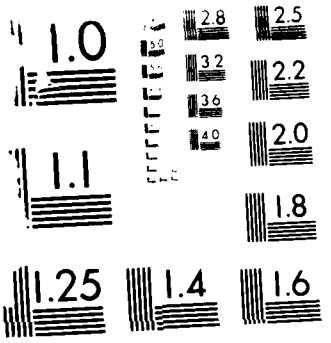
AD-A165 997 A PROJECTION METHOD FOR PARTIAL POLE ASSIGNMENT IN LINEAR STATE FEEDBACK(U) YALE UNIV NEW HAVEN CT DEPT OF COMPUTER SCIENCE Y SAAD JAN 86 YALEU/DCS/RR-449 1/1

UNCLASSIFIED N00014-82-K-0184 F/G 12/2 NL



MICROCOPY RESOLUTION TEST CHART

A Projection Method for Partial Pole Assignment in Linear State Feedback

Youcef Saad

Research Report YALEU/DCS/RR-449

January 1986

YALE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

Abstract. A projection method is proposed for the partial pole placement in linear control systems. The procedure is of interest in the common situation where the system is very large and only a few of its poles must be assigned. It 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. We also present an equivalent version of this method, which can be regarded as a variant of the Wielandt deflation technique used in eigenvalue methods.

A Projection Method for Partial Pole Assignment in Linear State Feedback

Youcef Saad

Research Report YALEU/DCS/RR-449

January 1986

This work was supported in part by the Office of Naval Research under contract N00014-82-K-0184, in part by the Department of Energy under contract DE-AC02-81ER0996 and in part by the Army Research Office under contract DAAG-29-83-0177.

1. Introduction

This paper 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 x N matrix, b is a vector, and u is a scalar function of t. For simplicity, we restrict ourselves to the single input case although our results can readily be extended to multiple input control systems. 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 simple 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 desired eigenvalues. This 'state feedback' technique constitutes one of the most popular ways of modifying dynamical behaviors of time invariant linear control systems [18].

Several methods to solve this problem are available for the case when the matrix A is small, the one generally preferred being an analogue of the QR method developed separately by Miminis and Paige [7] and Petkov [11].

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 [1] and in the control of electrical networks [6]. 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 standard 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 goal is then to place only those unstable eigenvalues in the left half plane and leave the others unchanged. In the next section we describe a simple algorithm to achieve this goal, which is based on invariant subspace techniques.

2. An algorithm for partial pole assignment by state feedback

2.1. The projection process

Let A be an N x N real nonsymmetric matrix whose eigenvalues are

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

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 matrix

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

has the given eigenvalues

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

In other words we would like to assign the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_k$ of A into $\mu_1, \mu_2, \ldots, \mu_k$, 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 $\{\lambda_1, \lambda_2, \ldots, \lambda_k\}$ and $\{\mu_1, \mu_2, \ldots, \mu_k\}$ are symmetric with respect to the real axis, i.e., if a complex value is in one of the set then so is its conjugate. Moreover, if a multiple eigenvalue belongs to the set $\{\lambda_1, \lambda_2, \ldots, \lambda_k\}$ 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 $\lambda_1, \lambda_2, \ldots, \lambda_k$. Let us assume that we have computed the partial Schur factorization for A^T:

$$A^T Q = QR, \tag{2.2}$$

where Q is an N x k matrix whose columns form an orthonormal basis of the left invariant subspace associated with $\lambda_i, i = 1, \ldots k$ and R is a k x k upper quasi-triangular matrix. We will seek a solution f in the form

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

Consider the matrix $B^T Q$:

$$B^T Q = \left[A^T - fb^T\right] Q = QR - Qsb^T Q \equiv Q\left[R - sr^T\right]$$

which can be rewritten as

$$B^T Q = Q\left[R^T - rs^T\right]^T \equiv QC_k^T. \tag{2.4}$$

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

Therefore, let s be chosen so that

$$\Lambda\ \{C_k\} \equiv \Lambda\ \{R^T - rs^T\} = \{\mu_i,\ i = 1, \ldots k\}, \tag{2.5}$$

where $\Lambda(X)$ denotes the spectrum of the matrix X. 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 s transforms the eigenvalues $\lambda_i, i = 1, ..k$ of A into $\mu_i, i = 1, \ldots k$. 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 s be determined so as to solve the projected k x k problem (2.5) and let f be given by (2.3). Then the matrix $B = A - bf^T$ has the eigenvalues $\mu_1, \mu_2, \ldots, \mu_k$, $\lambda_{k+1} \ldots . \lambda_N$.*

Proof. Let $W = [w_1, w_2, \ldots w_{N-k}]$ be the remaining Schur vectors of A^T associated with the remaining eigenvalues $\lambda_i, i>k$. In other words, W is an othogonal basis of the orthogonal complement of $span\{Q\}$ and is such that the matrix $X \equiv [Q, W]$ is unitary. Using

(2.4) we have.

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

Using (2.3) we get.

$$Q^T B^T W = Q^T (A^T - Qsb^T) W \equiv Q^T A^T W + E^T;$$

$$W^T B^T W = W^T (A^T - Qsb^T) W = W^T A^T W.$$

Hence.

$$X^T B X = \begin{pmatrix} C_k & O \\ W^T AQ + E & W^T AW \end{pmatrix}. \tag{2.6}$$

The proof follows immediatly.

∎

A numerical procedure to solve the partial pole assignment problem is therefore as follows.

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 $r = Q^T b$ and solve the projected inverse eigenvalue problem, i.e., find s such that $R^T - rs^T$ has the eigenvalues $\mu_1, \mu_2, \ldots \mu_k$.
3. Form the feedback solution vector $f = Qs$.

The cost of the above process is dominated by the first step which computes a k-dimensional invariant subspace associated with the k eigenvalues to be assigned. There are various methods for this computation, and the reader is referred, for example, to the recent report [14]. Note that we need a method that computes the eigenvalues with (algebraically) largest real parts. This excludes the subspace iteration method which delivers the eigenvalues of largest modulus. Here one can use either a Chebyshev (or least-squares polynomial) acceleration technique combined with Arnoldi's method [14, 13], or a Lanczos type algorithm [10, 3]. If the matrix is banded a safer alternative is to combine Arnoldi with a shift and invert strategy [9]. For time-discrete problems, we need to compute

eigenvalues with largest modulii. The same techniques as above can also be used but we can now add the subspace iteration method [4, 15], which is known to be a much slower method. Although the numerical methods for computing eigenvalues/eigenvectors of large nonsymmetric matrices are not as well established as those for symmetric problems, the above few alternatives will solve the problem more or less efficiently, depending on its conditioning. Large nonsymmetric eigenvalue problems are inheritently much more complex that symmetric ones. Thus, there are instances where an eigenvalue is so poorly conditioned that any iterative procedure will encounter serious difficulties in approximating it. In the symmetric case all eigenvalues are well conditioned, i.e., their condition number is one. Note that there are two condition numbers associated with the computation of an eigenpair, one for the eigenvectors (from which the Schur vectors are derived) and the other with the eigenvalues, see Chatelin [2] for details.

For the above algorithm to be practically feasible the number k of unstable modes must be relatively small. An acknowledged weakness of the above 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. If A has only real eigenvalues, a technique based on the inertia theorem can be employed to determine the number of unstable eigenvalues [17]. However, this has limited applicability as it requires the factorization of the matrix A and, more important, it assumes that A has a real spectrum. 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. We should mention that in contrast with the Schur method presented by Varga [16], the above algorithm requires only a partial Schur factorization.

2.2. Existence of a solution

We now examine the question of the existence of a solution. By looking at the proof of Theorem 2.1, it appears clearly that for a solution to exist it is necessary and sufficient that *a solution to the projected problem exists.* Thus, as a conseqeunce of a well known

result a solution exists *for any set* $\mu_1, \ldots \mu_k$ if and only if the system (R^T, r) is controllable [18]. We will refer to this as the partial controllability condition for (A, b) in the subspace *span* $\{Q\}$. As is well known, (R^T, r) is controllable iff $rank\{r, R^T r, \ldots (R^T)^{k-1} r\} = k$.

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 we have

$$\Pi_k A \Pi_k = \Pi_k A. \tag{2.7}$$

The reason for the above relation is that the subspace $span\{Q\}$ is invariant under A^T and hence $\Pi_k A^T \Pi_k = A^T \Pi_k$, which yields the result by transposition. Another simple proof is to explicitly use the matrix representation of Π_k. We should point out that we also have

$$\Pi_k A^j \Pi_k = \Pi_k A^j, \quad \forall j \tag{2.8}$$

since $span\{Q\}$ is also invariant under $(A^T)^j$. We are now ready to formulate the existence condition in several equivalent ways.

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

(i) $rank\ \{r, R^T r, \ldots, (R^T)^{k-1} r\} = k;$

(ii) $rank\ \{\Pi_k b, \Pi_k A \Pi_k b, \ldots, (\Pi_k A \Pi_k)^{k-1} b\} = k;$

(iii) $rank\ \{\Pi_k b, \Pi_k A \Pi_k b. \ldots, \Pi_k A^{k-1} \Pi_k b\} = k;$

(iv) $dim\ \left[\Pi_k\ span\ \{b_k, Ab_k, \ldots, A^{k-1} b_k\}\right] = k, \quad with \quad b_k = \Pi_k b;$

(v) $dim\ \left[\Pi_k\ span\ \{b, Ab, \ldots, A^{k-1} b\}\right] = k.$

Proof. The formulation (ii) is equivalent to (i) since the vectors in the system of (ii) are obtained from those of (i) by multiplying them by the matrix Q. The other formulations are easy to derive by application of (2.7) and (2.8).

∎

Of the above formulations (iv) and (v) are the most interesting. The condition (v) 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* $\{b, Ab \ldots A^{k-1}b\}$ *is of full rank and contains no vector orthogonal to span*$\{Q\}$, or equivalently iff this Krylov subspace is of full rank and its intersection with the orthogonal complement of the invariant subspace is reduced to $\{0\}$.

A consequence of (iv) is that if the system of vectors $\{b_k, Ab_k \ldots A^{k-1}b_k\}$ is of rank less than k then (A, b) is not partially controllable. In particular, it is clear that when $b_k = \Pi_k b = 0$ the system is not partially controllable. What this means is that the size of $\|\Pi_k b\|_2$ as compared to the size of $\|b\|_2$ can be a good first measure of how controllability has deteriorated by restricting the problem into the invariant subspace. If b is nearly orthogonal to $span(Q)$ then $\|\Pi_k b\|_2/\|b\|_2 << 1$ and the problem is badly conditioned. This will be futher discussed in Section 3 and in the numerical experiments section.

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 [18]. Our context is limited by the fact that we look for a solution of a particular form, namely the form (2.3).

3. Relation with deflation methods

A well-known technique in eigenvalue methods is the so-called Wielandt deflation, see Wilkinson [17] pp. 596-599. Suppose that we have computed the eigenvalue λ_1 of largest modulus and its corresponding eigenvector q_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, q_2. In contrast with other deflation techniques, Wielandt's deflation requires only the knowledge of the right eigenvector. The

deflated matrix is of the form

$$A - \frac{\sigma}{b^T q_1} q_1 b^T. \tag{3.1}$$

where b is an arbitrary vector not orthogonal to q_1 and σ is an appropriate shift. It can be shown that the eigenvalues of B are the same as those of A except for the eigenvalue λ_1 which is transformed into the eigenvalue $\lambda_1 - \sigma$. see [17].

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 $f = s_1 q_1$ where here s_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

$$A^T - s_1 q_1 b^T.$$

To shift the eigenvalue λ_1 to μ_1 we must take $s_1 = (\lambda_1 - \mu_1)/b^T q_1$.

This deflation technique has the property of changing the eigenvalue λ_1 into μ_1 while leaving the others unchanged. It is also known that 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 in general. An exception takes place when the vector b happens to be exactly the left eigenvector of A^T associated with λ_1. In eigenvalue methods the corresponding deflation technique is then a special case of Wielandt's deflation called Hotelling's deflation. This tells us that the sensitivity of the closed-loop system will change for a general vector b and that 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). Note that a comparison of the expressions (2.6) and the analoguous expression for A:

$$X^T A X = \begin{pmatrix} R^T & O \\ W^T A Q & W^T A W \end{pmatrix},$$

indicates that in some sense the variation of the sensitivity can be measured by the norm of $E = W^T b s^T Q$, which vanishes when b is in the left invariant subspace of A^T, or equivalently in the right invariant subspace of A, associated with the first k eigenvalues We do not pursue

this question on the sensitivity of the modified matrix here but we mention that a detailed analysis is currently being developed by Nichols [8].

The similarity with deflation techniques suggests a different algorithm which is based on a more progressive process similar to the progressive Schur-Wielandt deflation procedure used in [14] in the context of eigenvalue calculations. The principle of this approach is to construct f in a progressive way by adding to it multiples of newly computed Schur vectors of A^T which are associated with unstable eigenvalues. The reason why this is possible is that the left Schur vectors of A *that have not yet been deflated* are unchanged. For example, at the beginning of the process we 'compute' $A_1 = A - s_1 b q_1^T$ where s_1 is chosen so as to shift the eigenvalue λ_1 into μ_1. The key observation is that the left Schur vector q_2 of A_1 is the same as the left Schur vector of A associated with the eigenvalue λ_2 as was noted above. We can now get the rightmost eigenvalue of A_1 and the corresponding left eigenvector which is orthogonalized against q_1 to produce q_2. Then $A_2 = A_1 - s_2 b q_2^T$ is constructed, where s_2 is again appropriately chosen, and the process is carried on until exhaustion of all unstable eigenvalues. For simplicity, we describe the method only for the case where the eigenvalues are real.

Algorithm 2: Partial Pole Assignment by Successive Deflations

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

Iterate:

1. Compute the eigenvalue λ_i with largest real part of A_i together with its left Schur vector q_i, with $\|q_i\|_2 = 1$.
2. If $Re(\lambda_i) < 0$ then exit [No more unstable eigenvalues]
3. Else compute
$$f_i := f_{i-1} + s_i q_i, \quad \text{where} \quad s_i = \frac{\lambda_i - \mu_i}{b^T q_i} \tag{3.2}$$
and define $A_i = A - b f_i^T = A_{i-1} - s_i b q_i^T$.
4. If $i \leq k_{max}$ then set $i := i + 1$ and go to 1. else exit [too many eigenvalues to assign].

It is important to observe that the matrix A_i is never computed explicitly in step 3, since all that is required by the methods that compute eigenvalues and eigenvectors, are matrix by vector multiplications $x \leftarrow A_i x$. These operations can be performed by storing A and the vectors b and f_i.

In order to avoid complex arithmetic when a computed eigenvalue is complex we can proceed as follows. In step 1 we obtain *two* Schur vectors, instead of one, which are the results of orthogonalizing the real part and imaginary part of the pair of conjugate eigenvectors, against all previous Schur vectors. Then in Step 3 we can compute directly $f_{i+1} = f_{i-1} + s_{i-1}q_{i-1} + s_i q_i$ by solving a 2 x 2 inverse eigenvalue problem to obtain s_{i-1}, s_i.

One advantage of this algorithm over Algorithm 1 is that there is no need to solve a k x k inverse eigenvalue problem. Note, however, that since k is small as compared with the dimension N of the problem, the cost of solving the projected pole assignment problem is negligible. A disadvantage of the new algorithm is that if we encounter numerical difficulties with some eigenvalue λ_i then the following matrices $A_i, A_{i+1}, A_{i+2}, \ldots$ may be badly perturbed or may lead to badly conditioned eigenvalues that will slow down the computation of the eigenvalues and eigenvectors in Step 1 of the algorithm.

An interesting aspect of the above algorithm is the explicitness of the numerical complications that we may encounter. The inner product $b^T q_i$ in the denominator of the expression for s_i in (3.2) shows that if b is nearly orthogonal to any of the first Schur vectors 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 expolitng the extra freedom provided by the inputs [5]. This can easily be implemented for the method of Section 2 but it is not possible with the progressive method of this section which is defined for single input problems only. We should point out, however, that block-Wielandt deflation methods are perfectly defined, see [17] p. 599, and this can be used to generalize the above algorithm to multiple input case with the possible benefit of better stability.

4. 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 $F(x, y)$ 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 (Laplacien 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 [12].

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 square and of size $N = nm$.

In the first 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 1 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 10 largest eigenvalues after the transformation are listed in the second column of Table 1. The method used for computing the eigenvalues and the eigenspace is a combination of Arnoldi's method and least squares polynomial acceleration as described in [14]. The stopping criterion when

Original poles	Modified poles
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	-0.219759395742843D+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

Table 1: The 10 rightmost eigenvalues of the system before and after the pole assignment transformation.

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 by Petkov's method. To give an idea of the performance, we mention that it took about 5.1 seconds to solve this 200 x 200 problem by Algorithm 1. on a Vax-8600 and about 12.5 seconds on a Vax-11-785. This time is dominted by the computation of the three dominant righmost 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 (round-off unit of about 1.38 x 10^{-17}).

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

$$s = (-1.3144...0.1405..)^T.$$

The relative norm of the projection of b was $\|\Pi_k b\|_2/\|b\|_2 \approx 0.1050$. This means that this problem is relatively well conditioned. If b happens to be nearly orthogonal to the Schur vectors the conditioning of the problem can be disastrous. For example, if we take $b = (1, 1, ...1, 1)^T$ then $\|\Pi b\|_2/\|b\|_2 = 3.711$ x 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 s has a magnitude of 1.6 x 10^{12}. The resulting transformed matrix has

completely erroneous eigenvalues. This behavior is to be expected from our comments of Section 2 and Section 3. In this situation the system is nearly partially incontrollable in the sense defined at the end of Section 2. 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. Another important point is that this aspect may be substantially improved in the case of multiple inputs by using a method in the lines of the one introduced by Kautsky et al. [5].

5. Conclusion

The purpose of this paper was to show how eigenvalue methods can be put to work to solve large scale control problems. The advantage of the technique proposed is that it can handle pole assignment problems that are so large that standard algorithms are no longer applicable. Its 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. We feel that further research is needed to produce reliable pole assignment software for very large systems and to better understand the underlying theory of partial pole assignment.

Acknowledgements. I am indebted to Biswa Nath Datta for arousing my interest in the problem considered in this paper and for several stimulating discussions. I have also benefited from helpful comments by Nancy Nichols. Finally, I wish to thank Alan Laub for his encouraging comments and for providing me with very useful software.

References

[1] M.J. Balas, *Trends in large space structure control theory: fondest dreams, wildest hopes*, IEEE Trans. Aut. Contr., AC-2 (1982), pp. 522–535.

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

[3] J. Cullum and R. Willoughby, A Lanczos procedure for the modal analysis of very large nonsymmetric matrices, *Proceedings of the 23rd Conference on Decision and Control, Las Vegas*, 1984.

[4] A. Jennings and W.J. Stewart, *A simultaneous iteration algorithm for real matrices*, ACM. Trans. of Math. Software, 7 (1981), pp. 184–198.

[5] J. Kautsky, N.K. Nichols, P. Van Doren, *Robust pole assignment in linear state feedback*, Int. J. Contr., 41 (1985), pp. 1129–1155.

[6] B.H. Krogh, P. Kokotovic, *Feedback Control of overloaded networks*, IEEE Trans. Aut. Cont., AC-29 (1984), pp. 704–711.

[7] G. Miminis, C.C. Paige, *An algorithm for pole assignment of time invariant linear systems*, Int. J. Control, 35 (1981), pp. 341–354.

[8] N.K. Nichols, *Robustness in Partial Pole Placement*, Technical Report NA/1/86/, University of Reading, England, 1986.

[9] B.N. Parlett and Y. Saad, *Complex Shift and Invert Strategies for Real Matrices*, Technical Report YALEU/ DCS-RR-424, Yale University, Computer Science Dept., 1985.

[10] B.N. Parlett, D. R. Taylor, Z.S. Liu, The look ahead Lanczos algorithm for large nonsymmetric eigenproblems, J.L. Lions and R. Glowinski ed., *Proceedings of the 6-th International Conference on Computing Methods in Engineering and Applied Sciences, Versailles, France, Dec. 12-16 1984*, INRIA, North-Holland, 1985.

[11] P. H. Petkov, *A computational algorithm for pole assignment of linear single input systems*, Technical Report 81-2, EECS, Kingston Polytechnic, 1981.

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

[13] Y. Saad, *Least squares polynomials in the complex plane with applications to solving sparse nonsymmetric matrix problems*, Technical Report 276, Yale University, Computer Science Dept., 1983.

[14] ————, *Partial Eigensolutions of Large Nonsymmetric Matrices*, Technical Report YALEU/DCS/RR-397, Yale University, 1985.

[15] G.W. Stewart, *Simultaneous iteration for computing invariant subspaces of non-Hermitian matrices*, Numer. Mat., 25 (1976), pp. 123–136.

[16] A. Varga, *A Schur method for pole assignment*, IEEE Trans. Aut. Contr., AC-26 (1981), pp. 517–519.

[17] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.

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

END

DTIC

4-86