the unit square $\Omega = (0, 1) \times (0, 1)$ with the boundary condition $u(x, y, t) = 0 \ \forall t$, and some initial condition which is of no importance for this test. Here Δ denotes the Laplacian operator $\Delta = \partial^2/\partial x^2 + \partial^2/\partial y^2$. We assume that $F(x, y, t)$ has the form

$$F(x, y, t)=F(x, y)g(t).$$

The unknown u may represent, for example, the concentration of a chemical component that diffuses (Laplacian term) and convects (first-order spatial derivative). The term γu simulates a chemical reaction that results in an increase of the concentration that is proportional to u. Realistic chemical reactions involve two or more chemical components and systems of an equal number of coupled equations. Moreover, the reaction term γu is usually nonlinear. For examples of such models, see [17].

If we discretize the region with n interior points in the x direction and m interior points in the y direction, then the resulting matrix problem is of the form

$$\dot{u}=Au+bg$$

where A is a square and of size $N = nm$.

In this test we take $\beta = 20$, $\gamma = 180$, $n = 20$, and $m = 10$, which results in a matrix A of size 200 whose eigenvalues are all real and negative except two of them which are positive. The vector b which discretizes the function F is filled with random numbers between -1 and 1. The first column of Table I shows the first ten eigenvalues labeled in decreasing order. We used Algorithm 1 to move the two positive eigenvalues into the two new values -0.1 and -0.2, respectively. The resulting ten largest eigenvalues after the transformation are listed in the second column of Table I. The method used for computing the eigenvalues and the eigenspace is a combination of Arnoldi's method and least-squared polynomial acceleration as described in [21]. The stopping criterion when computing each eigenvector is $\|(A - \lambda I)u\| \leq \epsilon$ where u is the normalized eigenvector and $\epsilon = 10^{-08}$. Note that the 1-norm of A is equal to $8h^{-2} = 3200$. Each new Schur vector is obtained by orthogonalizing a new eigenvector against all previous Schur vectors. The projected problem was solved be Petkov's method. To give an idea of the performance, we mention that it took about 5.1 s to solve this 200 × 200 problem by Algorithm 1, on a Vax-8600 and about 12.5 s on a Vax-11-785. This time is dominated by the computation of the three dominant right-most eigenvalues and the corresponding eigenspace (we needed to compute three eigenvalues here assuming no knowledge of the number of positive eigenvalues). Double precision was used throughout (roundoff unit of about $1.38 \cdot 10^{-17}$).

TABLE I
THE 10 RIGHT-MOST EIGENVALUES OF THE SYSTEM BEFORE AND AFTER THE POLE ASSIGNMENT TRANSFORMATION

0.661904775009691D − 01	− 0.883292881569998D − 01
0.770009639638372D − 02	− 0.100002599933231D + 00
− 0.883308887716324D − 01	− 0.170289191634153D + 00
− 0.170288404278961D + 00	− 0.199995312092372D + 00
− 0.219757740327843D + 00	− 0.219759395742943D + 00
− 0.228778768259717D + 00	− 0.228779349544248D + 00
− 0.324809842594408D + 00	− 0.324809936649066D + 00
− 0.383643865022476D + 00	− 0.383644625862267D + 00
− 0.456236500465351D + 00	− 0.456236505002823D + 00
− 0.543074002456214D + 00	− 0.543073999830882D + 00

The feedback vector g for the projected problem was found to be

$$g=(-1.3144.., 0.1405..)^T.$$

The relative norm of the projection of b was $\|\Pi_k b\|_2/\|b\|_2 \approx 0.1050$. This suggests that this problem is not badly conditioned. If b happens to be nearly orthogonal to the Schur vectors the algorithm has serious difficulties. For example, if we take $b = (1, 1, \cdots, 1, 1)^T$ then $\|\Pi b\|_2/\|b\|_2 \approx 3.711 \times 10^{-11}$ which means that b is orthogonal to the first Schur vectors, within the expected accuracy for these Schur vectors. Here, the resulting vector g has a magnitude of 1.6×10^{12}. The resulting transformed matrix has completely erroneous eigenvalues. This behavior is to be expected from our comments of Sections II and IV. In this situation the system is nearly partially incontrollable in the sense defined at the end of Section II. The lesson to be learned from this is that it is important to check beforehand whether the angle between b and the invariant subspace is close to $\pi/2$. If it is the case then the problem is not safely solvable.

VII. Conclusion

The purpose of this note was to show how eigenvalue methods can be put to work to solve large scale pole assignment problems. The advantage of the techniques proposed is that they can handle systems that are so large that standard algorithms can no longer be used. Their limitations are those limitations of the eigenvalue techniques: all we need is to be able to compute the invariant subspace associated with the k unstable modes. Thus, problems with a moderate number of unstable modes can be tackled with no major difficulty. Further research is needed to better understand the underlying theory of partial pole assignment and to produce reliable pole assignment software for very large systems.

Acknowledgment

The author is indebted to B. Nath Datta for arousing his interest in the problem considered in this note and for several stimulating discussions. He has also benefitted from helpful comments by N. Nichols. Finally, he wishes to thank A. Laub for his encouraging comments and for providing him with very useful software.

References

[1] W. E. Arnoldi, "The principle of minimized iteration in the solution of the matrix eigenvalue problem," *Quart. Appl. Math.*, vol. 9, pp. 17–29, 1951.

[2] M. J. Balas, "Trends in large space structure control theory: Fondest dreams, wildest hopes," *IEEE Trans. Automat. Contr.*, vol. AC-2, pp. 522–535, 1982.

[3] F. Chatelin, *Spectral Approximation of Linear Operators.* New York: Academic, 1984.

[4] J. Cullum and R. Willoughby, "A Lanczos procedure for the modal analysis of very large nonsymmetric matrices," in *Proc. 23rd Conf. Decision Contr.*, Las Vegas, NV, 1984.

[5] A. Jennings and W. J. Stewart, A simultaneous iteration algorithm for real matrices," *ACM, Trans. Math. Software,* vol. 7, pp. 184–198, 1981.

[6] W. Kahan, B. N. Parlett, and E. Jiang, "Residual bounds on approximate

eigensystems of nonnormal matrices, *SIAM J. Numer. Anal.*, vol. 19, pp. 470–484, 1982.

[7] J. Kautsky, N. K. Nichols, and P. Van Dooren, "Robust pole assignment in linear state feedback, *Int. J. Contr.*, vol. 41, pp. 1129–1155, 1985.

[8] B. H. Krogh and P. Kokotovic, "Feedback control of overloaded networks," *IEEE Trans. Automat. Contr.*, vol. AC-29, pp. 704–711, 1984.

[9] G. Miminis and C. C. Paige, "An algorithm for pole assignment of time invariant linear systems," *Int. J. Contr.*, vol. 35, pp. 341–354, 1981.

[10] N. K. Nichols, "Robustness in partial pole placement," University of Reading, England, Tech. Rep. NA/1/86, 1986.

[11] C. C. Paige, "Properties of numerical algorithms related to computing controllability," *IEEE Trans. Automat. Contr.*, vol. AC-26, pp. 130–138, 1981.

[12] B. N. Parlett, *The Symmetric Eigenvalue Problem*. Englewood Cliffs, NJ: Prentice-Hall, 1980.

[13] ——, "The software scene in the extraction of eigenvalues from sparse matrices," *SIAM J. Sci. Stat. Comput.*, vol. 5, pp. 590–604, 1984.

[14] B. N. Parlett and Y. Saad, "Complex shift and invert strategies for real matrices," Dep. Comput. Sci., Yale Univ., New Haven, CT, Tech. Rep. YALEU/DCS-RR-424, 1985.

[15] B. N. Parlett, D. R. Taylor, and Z. S. Liu, "A look-ahead Lanczos algorithm for nonsymmetric matrices," *Math. Comput.*, vol. 44, pp. 105–124, 1985.

[16] P. H. Petkov, "A computational algorithm for pole assignment of linear single input systems," Dep. Elec. Eng. Comput. Sci., Kingston Polytech., Tech. Rep. 81-2, 1981.

[17] P. Raschman, M. Kubicek, and M. Maros, "Waves in distributed chemical systems: Experiments and computations," in *New Approaches to Nonlinear Problems in Dynamics—Proceedings of the Asilomar Conference Ground*, Pacific Grove, CA, 1979, P. J. Holmes, Ed. SIAM: The Engineering Foundation, 1980, pp. 271–288.

[18] A. Ruhe and T. Ericsson, "The spectral transformation Lanczos method in the numerical solution of large sparse generalized symmetric eigenvalue problems," *Math. Comp.*, vol. 35, pp. 1251–1268, 1980.

[19] Y. Saad, "Projection methods for solving large sparse eigenvalue problems," in *Matrix Pencils, Proceedings* (Lecture Notes in Math. Series, Number 973), Pitea Havsbad, University of Umea, Sweden, B. Kagstrom and A. Ruhe, Eds. Berlin: Springer-Verlag, 1982, pp. 121–144.

[20] ——, "Chebyshev acceleration techniques for solving nonsymmetric eigenvalue problems," *Math. Comput.*, vol. 42, pp. 567–588, 1984.

[21] ——, "Partial eigensolutions of large nonsymmetric matrices," Yale Univ., New Haven, CT, Tech. Rep. YALEU/DCS/RR-397, 1985.

[22] ——, "Variations on Arnoldi's method for computing eigenelements of large unsymmetric matrices," *Lin. Alg. Appl.*, vol. 34, pp. 269–295, 1980.

[23] G. W. Stewart, "Simultaneous iteration for computing invariant subspaces of nonHermitian matrices," *Numer. Mat.*, vol. 25, pp. 123–136, 1976.

[24] A. Varga, "A Schur method for pole assignment," *IEEE Trans. Automat. Contr.*, vol. AC-26, pp. 517–519, 1981.

[25] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*. Oxford: Clarendon, 1965.

[26] W. M. Wonham, *Linear Multivariable Control: A Geometric Approach*. New York: Springer-Verlag, 1979.

Maximal Stability Robustness for State Equations

IZCHAK LEWKOWICZ AND RAPHAEL SIVAN

***Abstract*—A measure for *stability* robustness of a linear time-invariant finite-dimensional system's state equations is introduced in this note. An upper bound for this measure, based on the characteristic values of the system is derived. It is shown that the set of *optimal* systems, namely, systems for which the stability robustness measure attains the bound, contains the *normal* set, which has been considered as the set of optimal robustness.**

Introduction

In this note we address ourselves to the problem of designing, for *linear, time-invariant, finite-dimensional,* systems, a *state feedback* which achieves *pole location* and *maximal stability robustness.* Previous research that dealt with a similar problem is Kautsky *et al.* [1] and Dickman [2].

The organization of this note is as follows. Section I formulates the problem, Section II contains the theoretical results, and Section III discusses related research.

I. Problem Formulation

Consider the *linear, time-invariant, finite-dimensional* system

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{1}$$

with $x \in \mathbb{C}^n$, the *state* vector and $u \in \mathbb{C}^m$ the *input* vector, $1 \leq m \leq n$. We assume that (A, B) is *completely controllable, B* has *full rank,* and denote by F the *state feedback gain matrix*

$$u(t) = -Fx(t) \tag{2}$$

so that the *closed-loop* system is

$$\dot{x}(t) = (A - BF)x(t). \tag{3}$$

We introduce the set $L = \{\lambda_1, \cdots, \lambda_n\}$ of required closed-loop characteristic values, where $\text{Re}\,\{\lambda_k\} < 0$, $\lambda_k \in \mathbb{C}$, $1 \leq k \leq n$. In the special case where A, B, and F are assumed to be real, L is restricted to be a symmetric set, i.e., $\lambda_k \in L \Rightarrow \lambda_k^* \in L$, where $(\cdot)^*$ stands for *complex conjugate* (or, for matrices, *complex conjugate transpose*). We shall order the characteristic values according to their real parts as follows:

$$\text{Re}\,\{\lambda_1\} \leq \cdots \leq \text{Re}\,\{\lambda_{n-l}\} < \text{Re}\,\{\lambda_{n-l+1}\} = \cdots = \text{Re}\,\{\lambda_n\} = -\lambda_o < 0 \tag{4}$$

indicating that the last l, $1 \leq l \leq n$ characteristic values have identical real parts. Note that λ_o is the minimal distance, in the complex plane, between the set L and the imaginary axis, i.e.,

$$\min_{1 \leq k \leq n} |\text{Re}\,\{\lambda_k\}| = \lambda_o. \tag{5}$$

Since (A, B) is completely controllable, there exist one, or more, matrices F, which achieve a closed-loop pole location at L. We denote by $\mathbf{F}$ the set of all such matrices F

$$\mathbf{F}(A, B, L) \triangleq \{F \in \mathbb{C}^{m \times n} | \lambda_k(A - BF) = \lambda_k,\ 1 \leq k \leq n\} \tag{6}$$

where $\lambda_k(A - BF)$, $1 \leq k \leq n$ denote the n characteristic values of $A - BF$, multiplicity included. Note that if $m = 1$, then $\mathbf{F}$ consists of a unique F.

Given a stable *nominal* closed-loop system

$$\dot{x}(t) = Qx(t),\ Q \triangleq A - BF \tag{7}$$

a measure of *stability robustness* of this system is roughly the maximal "size" of a *perturbation E*, that may be added to Q, so that the *perturbed* system is still stable. In order to rigorously define this measure of stability robustness, we introduce first the set $\mathbf{E}(F)$ of perturbations that do not disrupt stability

$$\mathbf{E}(F) \triangleq \{E \in \mathbb{C}^{n \times n} | \text{Re}\,\{\lambda_k(Q + E)\} < 0,\ 1 \leq k \leq n\}. \tag{8}$$

Definition 1—Stability Robustness Measure: We denote by $\rho(A, B, L, F)$ the *stability robustness measure* of the quadruple (A, B, L, F) for $F \in \mathbf{F}$

$$\rho(A, B, L, F) \triangleq \sup_{\alpha > 0} \{\alpha | \|E\|_2 < \alpha \Rightarrow E \in \mathbf{E}(F)\} \tag{9}$$

where $\|\cdot\|_2$ is the 2-norm. □

Given a system (1), (2), a unitary similarity transformation $U \in \mathbb{C}^{n \times n}$

Manuscript received September 26, 1986; revised December 11, 1986 and July 20, 1987.

The authors are with the Department of Electrical Engineering, Technion—Israel Institute of Technology, Haifa, Israel.

IEEE Log Number 8716992.

0018-9286/88/0300-0297$01.00 © 1988 IEEE