Robust Eigenstructure Assignment in State Feedback Control.

J. Kautsky* and N. K. Nichols†.

Numerical Analysis Rpt. NA/2/83

*School of Mathematical Sciences,
Flinders University,
Bedford Park, S.A.5042,
Australia.

†Department of Mathematics,
University of Reading,
Whiteknights,
Reading, Berks. RG6 2AX.
U.K.

Abstract

Numerical methods are described for determining robust, or well-conditioned, solutions to the problem of pole assignment by state feedback. The solutions obtained are such that the sensitivity of the assigned poles to perturbations in the system and gain matrices is minimized. It is shown that for these solutions, bounds on the norm of the feedback matrix and on the transient response are also minimized. A measure is derived which indicates the optimal conditioning that may be expected for a particular system with a given set of closed loop poles, and hence the suitability of the given poles for assignment.

Key words: automatic control, multivariable system, pole assignment, state feedback, numerical algorithm, inverse eigenvalue problem.

Acknowledgements

The methods described here were developed in collaboration with P. Van Dooren (Philips Research Laboratory, Brussels) and L. Fletcher (Salford University). Much of the implementation was completed at Stanford University Computer Science Dept. The authors would like to thank Professor G. H. Golub, Flinders University, Reading University, Philips Research Laboratory, Salford University and the British Council for supporting this co-operative research.

Contents

1. Introduction 1

2. The state feedback pole assignment problem 2

3. Numerical algorithms for robust eigenstructure assignment 8

3.1 Step A/O 9

3.2 Step F 11

3.3 Step X 12

3.4 Implementation 23

4. Applications 24

4.1 Numerical Examples 24

4.2 Discussion of numerical results 40

5. Conclusions 43

References 44

Appendix I - Solutions at convergence: sensitivities 46

Appendix II - Solutions at convergence: feedback and system matrices 55

Appendix III - Improved methods for Steps A/O and F 71

Appendix IV - Modifications for complex eigenvalues 74

1.0 Introduction

The state feedback pole assignment problem in control system design is essentially an inverse eigenvalue problem. The solution is, in general, under-determined, with many degrees of freedom. A desirable property of any system design is that the poles should be insensitive to perturbations in the coefficient matrices of the system equations. This criterion may be used to restrict the degrees of freedom in the assignment problem, and to produce a well conditioned or robust solution to the inverse eigenproblem.

A number of constructive methods for pole assignment by state-feedback are described in the literature [1], [5]-[6], [10]-[13], [15]-[16], [19], [22], [24]. For the single-input case, only one solution to the pole assignment problem may exist, and a numerically stable technique for computing the required feedback is available [13]. For the multi-input problem, however, the procedures which have so far been developed do not generally lead to robust solutions of the problem, and frequently they are computationally unstable.

In this paper we describe four algorithms for computing robust solutions to the multi-input, multivariable state-feedback pole assignment problem. Two of the methods are complementary. In all cases the feedback matrix is obtained by assigning linearly independent eigenvectors corresponding to the required eigenvalues (or poles),such that the matrix of eigenvectors is as well-conditioned as possible [23]. The assigned poles are then as insensitive to perturbations as possible and the resulting feedback matrix is as reasonably bounded as may be expected, given the original system.

In the next section the pole assignment problem is defined in detail, and theoretical considerations are discussed. In section 3 we describe the numerical algorithms. Applications and numerical results are presented in section 4, and concluding remarks follow in section 5.

2.0 The state-feedback pole assignment problem

We consider the completely controllable, time-invariant, linear, multivariable system with dynamic state equation

$$\dot{\underline{x}} = A\underline{x} + B\underline{u} \ , \tag{1}$$

where $\underline{x}$, $\underline{u}$ are n- and m- dimensional vectors, respectively, and A, B are real, constant matrices of compatible orders. Matrix B is assumed to be of full rank. The behaviour of system (1) is governed by the poles of the system, that is, by the eigenvalues of matrix A. It is often desirable to modify the poles of the system in order to obtain certain properties, such as stability. This may be achieved by using a state-feedback control

$$\underline{u} = F\underline{x} + \underline{v},$$

where F, called the feedback or gain matrix, is chosen such that the modified dynamic system

$$\dot{\underline{x}} = (A+BF)\ \underline{x} + B\underline{v} \ , \tag{2}$$

now with input $\underline{v}$, has the desired poles.

The state-feedback pole assignment problem for system (1) is formulated precisely as follows.

Problem 1. Given real matrices(A,B), of orders (nxn, nxm) respectively, and a set of n complex numbers, $\Delta = \{\lambda_1, \lambda_2, .. \lambda_n\}$, closed under complex conjugation, find a real mxn matrix F such that the eigenvalues of A + BF are λ_j, j = 1,2,..n.

Conditions for the existence of solutions to Problem 1 are well-known and the following theorem is well-established [24].

Theorem 1. A solution F to Problem 1 exists for every set Δ of self-conjugate complex numbers if and only if the pair (A,B) is completely controllable, that is, if and only if:

$$\underline{s}^T A = \mu \underline{s}^T \quad \text{and} \quad \underline{s}^T B = \underline{0} \iff \underline{s}^T = \underline{0} . \qquad \blacksquare$$

Indeed, if (A,B) is not controllable, i.e. there exists $\underline{s}^T \neq 0$ such that $\underline{s}^T A = \mu \underline{s}^T$ and $\underline{s}^T B = 0$, then $\underline{s}^T (A+BF) = \mu \underline{s}^T$ for all F. Thus μ is an eigenvalue of $A+BF$ for all F and must belong to any set Δ of poles to be assigned. The pole μ is said to be uncontrollable, and it cannot be modified by any feedback control.

In the single-input case $(m=1)$ the solution to Problem 1, when it exists, can be shown to be unique [12]. This result does not hold in general. In the case $m=n$, solutions always exist, since rank $(B) = n$ then implies that the left null space of B contains only the trivial solution, and the pair (A,B) is always completely controllable.

If we restrict the choice of feedback matrices such that the resulting system matrix $A+BF$ is non-defective (diagonalizable), then Problem 1 is equivalent to

Problem 1': Given (A,B) and Δ (as in Problem 1), find real matrix F such that

$$(A+BF)X = X\Lambda \tag{3}$$

for some non-singular X, where

$$\Lambda = \text{diag}\,\{\lambda_1, \lambda_2, \ldots, \lambda_n\} .$$

We note that system matrices which are defective are necessarily less robust than those which are non-defective, and that this restriction of the problem simply implies that eigenvalues of multiplicity greater than m cannot be assigned.

From (3) it is clear that the columns $\underline{x}_j$, $j = 1,2,\ldots n$, of matrix X are the right eigenvectors of $A+BF$ corresponding to the assigned eigenvalues λ_j. Similarly, the rows $\underline{y}_j^T$, $j = 1,2,\ldots n$, of matrix $Y^T \equiv X^{-1}$ are the corresponding left eigenvectors. It has been shown [23] that the sensitivity of the eigenvalue λ_j to perturbations in the components of A, B and F depends upon the magnitude of $1/c_j$, where

$$c_j \equiv |\underline{y}_j^T \underline{x}_j| \; / \; \|\underline{y}_j\|_2 \, \|\underline{x}_j\|_2 \leq 1.$$

(For real λ_j, c_j is just the cosine of the angle between the right and left eigenvectors corresponding to λ_j.) In the case $m=1$, if F exists, X is uniquely determined (up to scaling), and the sensitivities $1/c_j$ cannot be controlled. In the case $m=n$, X may always be chosen to be orthogonal ($X \equiv I$ suffices) and hence to be such that $c_j = 1$, $\forall_j$. For a general multi-input system ($1 < m < n$) we may control the sensitivities of the assigned poles to a restricted extent by an appropriate choice of the eigenvectors comprising X. We are interested, therefore, not only in eigenvalue assignment, but in assignment of the entire eigenstructure of the system.

Necessary and sufficient conditions for specific eigenstructure assignment are given by the authors for the more general output feedback problem elsewhere [7], [9]. In this report we are concerned with methods for determining an eigenstructure which is as well-conditioned as possible, that is, such that the sensitivities $1/c_j$ of the assigned eigenvalues are as small as possible. A general measure and upper bound for the sensitivities is given by the condition number $\kappa_2(X) \equiv \|X\|_2 \, \|X^{-1}\|_2$ of the matrix X of eigenvectors [23]. We observe that if a matrix V exists such that $FX=V$, then equation (3) may be rearranged in the form

$$AX - X\Lambda = -BV, \tag{4}$$

and we may formulate the robust eigenstructure assignment problem as follows:

Problem 2: Given (A,B) and Δ (as in Problem 1), find real matrices V and X (non-singular) satisfying (4) and minimising $\kappa_2(X)$.

The solution F is then given by $F = VX^{-1}$, and since X is well-conditioned, the matrix F can be computed accurately by a stable numerical process [4], [23].

The matrices V and X satisfy (4) if and only if their respective columns $\underline{v}_j$ and $\underline{x}_j$, $j = 1,2,\ldots n$, are such that

$$\begin{bmatrix} \underline{v}_j \\ \underline{x}_j \end{bmatrix} \in \mathcal{N}([B|A-\lambda_j I]) , \quad \lambda_j \in \Delta ,$$

where $\mathcal{N}(\cdot)$ denotes null space. We observe that $\mathcal{N}([B|A-\lambda_j I])$ is of dimension m, since complete controllability of the pair (A,B) implies $\text{rank}[B|A-\lambda I] = n$, $\forall \lambda \in \mathbb{C}$. If $\tilde{V}_j$, $\tilde{S}_j$ are $m \times m$ and $n \times m$ matrices, respectively, such that

$$\mathcal{R}\left(\begin{bmatrix} \tilde{V}_j \\ \tilde{S}_j \end{bmatrix}\right) = \mathcal{N}([B|A-\lambda_j I]), \quad \lambda_j \in \Delta, \quad j = 1,2,\ldots n,$$

where $\mathcal{R}(\cdot)$ denotes range, then Problem 2 reduces to the problem of selecting vectors $\underline{x}_j$ from the m-dimensional subspaces $S_j \equiv \mathcal{R}(\tilde{S}_j)$, $j = 1,2,\ldots,n$, such that $X = [\underline{x}_1, \underline{x}_2, \ldots \underline{x}_n]$ is as well-conditioned as possible. If the vectors $\underline{x}_j$, $j = 1,2,\ldots n$, can be chosen to form an orthonormal basis for $\mathbb{R}^n$, then matrix X is perfectly conditioned with $\kappa_2(X) = 1$. In essence, then, the general solution of Problem 2 is obtained by selecting vectors $\underline{x}_j \in S_j$ such that $\| \underline{x}_j \|_2 = 1$, $\forall_j$, and $\underline{x}_j$, $j = 1,2,\ldots n$, are as "orthogonal" as possible to each other.

For any given set $\Delta = \{\lambda_1, \lambda_2, \ldots \lambda_n\}$, the minimal conditioning that can be achieved is limited. We let S_j be a matrix such that the columns of S_j form an orthonormal basis for S_j and we let V_j be a corresponding matrix such that

$$R\left(\begin{bmatrix} V_j \\ S_j \end{bmatrix}\right) = R\left(\begin{bmatrix} \tilde{V}_j \\ \tilde{S}_j \end{bmatrix}\right)$$

If we define $S = [S_1, S_2, \ldots S_n]$, then clearly rank $S = n$ is a necessary condition for a non-singular solution X to exist. It can be shown (see [21]), furthermore, that a lower bound on the achievable value of $\kappa_2(X)$ is given by $\kappa_2(S)/\sqrt{n}$, where $\kappa_2(S)$ is defined as the ratio of the largest singular value of S to the smallest [23]. The conditioning of S thus gives a measure of the suitability of the set of poles Λ for assignment. The lower bound given by $\kappa_2(S)/\sqrt{n}$ cannot necessarily be realised, however. In particular, if the augmented matrix composed of any k of the submatrices S_j $(k > m)$ has rank less than k, a non-singular solution X does not exist; and if this augmented matrix has numerical rank less than k, or in other words, if it has a large condition number, then the solution X must, of necessity, be badly conditioned.

Bounds on the components of the feedback matrix F can also be derived in terms of the condition number of X and the given data of the problem. In particular, the authors have shown elsewhere [9] that

$$\| F \|_2 \leq (\| A \|_2 + \max \{\lambda_j\} \cdot \kappa_2(X))/\sigma_m(B), \qquad (5)$$

where $\sigma_m(B)$ is the smallest, non-zero singular value of matrix B. Hence, minimising the conditioning of the assigned eigenstructure also has the effect of minimising a bound on the feedback gains for the given system (in the sense of the ℓ_2 - norm). Furthermore, it is easy to see that the transient response of the modified system (2), where $\underline{x}(0) = \underline{x}_0$, is given by

$$\underline{x}(t) = e^{(A+BF)t}\underline{x}_0 = X e^{\Lambda t} X^{-1}\underline{x}_0 ,$$

and therefore the transient response is bounded by

$$\| \underline{x}(t) \|_2 \leq \kappa_2(X) \cdot \max_j | e^{\lambda_j t}| \cdot \| \underline{x}_0 \|_2 . \qquad (6)$$

Thus, optimising the conditioning of X also minimises a bound on the magnitude of the transient response (in the ℓ_2 -sense) for any given initial condition.

In the next section we describe several stable numerical algorithms for determining well-conditioned, or robust, solutions to the state-feedback pole assignment problem.

3.0 Numerical algorithms for robust eigenstructure assignment

We now consider the practical implementation of the theoretical results discussed in Section 2. We describe four numerical methods for obtaining approximate solutions to Problem 2. The methods all make use of standard library software for obtaining the QR and SVD (singular value) decompositions of matrices and for solving systems of linear equations [4]. The procedures all consist of three basic steps:

Step A/0: Construct a basis, comprised of the columns of the $(m+n)\times m$ matrix $\begin{bmatrix} V_j \\ S_j \end{bmatrix}$, for $N([B|A-\lambda_j I])$, $\lambda_j \epsilon \Delta$, $j = 1,2,\ldots,n$, such that $S_j^T S_j = I$.

Step X: Select vectors $\underline{x}_j \equiv S_j \underline{w}_j \epsilon S_j$ with $\| \underline{x}_j \|_2 = 1$ and set $X = [\underline{x}_1, \underline{x}_2, \ldots \underline{x}_n]$.

Step F: Compute F, the solution of $FX = [V_1\underline{w}_1, V_2\underline{w}_2, \ldots V_n\underline{w}_n]$.

The first and third steps, Steps A/0 and F, are identical for all four methods. The key step is Step X. Here the vectors $\underline{x}_j$, $j = 1,2,\ldots n$, must be chosen to minimise the conditioning of X, or more specifically to minimise the sensitivities, $1/c_j$, of the assigned eigenvalues λ_j. The methods described here do not produce an exact optimal, but produce good approximations to the solution of Problem 2. Methods 1 and 2/3 minimise another measure of conditioning, which generally gives good approximations for the optimal sensitivities, and Method 0 leads asymptotically to a solution which approximates the optimal $\kappa_2(X)$.

We first discuss the two basic steps common to all four methods.

3.1 Step A/O

The basis for each space $N([B|A-\lambda_j I])$ is constructed in two steps, Step A and Step O, each of which may be implemented using either the SVD procedure (Case 1) or the QR procedure (Case 2). In general, the SVD procedure is considerably more expensive than the QR method, but the SVD method gives useful additional information for evaluating the suitability of the set Δ of eigenvalues to be assigned, and for estimating bounds on the optimally attainable feedback matrix.

3.1.1 Step A: We first construct a basis, comprised of the columns of the $(m+n)xn$ matrix $\begin{bmatrix} \tilde{V}_j \\ \tilde{S}_j \end{bmatrix}$, for the space $N[B|A-\lambda_j I])$ for each distinct $\lambda_j \epsilon \Delta$.

Case 1 (SVD): We find the singular value decomposition of $[B|A-\lambda_j I]$ and partition the components in the form:

$$[B|A-\lambda_j I] \equiv \hat{U}_j [\Gamma_j, 0] \left[\begin{array}{c|c} \hat{V}_j & \begin{matrix} \tilde{V}_j \\ \tilde{S}_j \end{matrix} \end{array} \right]^T .$$

Then the column vectors comprising $\begin{bmatrix} \tilde{V}_j \\ \tilde{S}_j \end{bmatrix}$ form a set of m orthonormal vectors which clearly span the m-dimensional null space of $[B|A-\lambda_j I]$.

Case 2 (QR): In this case, we find the QR decomposition of $[B|A-\lambda_j I]^T$ which we partition in the form:

$$[B|A-\lambda_j I]^T = \left[\begin{array}{c|c} \hat{Q}_j & \begin{matrix} \tilde{V}_j \\ \tilde{S}_j \end{matrix} \end{array} \right] \begin{bmatrix} \hat{R}_j \\ 0 \end{bmatrix} .$$

The required orthonormal basis for the null space is again given by the column vectors comprising $\begin{bmatrix} \tilde{V}_j \\ S_j \end{bmatrix}$.

3.1.2 Step 0: We now construct an orthogonal basis, comprised of the columns of matrix S_j, for the space $S_j = R(\tilde{S}_j)$. We require also a basis, given by the columns of $\hat{S}_j$, for the complement of S_j and a matrix V_j corresponding to S_j such that the columns of $\begin{bmatrix} V_j \\ S_j \end{bmatrix}$ are also a basis for

$$R\left(\begin{bmatrix} \tilde{V}_j \\ \tilde{S}_j \end{bmatrix}\right) \equiv N([B|A-\lambda_j I]) \quad .$$

Case 1 (SVD): We determine the singular value decomposition of $\tilde{S}_j$ in the form

$$\tilde{S}_j = [S_j \mid \hat{S}_j] \begin{bmatrix} \Sigma_j \\ 0 \end{bmatrix} Z_j$$

and set $V_j = \tilde{V}_j Z_j^T \Sigma_j^{-1}$. Then matrices S_j, $\hat{S}_j$ and V_j have the required properties.

Case 2 (QR): We form the QR decomposition of $\tilde{S}_j$ partitioned as

$$\tilde{S}_j = [S_j \mid \hat{S}_j] \begin{bmatrix} R_j \\ 0 \end{bmatrix}$$

and set $\tilde{V}_j = V_j R_j^{-1}$. Again, S_j, $\hat{S}_j$ and V_j satisfy the required conditions.

3.2 Step F

Given eigenvectors $\underline{x}_j \in S_j$, expressed as a linear combination of the orthonormal basis, $\underline{x}_j = S_j \underline{w}_j$, we determine F satisfying (3). If we define $\underline{v}_j = V_j \underline{w}_j$ then

$$\begin{bmatrix} \underline{v}_j \\ \underline{x}_j \end{bmatrix} \in N([B|A-\lambda_j I])$$

and $\underline{v}_j$, $\underline{x}_j$ satisfy

$$A\underline{x}_j - \lambda_j \underline{x}_j = -B\underline{v}_j \quad .$$

Hence matrices $X = [\underline{x}_1, \underline{x}_2, \ldots \underline{x}_n]$ and $V = [\underline{v}_1, \underline{v}_2, \ldots \underline{v}_n]$ satisfy (4) and F is given by $F = VX^{-1}$. We construct F by solving the equation $X^T F^T = V^T$ for F^T using a direct LU-decomposition (or, equivalently, Gaussian elimination) method, designed for use with multiple right-hand sides [4].

3.3 Step X

We describe four methods for accomplishing the main step, Step X, in the basic algorithm. Each is based on a different principle, although two of the methods (Methods 2/3) are complementary to each other. For well-posed problems the results of all the methods are, in general, very similar; we have found, however, that in certain cases, one or another of the methods does produce significantly better solutions than the other techniques. The comparative efficiency of the procedures also varies from example to example, although, in general, Methods 2/3 require fewer operations than Method 0, and Method 1 requires more work than any of the other processes. The methods are presented in order of general efficiency. We describe the techniques only for the case where the eigenvectors are all required to be real. Modifications for the case of complex eigenvectors are described in an appendix.

3.3.1 Methods 2/3: The basic objective here is to determine an orthonormal set of vectors $\tilde{\underline{x}}_j$, $j = 1,2,\ldots n$, such that some measure of the distance between the vectors $\tilde{\underline{x}}_j$ and the subspaces S_j is minimized; then the required eigenvectors $\underline{x}_j$, $j = 1,2,\ldots n$, are taken as the normalised projection of $\tilde{\underline{x}}_j$ into S_j. The resulting $\underline{x}_j$ are then approximately orthogonal to each other and the conditioning of $X = [\underline{x}_1, \underline{x}_2 \ldots \underline{x}_n]$ is expected to be reasonably close to unity. The complementary objective is to choose an orthonormal set of vectors $\tilde{\underline{x}}_j$ such that a measure of the distance between the vectors and the complements, $\hat{S}_j$, of subspaces S_j is maximized, and then to project $\tilde{\underline{x}}_j$ into S_j.

The measure of distance used here is essentially a weighted sum of the angles between the vectors and the subspaces. The angle between a vector and a subspace is defined as the minimum angle between the given vector

and any vector in the subspace. The sine of this angle is precisely equal to the distance between the given vector, assuming it is of unit length, and its projection into the subspace.

In particular, for vector $\underset{\sim}{x}_j$, of unit length, and subspace S_j, it can be shown [4] that the projection $\underset{\sim}{p}_j$ of $\underset{\sim}{x}_j$ into S_j, which minimises $\|\underset{\sim}{x}_j - \underset{\sim}{p}\|_2$ over all $\underset{\sim}{p} \in S_j$, is given by $\underset{\sim}{p}_j = S_j S_j^T \underset{\sim}{x}_j$ and the distance between $\underset{\sim}{x}_j$ and its projection is given by

$$\|\underset{\sim}{x}_j - \underset{\sim}{p}_j\|_2 = \|\hat{S}_j^T \underset{\sim}{x}_j\|_2 .$$

By definition, then, the angle α_j between vector $\underset{\sim}{x}_j$ and subspace S_j satisfies $\sin \alpha_j = \|\hat{S}_j^T \underset{\sim}{x}_j\|_2$. Similarly, the angle β_j between vector $\underset{\sim}{x}_j$ and the complement $\hat{S}_j$ satisfies $\sin \beta_j = \|S_j^T \underset{\sim}{x}_j\|_2$. The precise objective of Method 2 is then to choose orthonormal vectors $\underset{\sim}{x}_j$, $j = 1,2,..n$, such that for given weights $\omega_j > 0$, the measure

$$\hat{\nu} = \sum_j \omega_j \|\hat{S}_j^T \underset{\sim}{x}_j\|_2^2$$

is a minimum; and the complementary objective of Method 3 is to choose $\underset{\sim}{x}_j$, $j = 1,2,..n$, such that for given $\omega_j > 0$, the measure

$$\nu = \sum_j \omega_j \|S_j^T \underset{\sim}{x}_j\|_2^2$$

is a maximum.

Since the projection of $\underset{\sim}{x}_j$ into S_j is orthogonal to its projection into $\hat{S}_j$, we have $\beta_j = \pi/2 - \alpha_j$ and, therefore, $\hat{\nu} = K - \nu$, where $K = \sum \omega_j$, for any choice of $\underset{\sim}{x}_j$, $j = 1,2,...n$. For the same choice of weights ω_j, the measures $\hat{\nu}$ and ν are therefore optimised by the same selection of vectors $\underset{\sim}{x}_j$, $j = 1,2,...n$. The amount of work involved in the calculation

of $\hat{\nu}$ is proportional to $n-m$, however, and the work in computing ν is proportional to m. We therefore expect Method 2 to be more efficient than Method 3 in the case where $n-m < m$, and conversely where $n-m > m$.

For Methods 2/3, Step X thus consists of two parts:

Step X(i), in which a set of vectors $\underline{\tilde{x}}_j$ optimizing $\hat{\nu}$ or ν, is determined; and

Step X(ii), in which the normalized projection

$$\underline{x}_j \equiv S_j \underline{w}_j = S_j S_j^T \underline{\tilde{x}}_j / \| S_j^T \underline{\tilde{x}}_j \|_2$$

of $\underline{\tilde{x}}_j$ into S_j is computed for $j = 1,2,\ldots n$.

Step X(i) is accomplished by applying a sequence of rotations (or other unitary transformations) to some initial set of orthonormal vectors in such a way that each rotation reduces the sums of squares of the angles between the vectors and the given subspaces. Explicitly, we start with $\tilde{X}^{(0)} \equiv [\underline{\tilde{x}}_1^{(0)},\ldots\underline{\tilde{x}}_n^{(0)}] = I$, and $H^{(0)} = \hat{S}^T \equiv [\hat{S}_1,\hat{S}_2,..\hat{S}_n]^T$, (or $H^{(0)} = S^T = [S_1,S_2,\ldots S_n]^T$), and generate sequences

$$\tilde{X}^{(k)} = \tilde{X}^{(k-1)} P_k, \quad H^{(k)} = H^{(k-1)} P_k,$$

where $H^{(k)}$ is partitioned in the form

$$H^{(k)} = \begin{bmatrix} \underline{h}_{11}^{(k)} & \underline{h}_{12}^{(k)} & \cdot & \cdot & \cdot & \underline{h}_{1n}^{(k)} \\ \underline{h}_{21}^{(k)} & \underline{h}_{22}^{(k)} & \cdot & \cdot & \cdot & \underline{h}_{2n}^{(k)} \\ \cdot & \cdot & & & & \cdot \\ \cdot & \cdot & & & & \cdot \\ \underline{h}_{n1}^{(k)} & \underline{h}_{n2}^{(k)} & \cdot & \cdot & \cdot & \underline{h}_{nn}^{(k)} \end{bmatrix}$$

and the rotations P_k are chosen to minimise $\Delta\hat{\nu}_k = \hat{\nu}_k - \hat{\nu}_{k-1} < 0$, where

$$\hat{\nu}_k = \sum_i \omega_i \| \underline{h}_{ii}^{(k)} \|_2^2 \equiv \sum_i \omega_i \| \hat{S}_i^T \tilde{\underline{x}}_i^{(k)} \|_2^2$$

(or to maximise $\Delta\nu_k = \nu_k - \nu_{k-1} > 0$,

where

$$\nu_k = \sum_i \omega_i \| \underline{h}_{ii}^{(k)} \|_2^2 \equiv \sum_i \omega_i \| S_i^T \tilde{\underline{x}}_i^{(k)} \|_2^2) .$$

We note that $\hat{\nu}_k$ (or ν_k) can be written as $\text{trace}(H^{(k)T} H^{(k)})$. The rotations are applied in natural order in sweeps through the matrix, each sweep comprising $\frac{1}{2}n(n-1)$ rotations. Each rotation is performed only if the value of $|\Delta\hat{\nu}_k|$ (or $|\Delta\nu_k|$) obtained is greater than a given tolerance, and the sweeps are repeated as long as the measure is improved by more than the tolerance given.

The angle of rotation ϕ to be applied to the pair of columns (i, j) in $H^{(k-1)}$ is determined as follows. We write $\underline{a} = \underline{h}_{ii}^{(k-1)}$, $\underline{b} = \underline{h}_{ij}^{(k-1)}$, $\underline{c} = \underline{h}_{ji}^{(k-1)}$ and $\underline{d} = \underline{h}_{jj}^{(k-1)}$. Then the transformed matrix $H^{(k)}$ has components $\underline{h}_{pq}^{(k)} = \underline{h}_{pq}^{(k-1)}$, $\forall p,q \neq i, j$, and

$$\underline{h}_{ii}^{(k)} \equiv \tilde{\underline{a}} = \underline{a} \cos\phi - \underline{b} \sin\phi, \quad \underline{h}_{jj}^{(k)} \equiv \tilde{\underline{d}} = \underline{c} \sin\phi - \underline{d} \cos\phi,$$

$$\underline{h}_{ij}^{(k)} \equiv \tilde{\underline{b}} = \underline{a} \sin\phi + \underline{b} \cos\phi, \quad \underline{h}_{ji}^{(k)} \equiv \tilde{\underline{c}} = \underline{c} \cos\phi - \underline{d} \sin\phi .$$

It can be shown further that

$$\tilde{\underline{a}}^T\tilde{\underline{a}} = \tfrac{1}{2}(\underline{a}^T\underline{a} + \underline{b}^T\underline{b} + \rho(\underline{a}^T\underline{a} - \underline{b}^T\underline{b}) - 2\sigma\underline{a}^T\underline{b})$$

$$\tilde{\underline{d}}^T\tilde{\underline{d}} = \tfrac{1}{2}(\underline{d}^T\underline{d} + \underline{c}^T\underline{c} + \rho(\underline{d}^T\underline{d} - \underline{c}^T\underline{c}) + 2\sigma\underline{d}^T\underline{c}),$$

where $\rho = \cos 2\phi$ and $\sigma = \sin 2\phi$. For Method 2 we choose angle ϕ to minimise

$$\Delta\hat{\nu}_k \equiv \hat{\nu}_k - \hat{\nu}_{k-1} = \omega_i(\tilde{\underline{a}}^T\tilde{\underline{a}} - \underline{a}^T\underline{a}) + \omega_j(\tilde{\underline{d}}^T\tilde{\underline{d}} - \underline{d}^T\underline{d}) .$$

The optimal choice of ϕ is deduced to be such that $\cos\phi = \text{Re}(\zeta)$, $\sin\phi = \text{Im}(\zeta)$, where $\zeta = ((\rho+i\sigma)/|\rho+i\sigma|)^{\frac{1}{2}}$ and ρ,σ take the values

$$\rho = -(\omega_i(\underline{a}^T\underline{a} - \underline{b}^T\underline{b}) + \omega_j(\underline{d}^T\underline{d} - \underline{c}^T\underline{c}))/\gamma$$

$$\sigma = (\omega_i\underline{a}^T\underline{b} - \omega_j\underline{d}^T\underline{c})/\gamma ,$$

with

$$\gamma = [(\omega_i(\underline{a}^T\underline{a}-\underline{b}^T\underline{b}) + \omega_j(\underline{d}^T\underline{d}-\underline{c}^T\underline{c}))^2 + 4(\omega_i\underline{a}^T\underline{b}-\omega_j\underline{d}^T\underline{c})^2]^{\frac{1}{2}} .$$

For Method 3 we choose angle ϕ to maximise

$$\Delta\nu_k = \nu_k - \nu_{k-1} = \omega_i(\tilde{\underline{a}}^T\tilde{\underline{a}}-\underline{a}^T\underline{a}) + \omega_j(\tilde{\underline{d}}^T\tilde{\underline{d}}-\underline{d}^T\underline{d}) .$$

The optimal ϕ is found to satisfy the same expressions, but with ζ now given by $\zeta = ((-\rho-i\sigma)/|\rho+i\sigma|)^{\frac{1}{2}}$.

Clearly, then, $\hat{\nu}_k \to \hat{\nu}_{opt} \geq 0$ ($\nu_k \to \nu_{opt} \leq K$), and Step X(i) of Methods 2/3 is convergent. Since the initial vectors $\tilde{X}^{(0)} = I$ are orthonormal and the rotations P_k are also orthogonal, we conclude that the columns $\tilde{\underline{x}}_j$ of $\tilde{X} = \prod_k P_k$ are orthonormal and give a minimum to $\hat{\nu}$ (or a maximum to ν) within some tolerance. We note that the matrix X, obtained in Step X(ii) by projecting the columns of $\tilde{X}$ into the required subspaces, does not necessarily have the minimum attainable condition number. It is found, however, that $\kappa_2(X)$ is close to unity if the columns of $\tilde{X}$ are close to the required subspaces.

3.3.2 Method O: The objective here is to choose vectors $\underline{x}_j \in S_j$, $j = 1,2,\ldots n$, such that each vector is as orthogonal as possible to the space spanned by the remaining vectors; that is, such that the angle between vector $\underline{x}_j \in S_j$ and the space $X_j = \langle \underline{x}_i, i \neq j \rangle$ is maximized $\forall j$. Equivalently, for $j = 1,2,\ldots n$, we choose $\underline{x}_j \in S_j$ to minimise the angle between $\underline{x}_j$ and the normalized vector $\underline{y}_j$ orthogonal to the space X_j.

A global solution is not attempted. Instead, an iterative method is used, starting with some set of initial vectors $\underline{x}_j^{(0)} \in S_j$. At each sweep of the iteration, for $j = 1,2,\ldots n$, the vector $\underline{x}_j^{(k)}$ is replaced by a new vector $\underline{x}_j^{(k+1)}$ with maximum angle to the space $X_j^{(k+1)} = R(X_j^{(k+1)})$,

where $X_j^{(k+1)} = [\underline{x}_1^{(k+1)}, \underline{x}_2^{(k+1)}, \ldots \underline{x}_{j-1}^{(k+1)}, \underline{x}_{j+1}^{(k)} \ldots \underline{x}_n^{(k)}]$. The iterations are continued for increasing k as long as $\kappa_2(X^{(k)})$ decreases, where $X^{(k)} = [\underline{x}_1^{(k)}, \underline{x}_2^{(k)}, \ldots \underline{x}_n^{(k)}]$.

The new vector $\underline{x}_j^{(k+1)}$ may be obtained by using either an SVD (Case 1) or QR (Case 2) procedure. In Case 1 an orthonormal basis $\hat{\hat{X}}_j^{(k+1)}$ for $X_j^{(k+1)}$ is first obtained. This may be determined from $X_j^{(k+1)}$ by a Gram-Schmidt procedure, or by using an SVD or QR technique. Then the new vector $\underline{x}_j^{(k+1)}$ is chosen to be the image in S_j of the singular vector which corresponds to the smallest singular value σ_m of $S_j^T \hat{\hat{X}}_j^{(k+1)}$; that is, we form the decomposition

$$S_j^T \hat{\hat{X}}_j^{(k+1)} = \hat{\hat{U}}_j \hat{\hat{\Sigma}}_j \hat{\hat{V}}_j^T$$

and choose $\underline{x}_j^{(k+1)} \equiv S_j \underline{w}_j = S_j \hat{\hat{U}}_j \underline{e}_m$. Then

$$\| \hat{\hat{X}}_j^{(k+1)T} \underline{x}_j^{(k+1)} \|_2 = \| \underline{e}_m^T \hat{\hat{U}}_j^T S_j^T \hat{\hat{X}}_j^{(k+1)} \|_2 = \sigma_m,$$

and the angle between $\underline{x}_j^{(k+1)}$ and $X_j^{(k+1)}$ is maximised over all vectors in S_j; equivalently, the angle between $\underline{x}_j^{(k+1)}$ and the (one-dimensional) complement of $X_j^{(k+1)}$ is minimised.

In Case 2 an equivalent solution is obtained by a simpler and less expensive QR method. We form the decomposition

$$X_j^{(k+1)T} = (\hat{\hat{Q}}_j, \underline{y}_j) \begin{pmatrix} \hat{\hat{R}}_j \\ \underline{0} \end{pmatrix}$$

in order to find $\underline{y}_j$ orthogonal to $X_j^{(k+1)}$. Then the projection of $\underline{y}_j$

into S_j is the vector in S_j which has minimum angle to $\underline{y}_j$; that is, $\underline{x}_j^{(k+1)} \equiv S_j \underline{w}_j = S_j S_j^T \underline{y}_j / \| S_j^T \underline{y}_j \|_2$. (This vector is identical to that determined in Case 1).

In effect, at each stage of the iteration this method chooses the vector in S_j which minimises the sensitivity of the eigenvalue λ_j with respect to the other given vectors. The vector $\underline{y}_j^T$ is just the normalized left eigenvector corresponding to λ_j and therefore the sensitivity $1/c_j = 1/ |\underline{y}_j^T \underline{x}_j^{(k+1)}|$ is minimised by the choice of $\underline{x}_j^{(k+1)}$. However, the conditioning of the remaining eigenvalues λ_i, $i \neq j$ is disturbed when $\underline{x}_j^{(k)}$ is replaced by $\underline{x}_j^{(k+1)}$, and the over-all condition number is not necessarily improved at each stage.

The iteration is continued until the reduction in $\kappa_2(X)$ is less than some positive tolerance. The process does not necessarily converge, and a fixed point of the iteration may or may not be reached before the process is stopped. The solution obtained depends directly on the choice of the initial vectors. Since the procedure requires n QR decompositions at each iteration, it is also much more expensive per sweep than Methods 2/3. However, Method 0 does appear to perform well, starting from an arbitrary initial choice of vectors, and it can lead to more robust solutions than Methods 2/3. In the case where orthogonal solutions X with optimal conditioning $\kappa_2(X) = 1$ exist, it can perform particularly efficiently if appropriate initial vectors are given. The original S.V.D. (Case 1) version of this method is due to P. Van Dooren.

3.3.3 Method 1: The objective of this method is to choose linear combinations of the basis vectors of S_j, $j = 1,2,\ldots n$, such that the resulting linear combinations are as "orthogonal" as possible; that is, we require m-dimensional vectors $\underline{w}_j$ such that $\underline{x}_j = S_j \underline{w}_j$ are as nearly orthogonal as possible for $j = 1,2,\ldots n$. If we write $X = SW$, where W is the block $nm \times n$ "diagonal" matrix $W = \underline{\text{diag}}_m \{\underline{w}_1, \underline{w}_2, \ldots \underline{w}_n\}$, then we

require W such that $X^TX \approx I$. To determine W, we look for a block $nm \times n$ matrix $\tilde{W} = \{\underline{w}_{ij}\}$ (with components of dimension $m \times 1$) such that $\tilde{X} = S\tilde{W}$ is orthogonal, i.e. $\tilde{X}^T\tilde{X} = \tilde{W}S^TS\tilde{W} = I$, and such that $\tilde{W}$ is as close to block diagonal as possible, that is, such that $\| \tilde{W} - \text{diag}_{\underline{m}}(\tilde{W}) \|_F$ is minimized. Then we choose $W = \text{diag}_{\underline{m}}(\tilde{W})$. Here $\text{diag}_{\underline{m}}(\tilde{W})$ denotes the block diagonal matrix with diagonal components $\underline{w}_{jj}$, $j = 1,2,\ldots n$.

The error $\| X^TX-I \|_F$ can then be bounded in terms of $\varepsilon = \| \tilde{W}-W \|_F$. We have $X^TX-I \equiv X^TX-\tilde{X}^T\tilde{X} = (W-\tilde{W})^TS^TSW + \tilde{W}^TS^TS(W-\tilde{W})$. Therefore

$$\| X^TX-I \|_F \leq \varepsilon \| S^TS \|_F (\| W \|_F + \| \tilde{W} \|_F) ,$$

and noting that $$\| S^TS \|_F \leq \| S \|_F^2 = \text{tr}(S^TS) = nm$$

and $$\| \tilde{W} \|_F = \| W+\tilde{W}-W \| \leq \| W \|_F + \varepsilon ,$$

we find $$\| X^TX-I \|_F \leq 2\varepsilon nm \| W \|_F + O(\varepsilon^2) ,$$

where ε is minimal.

An appropriate general form for $\tilde{W}$ is easily obtained by using the singular value decomposition of $S = [S_1, S_2,\ldots S_n]$. We form $S = U(\Sigma,0)\,(V_1,V_2)^T$, and take $\tilde{W} = V_1\Sigma^{-1}P + V_2Z$ where P is any orthogonal matrix and Z is arbitrary. Then $S\tilde{W} = U\Sigma V_1^T\,(V_1\Sigma^{-1}P+V_2Z) = UP = \tilde{U}$, which is clearly orthogonal. The objective of Method 1, then, is to choose P and Z such that

$$\| \tilde{W} - \text{diag}_{\underline{m}}(\tilde{W}) \|_F^2 \equiv \sum_{\substack{i,j \\ i \neq j}} \| \underline{w}_{ij} \|_2^2$$

is minimized.

For a given matrix P, Z is determined as the best least-square solution of a linear system. If we write $V_2^T = [V_{21}, V_{22}, \ldots V_{2n}]$ and $Z = [\underline{z}_1, \underline{z}_2, \ldots \underline{z}_n]$, then Z is the best least-square solution of the equations

$$V_2 Z - \mathrm{diag}_{\underline{m}}(V_2 Z) \equiv \begin{bmatrix} \underline{0} & V_{21}^T \underline{z}_2 & \cdots & V_{21}^T \underline{z}_n \\ V_{22}^T \underline{z}_1 & \underline{0} & \cdots & \vdots \\ \vdots & \vdots & & \vdots \\ V_{2n}^T \underline{z}_1 & V_{2n}^T \underline{z}_2 & \cdots & \underline{0} \end{bmatrix} = K,$$

where $K = \mathrm{diag}_{\underline{m}}(V_1 \Sigma^{-1} P) - V_1 \Sigma^{-1} P$. We observe that we obtain $nm-m$ equations for determining the $nm-n$ dimensional vectors $\underline{z}_j$, for each $j = 1,2,\ldots n$. These equations may be written, in the case $P = I$, in the form

$$V_2^j \underline{z}_j = \underline{k}_{i_j}, \qquad j = 1,2,\ldots n,$$

where $V_2^{jT} = [V_{21}, V_{22}, \ldots V_{2,j-1}, V_{2,j+1}, \ldots V_{2n}]$, $\underline{k}_{i_j}$ is the i_j-th column of $-V_1 \Sigma^{-1}$ with the j^{th} block of m components deleted, and $i_j = j$. For any permutation matrix P, the equations are of the same form, with the integers i_j given by the corresponding permutation of the integers $\{1,2,\ldots n\}$.

The best least-square solutions $\underline{z}_j$ are found using singular value or QR decompositions of the matrices V_2^j, $j = 1,2,\ldots n$. To determine a best choice of P amongst all permutation matrices, we compute the residuals $\underline{r}_j^i \equiv V_2^j \underline{z}_j - \underline{k}_i$, for all $i, j = 1,2,\ldots n$. The residuals are given explicitly in terms of the decompositions. If we define matrix $N^T = \{\| \underline{r}_j^i \|_2^2\}$, then for a given permutation

$$\| \tilde{W}\text{-diag}_{\underline{m}}(\tilde{W}) \|_F = \sum_j \| \underline{r}_j^{i_j} \|_2^2 = \text{trace } (NP) .$$

Hence we simply form matrix N and apply a sequence of interchanges which reduce the trace at each step. The optimal permutation P is then given by the product of the elementary interchanges.

Explicit expressions for $\underline{z}_j$, $\underline{r}_j^i$ and $\underline{w}_j$ are obtained from the singular value decomposition of V_2^j, which we write as

$$V_2^j = U_2^j \begin{pmatrix} E_2^j \\ 0^2 \end{pmatrix} \hat{Z}^{jT} .$$

Then the residuals are given by

$$\underline{r}_j^i = (0, I_{n-m}) U_2^{jT} \underline{k}_i , \qquad \forall\ i,j = 1,2,\ldots n,$$

and for the chosen permutation, the solutions $\underline{z}_j$ of the least-square problem take the form

$$\underline{z}_j = \hat{Z}^j (E_2^j)^{-1} (I_{nm-n}, 0)\, U_2^{jT} \underline{k}_{i_j} , \quad j = 1,2,\ldots n,$$

where the values i_j are determined by the permutation. Finally, the required vectors $\underline{w}_j$ are computed from

$$\underline{w}_j \equiv \tilde{\underline{w}}_{jj} = V_{2,j} \underline{z}_j - \hat{\underline{k}}_{i_j}$$

where $\hat{\underline{k}}_{i_j}$ is the j-th m-dimensional block of the i_j-th column of $-V_1 \Sigma_1^{-1}$. Similar explicit expressions are obtained from the QR decomposition of the matrix V_2^j. Then we have

$$\underline{r}_j^i = \hat{U}_2^{jT}\,\underline{k}_i, \quad \underline{z}_j = R_j^{-1}\,U_2^{jT}\,\underline{k}_{i_j},$$

where

$$V_2^j = \left(U_2^j,\ \hat{U}_2^j\right)\begin{pmatrix} R_j \\ 0^j \end{pmatrix}.$$

Since P is selected only amongst all permutation matrices, rather than from amongst the more general class of all orthogonal matrices, this procedure does not compute the optimal choice possible for $\tilde{W}$. The method could be modified to generalise further the selection of $\tilde{W}$, but at the expense of considerable complication. The method already requires a relatively large number of operations due to the number of matrix decompositions employed, and for problems in which well-conditioned results are expected, it obtains good solutions even in its present restricted form.

3.4 Implementation

The three steps, Step A/O, Step X and Step F, of the numerical methods described in Sections 3.1-3.3 have been implemented using the system MATLAB [14]. This system makes use of standard library routines from the software packages LINPACK[4] and EISPACK[20]. For experimental purposes, we have developed a set of executive files for use with MATLAB to carry out the various steps of the pole assignment procedures. These files have been incorporated in a small package, together with a number of test examples. A detailed description of the package, with listings, and instructions for its use are available in another document [8]. The test examples are given in the next section of this paper, and the results of numerical experiments with the pole assignment methods are reported.

Recently we have developed new, more efficient procedures to replace the techniques used in Steps A/O and F. These procedures are based on theory for eigenstructure assignment derived elsewhere by the authors [7] [9]. Executive files for implementing the new steps with MATLAB have been written, and a brief description of the revised methods is given here in an appendix.

4.0 Applications

The four procedures described in section 3 have been applied to a number of examples collected from the literature. In some cases, the given control system is unstable, and a feedback matrix which stabilizes the system is to be assigned. In other cases, the system is already stable, and the objective of the pole assignment is then to move some of the smaller eigenvalues further into the left half-plane, away from the imaginary axis, and also to improve the conditioning of the system. We first describe each of the test examples, and give the results of numerical experiments. Then a summary and general discussion is presented.

4.1 Numerical Examples

For each example we give the type of system and source of the problem, followed by the order (n, m) of the system, the coefficient matrices A, B, and the poles of the original system, EIG(A).

Example 1 (EX4) TEST (Barnett, [1]).

$n = 3 \qquad m = 2$

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 6 & -11 & 6 \end{bmatrix}$$

$$B^T = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

EIG(A) = 1.0, 2.0, 3.0.

We first assign the eigenvalue set $\Lambda = \{1, 1, 3\}$ as in [1]. The condition of S is $\kappa_2(S) = 8.32$, and therefore we expect to be able to obtain a reasonably well-conditioned solution. After two sweeps with

Method 2, (taking approximately 743 flops) we obtain the solution

$$F = \begin{bmatrix} -1.6053 & 3.0941 & -1.4887 \\ -2.0941 & 4.2907 & -2.1966 \end{bmatrix}.$$

Here the condition of X is $\kappa_2(X) = 7.8098$ and the sensitivities of the assigned eigenvalues are $\{3.92, 1.40, 3.95\}$. With tolerance level 10^{-5}, no change is made in the third sweep and the iteration is halted.

Using two sweeps of Method 0 (Case 2) (approximately 1278 flops) we obtain almost the same solution

$$F = \begin{bmatrix} -1.6073 & 3.0972 & -1.4899 \\ -2.0972 & 4.2955 & -2.1984 \end{bmatrix}$$

with $\kappa_2(X) = 7.7772$. The eigenvalue sensitivities are here $\{3.04, 2.72, 3.95\}$. The iteration stops after two sweeps with tolerance 10^{-5}.

From Method 1 (approximately 3443 flops) we obtain the identical solution F, to 5-figures, as that of Method 0, but with a different set of corresponding eigenvectors. The condition of X is still $\kappa_2(X) = 7.7772$, but the eigenvalue sensitivities are $\{1.00, 3.95, 3.95\}$.

These solutions all compare favourably with the solution

$$F = \begin{bmatrix} -3 & 4 & -1 \\ -3 & 4 & -1 \end{bmatrix}$$

derived in [1], which has very poor conditioning. If errors $\pm$ 0.001 are introduced into the resulting feedback system matrix $A + BF$, perturbations of up to 13.9% occur in the assigned poles. Introducing the same errors into the system matrices obtained by Methods 2, 0 and 1 leads to errors of at most 0.2% in the assigned eigenvalues.

We have also assigned the eigenvalue set $\Delta = \{-0.2, -0.2, -10.0\}$, which produces a stable system. The conditioning of S, $\kappa_2(S) = 3.65$ is again satisfactory. With two sweeps of each method and tolerance level 10^{-5}, very similar results are obtained. For Method 2, $\kappa_2(X) = 3.2827$, $\|F\|_2 = 16.541$ and the sensitivities of the eigenvalues are $\{1.57, 1.43, 1.79\}$. For Method 0, $\kappa_2(X) = 3.2732$, $\|F\|_2 = 16.461$ and the sensitivities are $\{1.34, 1.55, 1.79\}$; and for Method 1 we have again $\kappa_2(X) = 3.2732$, $\|F\|_2 = 16.461$, but with sensitivities $\{1.00, 1.79, 1.79\}$. Method 2 converges in four sweeps with minor improvements on these results and Method 0 halts after two sweeps.

Example 2 (EX1) AIRCRAFT CONTROL (Okada, Kihara, Kobayaski, [18])

$n = 4 \qquad m = 3$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0.000140 & -2.04 & -1.95 & 0.0133 \\ -0.000251 & 1 & -1.32 & -0.0238 \\ -0.561 & 0 & 0.358 & -0.279 \end{bmatrix}$$

$$B^T = \begin{bmatrix} 0 & -5.33 & -0.160 & 0 \\ 0 & 0.00645 & -0.0116 & 0.106 \\ 0 & -0.267 & -0.251 & 0.0862 \end{bmatrix}$$

$EIG(A) = -3.12_{10}-2, \quad -2.46_{10}-1, \quad -1.68 \pm 1.35i.$

We assign the eigenvalue set $\Delta = \{-1, -2, -3, -4\}$. The condition of S is $\kappa_2(S) = 4.9040$ and we may expect to find robust solutions to the problem. After two sweeps with Method 2 (approximately 1503 flops) we obtain a result with $\kappa_2(X) = 3.6103$, $\|F\|_2 = 28.255$ and eigenvalue sensitivities $\{1.94, 1.00, 1.00, 1.94\}$. The corresponding system

matrix A + BF is

$$A + BF = \begin{bmatrix} 0.0000 & 1.0000 & 0.0000 & 0.0000 \\ -4.0000 & -5.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & -2.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & -3.0000 \end{bmatrix},$$

which is as decoupled as it may be for the given matrix B. With tolerance 10^{-5} no changes are made on the third sweep and the iteration is stopped.

With Method 0, in Case 2, after two sweeps (approximately 2830 flops) the solution is such that $\kappa_2(X) = 3.6690$, $\|F\|_2 = 26.578$ and the sensitivities are {1.97, 1.01, 1.00, 1.97}. With Method 0, in Case 1, after two sweeps (approximately 5024 flops) a solution is obtained with $\kappa_2(X) = 3.7361$, $\|F\|_2 = 25.931$, and eigenvalue sensitivities {1.99, 1.02, 1.00, 2.00}. In both cases, after ten sweeps, Method 0 converges with tolerance 10^{-5} to a solution with $\kappa_2(X) = 3.6103$ and the same sensitivities as the solution obtained by Method 2, but with different gain matrices F; in Case 2, $\|F\|_2 = 26.540$ and in Case 1, $\|F\|_2 = 25.543$. The system matrices A + BF obtained by Method 0 are not decoupled, however, as is the result of Method 2.

With Method 1 (taking approximately 19,371 flops) the system matrix obtained is partially decoupled. The conditioning is $\kappa_2(X) = 6.1623$, and $\|F\|_2 = 30.294$; the sensitivities are {3.16, 3.16, 1.00, 1.00} and the system matrix is

$$A + BF = \begin{bmatrix} 0.0000 & 1.0000 & 0.0000 & 0.0000 \\ -2.0000 & -3.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & -3.6280 & 0.48334 \\ 0.0000 & 0.0000 & 0.48334 & -3.3720 \end{bmatrix}.$$

Rounding the gain matrix F obtained by these methods to three figures of accuracy corresponds to introducing maximum absolute errors of $\pm 5_{10}{}^{-4} \max_{i,j} \{(BF)_{ij}\}$ into the system matrices. For robust solutions such perturbations should only cause errors of the same order of magnitude (x n) in the poles of the feedback system. For this example, rounding the gain matrix F obtained by Method 2 introduces a maximum absolute error of $\pm$ 0.004 into the eigenvalues of the system matrix A + BF, or a maximum relative error of 0.13%. Similarly, rounding the matrix F constructed by Method 0 gives maximum errors of $\pm$ 0.003, or 0.17%, in the poles in Case 2, and maximum errors of $\pm$ 0.008, or 0.28%, in Case 1. For Method 1, rounding matrix F gives a maximum absolute error $\pm$.004 and a maximum relative error 0.27%. The maximum percentage error in Method 1 occurs in the eigenvalue of smallest modulus, however, while for Methods 2 and 0 this eigenvalue is only in error to 0.03-0.05%.

Experiments have also been carried out using Method 2 with a weighted measure in order to obtain solutions in which the eigenvalues of smallest modulus have the best conditioning. It is clear that the eigenvalues nearest the imaginary axis should be least sensitive to perturbations in the system, since these are the modes most likely to become unstable. It is found that with weights $\omega_1 = 5$, $\omega_2 = 25$, $\omega_3 = 5$ and $\omega_4 = 1$, a solution is obtained with eigenvalue sensitivities {1.00, 1.00, 13.04, 13.04}. The overall conditioning $\kappa_2(X) = 26.038$ is considerably worse than in the unweighted solution, but $\|F\|_2$ is reduced to $\|F\|_2 = 12.584$ and the conditioning of the first eigenvalue is clearly improved. The system matrix obtained is

$$A + BF = \begin{bmatrix} 0.0000 & 1.0000 & 0.0000 & 0.0000 \\ -12.000 & -7.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & -2.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & -1.0000 \end{bmatrix}$$

which is still decoupled. The errors introduced by rounding matrix F are more uniform, the maximum absolute error being $\pm$ 0.0015 and the maximum relative error being 0.06%. The error introduced into the eigenvalue of least modulus by the rounding procedure is, however, reduced only slightly, but this error is already very small.

Example 3 (EX13) CHEMICAL REACTOR (Munro, [17]).

$n = 4 \qquad m = 2$

$$A = \begin{bmatrix} 1.380 & -0.2077 & 6.715 & -5.676 \\ -0.5814 & -4.290 & 0 & 0.6750 \\ 1.067 & 4.273 & -6.654 & 5.893 \\ 0.0480 & 4.273 & 1.343 & -2.104 \end{bmatrix}$$

$$B^T = \begin{bmatrix} 0 & 5.679 & 1.136 & 1.136 \\ 0 & 0 & -3.146 & 0 \end{bmatrix}$$

$EIG(A) = 1.991, 6.351_{10}-2, -5.057, -8.666.$

This system is unstable and a feedback matrix is required to stabilize the system. We therefore move the two positive real modes into the left-half plane, keeping the original stable modes. We assign the set $\Delta = \{-0.2, -0.5, -5.0566..., -8.6659...\}$. The condition of S is $\kappa_2(S) = 3.761$ and a feedback system with good conditioning is expected. After two sweeps of Method 2 (approximately 3472 flops) a solution with $\kappa_2(X) = 4.5589$, $\|F\|_2 = 1.1553$ and eigenvalue sensitivities {2.36, 1.19, 2.36, 1.11} is obtained. The process converges with tolerance 10^{-5} in six sweeps with slightly improved conditioning: $\kappa_2(X) = 4.5355$ and sensitivities {2.37, 1.07, 2.34, 1.09}, but with slightly increased $\|F\|_2 = 1.1656$.

With Method 0, in Case 2, the best result is obtained from the first sweep with $\kappa_2(X) = 3.5834$, $\|F\|_2 = 1.4117$ and sensitivities {1.87, 1.41, 1.52, 1.72}. On the next sweep the value of $\kappa_2(X)$ increases and if the iteration is allowed to proceed, the conditioning continues to increase. After 60 sweeps a condition number $\kappa_2(X) = 4.1301$ is obtained, and there is no indication of convergence. With Method 0, in Case 1, the best result is obtained after eleven sweeps and is such that $\kappa_2(X) = 3.2811$, $\|F\|_2 = 1.3292$ and the pole sensitivities are {1.69, 1.53, 1.49, 1.75}. Further sweeps increase the conditioning $\kappa_2(X)$ of the solution and the process does not show convergence. We observe also that if the solution obtained by Method 2 is used as a starting approximation for Method 0, the conditioning is made worse by applying this process.

For this example Method 1 also produces a good result (taking approximately 8548 flops). The solution has $\kappa_2(X) = 3.4391$, $\|F\|_2 = 1.5408$ and sensitivities {1.42, 1.84, 1.41, 1.84}.

Examination of the augmented matrix $S = [S_1, S_2, S_3, S_4]$ suggests that the space S_3 is unduly constraining the choice of eigenvectors, and that moving the pole $\lambda_3 = -5.0566...$ may improve the results. Therefore we also assign the set $\Delta = \{-0.2, -0.5, -8.6659..., -8.6659...\}$, with a double pole at the position of the eigenvalue of largest modulus in the original system. The overall condition of S is now $\kappa_2(S) = 3.2934$, giving an improvement.

Method 3 here performs much as before, giving a solution with $\kappa_2(X) = 3.2182$, $\|F\|_2 = 1.3911$ and sensitivities {1.73, 1.13, 1.66, 1.27} after two sweeps (approximately 3056 flops). The procedure converges with tolerance 10^{-5} after five sweeps to a solution with $\kappa_2(X) = 3.2122$, $\|F\|_2 = 1.4039$, and sensitivities {1.75, 1.08, 1.67, 1.27}.

For this set Λ of poles, Method 0, in Case 2, obtains a good result on the first sweep and improves the result on the second sweep (3180 flops), giving a solution with $\kappa_2(X) = 3.1974$, $\|F\|_2 = 1.4035$ and sensitivities {1.75, 1.08, 1.11, 1.73}. With tolerance 10^{-5} no essential changes are made on the third sweep and the process here converges. In Case 1, Method 0 gives a solution with $\kappa_2(X) = 3.2552$, $\|F\|_2 = 1.4013$ and sensitivities {1.50, 1.45, 1.07, 1.78} after two sweeps (4384 flops) and converges with tolerance 10^{-5} after 34 sweeps to a solution with $\kappa_2(X) = 3.1969$, $\|F\|_2 = 1.3970$, and sensitivities {1.74, 1.09, 1.07, 1.75}. If the solution obtained by Method 3 is used as an initial approximation in Method 0, (Case 2), the solution is improved further, and Method 0 converges again to essentially the same result as before.

Method 1, however, fails completely to find a sensible solution for this data. The resulting feedback system matrix has eigenvalue conditioning $\kappa_2(X) = 6.8_{10}+5$ and the eigenvalue sensitivities are $\{1.5_{10}3, 1.6_{10}3, 3.4_{10}5, 3.4_{10}5\}$.

For this example the effect of rounding the gain matrix F, constructed by Methods 3 and 0 (Case 2), to three figures of accuracy is to introduce maximum errors of ± 0.003 into the assigned eigenvalues, or a maximum relative error of about 1% in the eigenvalue of smallest modulus. As would be expected, rounding the solution F obtained by Method 1 gives a feedback system matrix with eigenvalues $-6.58 \pm 10.54i$, $+ 3.648$, $- 8.6642$, which bear no relation to the assigned values.

Example 4 (EX7) NUCLEAR ROCKET ENGINE (Davison and Chow, [3]).

$n = 4 \qquad m = 2$

$$A = \begin{bmatrix} -65.0 & 65.0 & -19.5 & 19.5 \\ 0.1 & 0.1 & 0 & 0 \\ 1.0 & 0 & 0.5 & -1.0 \\ 0 & 0 & 0.4 & -0.4 \end{bmatrix}$$

$$B^T = \begin{bmatrix} 65.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.4 \end{bmatrix}$$

EIG (A) = -64.795, -0.60234 $\pm$ 0.51972i, 0.0.

We first arbitrarily assign the pole set $\Delta = \{-1, -2, -3, -4\}$. For these eigenvalues the conditioning of S is $\kappa_2(S) = 42.506$ which indicates that we cannot obtain a very robust solution to the problem. After two sweeps of Method 3 (approximately 3448 flops) we obtain a solution with $\kappa_2(X) = 156.03$, $\|F\|_2 = 137.74$ and sensitivities {1.33, 78.02, 78.02, 1.46}. With tolerance 10^{-5} the procedure converges in four sweeps to a slightly improved solution with $\kappa_2(X) = 154.79$, $\|F\|_2 = 133.18$ and sensitivities {1.62, 77.40, 77.40, 2.36}.

After two sweeps of Method 0, in Case 2, (approximately 2619 flops) a rather better solution is obtained with $\kappa_2(X) = 54.83$, $\|F\|_2 = 109.05$ and sensitivities {20.09, 20.10, 23.01, 11.65}. The best result obtained by this method is produced in three sweeps, with $\kappa_2(X) = 36.957$, $\|F\|_2 = 64.943$ and sensitivities {18.46, 2.96, 18.48, 2.76}. The procedure does not converge to this solution, however, and if the iteration is allowed to proceed, solutions with increasing condition numbers $\kappa_2(X)$ are produced. The iteration does eventually converge after 26 sweeps to a solution

with $\kappa_2(X) = 39.292$, $\|F\|_2 = 48.884$ and sensitivities {17.88, 8.93, 17.77, 8.41}. In Case 1, with Method 0, a similar pattern is observed, the best result being achieved on the third sweep, with $\kappa_2(X) = 36.904$, $\|F\|_2 = 62.675$ and sensitivities {18.04, 3.35, 18.41, 3.006}. In this case the method converges after seven sweeps to a less well-conditioned result.

A reasonable solution is also constructed by Method 1 (approximately 7170 flops), with $\kappa_2(X) = 48.982$, $\|F\|_\infty = 43.691$ and sensitivities {24.24, 24.31, 7.90, 7.82}. We note that with a different choice of permutation this procedure can give an even better solution with $\kappa_2(X) = 37.088$, $\|F\|_2 = 65.856$ and sensitivities {18.50, 2.77, 18.51, 2.76}.

For this set of arbitrary poles, however, the resulting system matrices are clearly not robust and are sensitive to small perturbations in the data. We therefore consider a more reasonable pole assignment problem where we simply move the smallest eigenvalues away from the imaginary axis. We assign set $\Delta = \{-0.1, -1, -3, -64.593 \ldots\}$. With this data $\kappa_2(S) = 1.7655$. After two sweeps of Method 3 (approximately 3056 flops) a reasonably good solution is obtained with $\kappa_2(X) = 1.4579$, $\|F\|_2 = 131.31$ and sensitivities {1.00, 1.07, 1.00, 1.07}. The procedure converges with tolerance 10^{-5} in seven sweeps to a slightly improved solution with $\kappa_2(X) = 1.4478$, $\|F\|_2 = 122.16$, and sensitivities {1.00, 1.07, 1.00, 1.07}.

With Method 0, in Case 2, after two sweeps (2965 slops) the solution obtained has $\kappa_2(X) = 1.5117$, $\|F\|_2 = 143.26$ and sensitivities {1.00, 1.08, 1.00, 1.08}. After two sweeps (5389 flops) with Method 0, in Case 1, a somewhat better solution is obtained, with $\kappa_2(X) = 1.4706$, $\|F\|_2 = 133.75$ and sensitivities {1.00, 1.07, 1.00, 1.08}. In both cases, Method 0 converges to a solution almost identical to that obtained by Method 3, with the same conditioning $\kappa_2(X)$, the same eigenvalue sensitivities to five figures accuracy, and the same value of $\|F\|_2$ to three figures accuracy.

In Case 2 convergence is reached after 16 sweeps and in Case 1 after 12 sweeps.

With Method 1 (approximately 7687 flops) a reasonably good solution is also obtained, although it is not as well-conditioned as the results of Method 3 or Method 0. The solution has $\kappa_2(X) = 3.5436$, $\|F\|_2 = 151.74$ and sensitivities {1.00, 1.23, 1.82, 1.91}.

If the gain matrix F obtained by these methods is rounded to three figures accuracy, errors of maximum modulus 0.014-0.032 are introduced into the assigned poles. For Method 3 the maximum relative error is 1.8%, which occurs in the eigenvalue of smallest modulus. For Method 0, in both cases, the maximum relative error is about 3.3%, also occurring in the first eigenvalue. For Method 1, however, the eigenvalue of smallest modulus has an error of only 0.07%, but the maximum relative error is 1.6%, occurring in the second eigenvalue.

Example 5 (EX12) DRUM BOILER (Bengtsson, [2])

n = 5 m = 2

$$A = \begin{bmatrix} -0.129 & 0 & 0.0396 & 0.0250 & 0.0191 \\ 0.00329 & 0 & -0.779_{10}-4 & 0.122_{10}-3 & -0.621 \\ 0.0718 & 0 & -0.100 & 0.887_{10}-3 & -3.85 \\ 0.0411 & 0 & 0 & -0.0822 & 0 \\ 0.351_{10}-3 & 0 & 0.350_{10}-4 & 0.426_{10}-4 & -0.0743 \end{bmatrix}$$

$$B^T = \begin{bmatrix} 0 & 0 & 0 & 0.249_{10}-4 & 0 \\ 0.00139 & 0.359_{10}-4 & -0.00989 & 0 & -0.534_{10}-5 \end{bmatrix}$$

EIG (A) = -0.180, -0.0597 ± 0.0168i, 0.0, -0.0859.

In the form given, this problem is very badly scaled. Therefore,

before applying the pole assignment algorithms, the state and control variables are scaled, giving new variables $\underline{z} = D_1\underline{x}$ and $\underline{v} = D_2\underline{u}$ where $D_1 = \text{diag}\ \{1.0, 10.0, 0.1, 0.1, 10.0\}$ and $D_2 = \text{diag}\ \{1_{10}{}^{-4}, 1_{10}{}^{-2}\}$. The coefficient matrices of the scaled system are then given by $\tilde{A} = D_1 A D_1^{-1}$ and $\tilde{B} = D_1 B D_2^{-1}$. The scaling is symmetric and the eigenvalues of the system matrix are unaltered.

For the scaled system with a fairly arbitrary set of assigned poles, $\Delta = \{-0.01, -0.02, -0.03, -0.04, -0.05\}$, the condition of S is $\kappa_2(S) = 106.89$, and well-conditioned solutions are not to be expected. After two sweeps of Method 3 (approximately 9405 flops) a solution with $\kappa_2(X) = 131.83$, $\|F\|_2 = 6.3785$ and sensitivities $\{27.18, 29.30, 60.04, 41.24, 21.17\}$ is obtained. With tolerance 10^{-5} this procedure converges in six sweeps with an improved result such that $\kappa_2(X) = 113.63$, $\|F\|_2 = 6.1610$, and the pole sensitivities equal $\{26.92, 26.62, 50.55, 37.07, 22.08\}$.

Method 0, in Case 2, provides a solution with $\kappa_2(X) = 122.70$, $\|F\|_2 = 5.5067$ and sensitivities $\{29.67, 24.10, 52.64, 22.79, 32.22\}$ after two sweeps (approximately 4916 flops). The method here converges with tolerance 10^{-5} after 27 iterations to a solution with $\kappa_2(X) = 88.564$, $\|F\|_2 = 5.1424$ and sensitivities $\{25.68, 26.35, 41.43, 23.91, 23.27\}$.

The results produced by Method 1 (13697 flops) are considerably worse for this example than the results of Method 3 or 0. The solution here has $\kappa_2(X) = 226.85$, $\|F\|_2 = 10.845$ and sensitivities $\{24.55, 92.14, 99.27, 42.99, 35.73\}$.

With a slight modification of the assigned poles, a somewhat more satisfactory solution is obtained. We assign $\Delta = \{-0.01, -0.02, -0.03, -0.05, -0.06\}$ and obtain $\kappa_2(S) = 67.036$. With Method 3, after two sweeps (approximately 7849 flops) we achieve a solution with $\kappa_2(X) = 87.836$, $\|F\|_2 = 2.700$ and sensitivities $\{20.01, 27.56, 40.73, 22.92, 10.08\}$.

After seven sweeps the method is converged with a tolerance of 10^{-5} and produces a solution with $\kappa_2(X) = 58.131$, $\|F\|_2 = 2.3754$, and sensitivities {20.27, 18.19, 26.47, 19.46, 11.19}.

With Method 0, two sweeps (5026 flops) gives a solution with $\kappa_2(X) = 63.301$, $\|F\|_2 = 4.2217$ and sensitivities {23.55, 18.32, 28.22, 12.03, 16.33}. The procedure converges with tolerance 10^{-5} after 25 sweeps and gives an improved solution with $\kappa_2(X) = 51.413$, $\|F\|_2 = 2.1810$, and sensitivities {19.75, 17.40, 24.68, 16.23, 11.66}.

Method 1 (taking approximately 13930 flops) again gives rather poorer results than either Method 3 or 0. The solution has $\kappa_2(X) = 128.12$, $\|F\|_2 = 6.6642$ and sensitivities {19.34, 48.84, 55.82, 19.35, 15.98}.

Example 6 (EX 5) AIRCRAFT CONTROL PMF SYSTEM (Okada, Kihara, Kobayaski [18])

$n = 4 \qquad m = 2$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 5.32_{10}{-7} & -0.418 & -0.12 & 0.00232 \\ -4.62_{10}{-9} & 1 & -0.752 & -0.0239 \\ -0.561 & 0 & -0.3 & -0.0174 \end{bmatrix}$$

$$B^T = \begin{bmatrix} 0 & -0.172 & -0.0238 & 0 \\ 0 & 7.45_{10}{-6} & -7.78_{10}{-5} & 0.00369 \end{bmatrix}$$

$EIG(A) = -0.0048 \pm 0.0761i,\ -0.589 \pm 0.313i$.

This is a manufactured plant designed for the control of the system described in Example 2. We assign the same arbitrary set of eigenvalues $\Delta = \{-1, -2, -3, -4\}$. The condition of S is here $\kappa_2(S) = 24.251$ and we cannot expect such well-conditioned solutions as in Example 2. Using Method 3, after two sweeps (approximately 2854 flops) we obtain a solution with $\kappa_2(X) = 19.053$, $\|F\|_2 = 817.25$ and sensitivities {4.18, 9.33, 1.34, 6.77}. The method converges with tolerance 10^{-5} to a very slightly

improved result with $\kappa_2(X) = 19.033$, $\|F\|_2 = 813.79$, and sensitivities {4.10, 9.32, 1.20, 6.77}.

With Method 0, in Case 2, after two sweeps (approximately 2794 flops) the solution obtained has $\kappa_2(X) = 19.686$, $\|F\|_2 = 1388.8$ and sensitivities {4.14, 9.46, 2.08, 7.23}. This result is improved by further iteration and the solution converges with tolerance 10^{-5} after eleven sweeps to a solution such that $\kappa_2(X) = 18.974$, $\|F\|_2 = 808.64$ and the sensitivities are {4.19, 9.29, 1.02, 6.76}.

With Method 1 similar results are obtained (taking approximately 7458 flops) with $\kappa_2(X) = 19.021$, $\|F\|_2 = 934.68$ and sensitivities {4.21, 9.32, 1.02, 6.79}.

The two following examples were both obtained by random number generation, and have been designed such that a feedback matrix F exists which solves the pole assignment problem and has the property that A + BF is symmetric and has an orthonormal set of eigenvectors, that is, A + BF is perfectly conditioned.

Example 7 (EXSYM1) SYMMETRIC 1

n = 4 m = 2

$$A = \begin{bmatrix} -3.6240 & 4.9567_{10}{-2} & -2.4564_{10}{-1} & 1.3853_{10}{-2} \\ 3.3486_{10}{-1} & -1.8875 & -8.1251_{10}{-1} & -2.8102_{10}{-1} \\ -1.9958_{10}{-1} & -1.1335 & -2.2039 & -4.5523_{10}{-1} \\ 1.3784_{10}{-1} & -4.7140_{10}{-1} & -3.3229_{10}{-1} & -4.0605 \end{bmatrix}$$

$$B^T = \begin{bmatrix} 2.3122_{10}{-1} & 3.0761_{10}{-1} & 3.6164_{10}{-1} & 3.3217_{10}{-1} \\ 8.8339_{10}{-1} & 2.1460_{10}{-1} & 5.6642_{10}{-1} & 5.0153_{10}{-1} \end{bmatrix}$$

EIG(A) = -1.0427, -2.7966, -3.6660, -4.2706

Λ = -1, -2, -3, -4 .

Example 8 (EXSYM2) SYMMETRIC 2

n = 5 m = 2

$$A = \begin{bmatrix} -1.9437 & -7.5427_{10}12 & -5.2558_{10}{-1} & -4.5251_{10}{-1} & -7.2736_{10}{-1} \\ -2.8017_{10}{-1} & -2.4305 & 3.0264_{10}{-1} & 4.1276_{10}{-1} & -6.9057_{10}{-1} \\ -1.0495 & -2.3849_{10}{-3} & -2.1983 & -3.9156_{10}{-1} & -5.4305_{10}{-1} \\ -6.5064_{10}{-1} & 3.3864_{10}{-1} & -1.8021_{10}{-1} & -2.2044 & -5.6839_{10}{-1} \\ -5.8869_{10}{-1} & -3.0092_{10}{-1} & -8.8512_{10}{-3} & -2.4011_{10}{-1} & -1.6278 \end{bmatrix}$$

$$B^T = \begin{bmatrix} 2.9223_{10}{-1} & 5.9351_{10}{-1} & 6.3257_{10}{-1} & 4.8185_{10}{-1} & 1.2801_{10}{-1} \\ 4.8265_{10}{-1} & 4.3686_{10}{-1} & 9.1847_{10}{-1} & 4.1481_{10}{-1} & 2.1190_{10}{-1} \end{bmatrix}$$

EIG(A) = -0.99653, -1.2653, -1.9773, -2.7480, -3.4175

Δ = -1, -1, -2, -2, -3 .

Applying Method 2 to the first of these symmetric examples (Example 7), with decreasing tolerance levels, a solution with $\kappa_2(X) = 1.0002$ and sensitivities $1/c_i = 1.0000$ (to five figures), i = 1, 2, 3, 4, is obtained in 23 sweeps (approximately 24931 flops). Comparing the computed system matrix A + BF with the known symmetric solution we obtain maximum component errors of $\pm\ 1.78_{10}{-4}$.

Using Method 0 to solve the same problem we obtain a solution with $\kappa_2(X) = 1.0001$ and $1/c_i = 1.0000$, i = 1, 2, 3, 4 after 19 sweeps (approximately 24852 flops). The maximum component differences between the computed system matrix A + BF and the symmetric solution are now $\pm\ 1.06_{10}{-4}$.

For the second symmetric example (Example 8), Method 3 gives a solution with $\kappa_2(X) = 1.1393$ after two sweeps (approximately 7682 flops). Using this solution as an initial approximation for Method 0, a solution with $\kappa_2(X) = 1.0000$ and sensitivities $1/c_i = 1.0000$, $i = 1, 2, 3, 4, 5$, is obtained with one sweep of the method (a total of approximately 10720 flops). The component differences between the computed system matrix $A + BF$ and the known symmetric result are of maximum order 10^{-14}.

For Examples 1-6 complete results of the experiments described here are given in the Appendices. In Appendix I the conditioning of the converged solutions and the operation counts are summarized in tabular form. In Appendix II the computed gain matrices, system matrices and matrices of eigenvectors are listed for each example.

4.2. Discussion of Numerical Results

For problems where well-conditioned solutions may be expected, that is, where the conditioning of S, $\kappa_2(S)$, is reasonably close to unity, the methods all perform well and lead to robust solutions which are generally very similar. As shown in Table 1a), Methods 2/3 and Method 0 give good results after only two sweeps in most cases. Further iteration gives some improvement, as shown in Table 1b). The results of Method 1 are not, in general, as satisfactory as those of the other methods, however, even when these procedures are applied for only two sweeps.

Methods 2/3 do not attempt to minimize the conditioning of X, but to minimize a measure close to $\kappa_2(X)$. Thus, as expected, the results of Methods 2/3 have, in general, slightly less good over-all conditioning than the results of Method 0. Methods 2/3, however, always converge, and convergence is achieved very quickly, after only a small number of sweeps.

The performance of Method 0, on the other hand, depends very heavily on the bases selected for the spaces S_j and on the initial estimates chosen for the eigenvectors, as may be seen by comparing the results of Method 0, Case 2, with those of Method 0, Case 1. Method 0 does not necessarily converge, and when it does it tends to require rather a large number of sweeps to achieve the converged result. For systems where $n-m$ is small, Methods 2/3 are also more efficient per sweep than Method 0, although this does not appear to be the case where n is large compared to m. (We note that Methods 2/3 and Method 0, Case 2, are nearly always more efficient than Method 0, Case 1.)

Table 1

			Method	a) Solutions after two sweeps. κ(X)	‖F‖	flops	b) Solutions at convergence κ(X)	‖F‖	No. of sweeps
1.	EX4-A	n = 3 m = 2							
		‖A‖ = 13.922	2	7.8098	6.4788	743	7.8098	6.4788	3
		σ_m(B) = 1.0000	0(2)	7.7772	6.4859	1278	7.7772	6.4859	2
		κ(S) = 8.3427	1	7.7772	6.4859	3443	7.7772	6.4859	-
	EX4-B	n = 3 m = 2							
		‖A‖ = 13.922	2	3.2830	16.541	743	3.2827	16.469	4
		σ_m(B) = 1.000	0(2)	3.2732	16.461	1580	3.2732	16.461	2
		κ(S) = 3.6506	1	3.2732	16.461	3394	3.2732	16.461	-
2.	EX1	n = 4 m = 3							
		‖A‖ = 2.9309	2	3.6103	28.255	1503	3.6103	28.255	2
		σ_m(B) = 0.0949	0(2)	3.6690	25.931	1777	3.6103	26.540	10
		κ(S) = 4.9040	0(1)	3.7361	25.931	5024	3.6103	25.543	10
			1	6.1623	30.294	19371	6.1623	30.294	-
3.	EX13-A	n = 4 m = 2							
		‖A‖ = 12.998	3	4.5589	1.1553	3493	4.5355	1.1656	6
		σ_m(B) = 3.0652	0(2)	3.5834*	1.4093	2902	3.5834*	1.4093	1*
		κ(S) = 3.761	0(1)	3.4155	1.4003	4992	3.2811*	1.3292	11*
			1	3.4391	1.5408	8668	3.4391	1.5408	-
	EX13-B	n = 4 m = 2							
		‖A‖ = 12.998	3	3.2182	1.3911	3056	3.2122	1.4039	5
		σ_m(B) = 3.0652	0(2)	3.1974	1.4035	3180	3.1974	1.4035	3
		κ(S) = 3.2934	0(1)	3.2552	1.4013	4384	3.1969	1.3970	34
			1	$6.8_{10}5$	164.92	7602	$6.8_{10}5$	164.92	-
4.	EX7-A	n = 4 m = 2							
		‖A‖ = 95.975	3	156.03	137.74	3364	154.79	133.18	4
		σ_m(B) = 0.4000	0(2)	54.702	109.05	2613	36.957†	64.943	3(26)†
		κ(S) = 42.506	0(1)	46.654	94.717	4783	36.904†	62.675	3(7)†
			1	48.982	43.712	7242	48.882	43.712	-
	EX7-B	n = 4 m = 2							
		‖A‖ = 95.975	3	1.4579	131.31	3056	1.4478	122.16	7
		σ_m(B) = 0.4000	0(2)	1.5117	143.26	2965	1.4477	121.44	16
		κ(S) = 1.7655	0(1)	1.4706	133.75	5389	1.4478	121.76	12
			1	3.5436	151.74	7687	3.5436	151.74	-
5.	EX12-A	n = 5 m = 2							
		‖A‖ = 0.62810	3	131.83	6.3785	9405	113.63	6.1610	6
		σ_m(B) = 0.02490	0(2)	122.70	5.5067	4916	88.564	5.1424	27
		κ(S) = 106.89	0(1)	137.31	6.8804	7844	88.564	5.1424	26
			1	226.85	10.845	13697	226.85	10.845	-
	EX12-B	n = 5 m = 2							
		‖A‖ = 0.62810	3	87.836	2.700	7849	58.131	2.3754	7
		σ_m(B) = 0.02490	0(2)	63.301	4.2217	5026	51.413	2.1811	25
		κ(S) = 67.036	0(1)	69.499	2.6990	8426	51.219	2.2469	5
			1	128.12	6.6642	13930	128.12	6.6642	-
6.	EX5	n = 4 m = 2							
		‖A‖ = 1.5663	3	19.053	817.25	2854	19.033	813.79	5
		σ_m(B) = 0.00369	0(2)	19.686	1388.8	2794	18.974	808.64	11
		κ(S) = 24.251	0(1)	19.683	1386.7	4846	18.974	808.74	10
			1	19.021	934.68	7458	19.021	934.68	-

* Best result - process does not converge.
† Best result - process converges after (.) sweeps.

For problems where $\kappa_2(S)$ is small, the conditioning of X closely reflects the actual eigenvalue sensitivities, and the condition numbers $\kappa_2(X)$ obtained by all the methods are near to the optimal bound $\kappa_2(S)/\sqrt{n}$. In these examples the components of the gain matrices obtained are also as reasonably small as may be expected, given A, B and the choice of the assigned eigenvalue set Δ. The upper bound given by (5) considerably over-estimates $\|F\|_2$, and we find that $\|F\|_2$ is generally of the same order of magnitude as $\|A\|_2/\sigma_m(B)$, or smaller. As expected for these problems, small perturbations in the gain matrices then lead to proportionately small errors in the assigned poles.

We conclude that for problems where $\kappa_2(S)$ is small, Methods 2/3 are generally more reliable and efficient than the other methods, although all the methods can be expected to give good results. For problems where well-conditioned solutions cannot be achieved, the methods perform more erratically, and Method 0 can lead to rather better solutions than the other procedures, even though it may not be convergent. Within the limitations of the problem, however, all the methods can be expected to produce acceptable solutions. On the whole, Method 1 appears to be less reliable than Methods 0 or 2/3 and we recommend the latter as being generally most satisfactory.

5. Conclusions

The problem of pole assignment by state feedback for multivariable control systems is essentially under-determined. We demonstrate here that the extra degrees of freedom in the problem may be used to determine a robust, or well-conditioned, solution such as to minimize the sensitivities of the closed loop poles to perturbations in the system and gain matrices. For such robust solutions it is shown that bounds on the (mean square) magnitude of the closed loop transient response and on the norm of the feedback gain matrix are also minimized. A measure of the optimal conditioning that may be expected for a particular system with a given set of closed loop poles is described and used to assess the suitability of the given poles for assignment.

Four novel numerical methods are derived for constructing robust, well-conditioned solutions to the state feedback pole placement problem. The methods are applied to a number of practical test examples, and numerical results are presented and discussed. The tests indicate that the methods are stable and efficient. In cases where well-conditioned solutions may be expected, near optimal results are obtained. Introducing perturbations in the computed gain matrices leads only to correspondingly small errors in the assigned poles. The methods are all formed on different principles and exhibit different behaviour, however, and certain of the procedures may be regarded as more reliable than others.

Generalizations of these methods for degenerate systems and for the output feedback problem are expected to be easy to develop. Certain necessary theoretical results have already been derived [7] and numerical techniques are at present being explored. Extensions to techniques for modifying the locations of the assigned closed loop poles to improve further their insensitivity are also being considered.

6. References

[1] Barnett, S. Introduction to mathematical control theory. Oxford University Press, Oxford (1975).

[2] Bengtsson, G. A theory for control of multivariate systems. Lund Institute of Technology, Division of Auto. Control. Rpt. 7341 (1973).

[3] Davison, E. J. and Chow, S. G. "Perfect control in linear time-invariant multivariable systems: the control inequality principle," Proc. 8th Annual Princeton Conference on Information Sciences and Systems, Princeton University, pp. 73-79 (1974).

[4] Dongarra, J. J., Moler, C. B., Bunch, J. R. and Stewart, G. W. LINPACK User's Guide. SIAM, Philadelphia (1979).

[5] Fahmy M. M. and O'Reilly, J. "On eigenstructure assignment in linear multivariable systems", IEEE Trans. Auto Control AC-27. 690-693 (1982).

[6] Flamm, D. S. "A new proof of Rosenbrock's theorem on pole assignment," IEEE Trans. Auto. Control AC-25, 1128-1133 (1980).

[7] Fletcher, L. R., Kautsky, J., Kolka, G. K. G. and Nichols, N. K. Some necessary and sufficient conditions for eigenstructure assignment. University of Salford Department of Mathematics Rpt. (to appear).

[8] Kautsky, J. and Nichols, N. K. MEAP-1: MATLAB Eigenstructure Assignment Package - Mark 1. Flinders University School of Mathematical Sciences Rpt. (1982).

[9] Kautsky, J., Nichols, N. K., Van Dooren, P. and Fletcher, L., "Numerical Methods for robust eigenstructure assignment in control system design," Proc. of Workshop on Numerical Treatment of Inverse Problems for Differential and Integral Equations, Heidelberg, 1982, Birkhauser/Boston (to appear).

[10] Klein, G. and Moore, B. C. "Eigenvalue-generalized eigenvector assignment with state feedback," IEEE Trans. Auto. Control AC-22, 140-141 (1977).

[11] Maki, M. C. and Van de Vegte, J. "Optimization of multiple-input systems with assigned poles," IEEE Trans. Auto. Control AC-19, 130-133 (1974).

[12] Mayne, D. Q. and Murdock, P. "Model control of linear time invariant systems," Int. J. Control, 11, 223-227 (1970).

[13] Minimis, G. S. and Paige, C. C. An algorithm for pole assignment of time invariant linear systems. McGill University School of Computer Science Rpt. (1982).

[14] Moler, C. B. MATLAB User's Guide. University of New Mexico Dept. of Computer Science (1981).

[15] Moore, B. C. "On the flexibility offered by state feedback in multivariable systems beyond closed loop eigenvalue assignment," IEEE Trans. Auto. Control. AC-21, 689-692 (1976).

[16] Munro, N. Pole assignment, Proc. IEE 126, 549-555 (1979).

[17] Munro, N. "Design of controllers for open-loop unstable multivariate systems using the inverse Nyquist array," Proc. IEE 119, 1377-82 (1972).

[18] Okada, T., Kihara, M., and Kobayashi, S., "Design of perfect model following systems by geometric approach," Trans. Japan Soc. Aero. Space Sci., 22, 179-190 (1980).

[19] Porter, B. and D'Azzo, J. J. "Closed-loop eigenstructure assignment by state feedback in multivariable linear systems," Int. J. Control, 27, 487-492 (1978).

[20] Smith, B. T., Boyle, J. M., Dongarra,J. J., Garbow, B. S., Ikebe, Y., Klema, V. C., and Moler, C. B., Matrix Eigensystem Routines - EISPACK Guide. Lecture Notes in Computer Science, Vol. 6 2nd edition, Springer-Verlag (1976).

[21] Van Dooren, P. M. and De Wilde, P. Minimal cascade factorization of real and complex rational transfer matrices, IEEE Trans. Circ. and Syst. CAS-28, 390-400 (1981).

[22] Varga, A. "A Schur method for pole assignment," IEEE Trans. Auto. Control AC-26, 517-519 (1981).

[23] Wilkinson, J. H. The algebraic eigenvalue problem. Oxford University Press, Oxford (1965).

[24] Wonham, W. M. On pole assignment in multi-input controllable systems, IEEE Trans. Auto. Control AC-12, 660-665 (1967).

APPENDIX I Solutions at Convergence : Sensitivities

Example 1 (EX4) TEST : $\|A\|_2 = 13.922$ $\sigma_m(B) = 1.0000$

Table 1A

			Method 2	Method 0	Method 1
λ_1	1	$1/c_1$	3.9202	3.0420	1.0000
λ_2	1	$1/c_2$	1.4030	2.7151	3.9529
λ_3	3	$1/c_3$	3.9529	3.9529	3.9529
$\kappa(S)$	8.3427	$\kappa(X)$	7.8098	7.7772	7.7772
		$\|F\|_2$	6.4788	6.4859	6.4859
		flops	947	1278	3443
		sweeps	3	2	-

Table 1B

			Method 2	Method 0	Method 1
λ_1	-0.2	$1/c_1$	1.5898	1.3420	1.0000
λ_2	-0.2	$1/c_2$	1.4122	1.5494	1.7893
λ_3	-10	$1/c_3$	1.7893	1.7893	1.7893
$\kappa(S)$	3.6506	$\kappa(X)$	3.2827	3.2732	3.2732
		$\|F\|_2$	16.469	16.461	16.461
		flops	1273	1580	3394
		sweeps	4	2	-

Example 2 (EX1) AIRCRAFT CONTROL : $\|A\|_2 = 2.9309$ $\sigma_m(B) = 0.09488$

Table 2

			Method 2	Method 0 (2)	Method 0 (1)	Method 1
λ_1	-1	$1/c_1$	1.9437	1.9437	1.9437	3.1623
λ_2	-2	$1/c_2$	1.0000	1.0000	1.0000	3.1623
λ_3	-3	$1/c_3$	1.0000	1.0000	1.0000	1.0000
λ_4	-4	$1/c_4$	1.9437	1.9436	1.9437	1.0000
$\kappa_2(S)$	4.9040	$\kappa(X)$	3.6103	3.6103	3.6103	6.1623
		$\|F\|$	28.255	26.540	25.543	30.294
		flops	1749	12093	22714	19371
		sweeps	3	10	10	-

Table 2-E

Errors

	Method 2		Method 0 (2)		Method 0 (1)		Method 1	
λ_i	Abs.	%	Abs.	%	Abs.	%	Abs.	%
-1	0.0004	0.04	0.0003	0.03	0.0005	0.05	0.0027	0.27
-2	0.0012	0.06	0.0033	0.17	0.0023	0.11	0.0028	0.14
-3	0.0040	0.13	0.0000	0.00	0.0084	0.28	0.0013	0.04
-4	0.0012	0.03	0.0006	0.02	0.0009	0.02	0.0042	0.11

Table 2W Method 2 WE = <5, 25, 5, 1>

Errors

λ_i	$1/c_i$	Abs.	%		
-1	1.0000	0.0003	0.03	$\kappa(X)$	26.038
-2	1.0000	0.0012	0.06	$\|F\|$	12.584
-3	13.038	0.0008	0.03	flops	1500
-4	13.038	0.0015	0.04	sweeps	2

Example 3 (EX13) CHEMICAL REACTOR : $\|A\|_2 = 12.998$ $\sigma_m(B) = 3.0652$

Table 3A

			Method 3	Method 0(2)	Method 0(1)	Method 1
λ_1	-0.2	$1/c_1$	2.3747	1.8565	1.6904	1.4153
λ_2	-0.5	$1/c_2$	1.0739	1.4242	1.5331	1.8358
λ_3	-5.0566..	$1/c_3$	2.3590	1.5155	1.4918	1.4073
λ_4	-8.6659..	$1/c_4$	1.0899	1.7244	1.7541	1.8440
$\kappa(S)$	3.7610	$\kappa(X)$	4.5355	3.5834	3.2811	3.4391
		$\|F\|$	1.1656	1.4093	1.3292	1.5408
		flops	.7587	2901	27962	8668
		sweeps	6	1*	11*	-

Table 3B

			Method 3	Method 0(2)	Method 0(1)	Method 1
λ_1	-0.2	$1/c_1$	1.7515	1.7495	1.7400	$1.5_{10}3$
λ_2	-0.5	$1/c_2$	1.0813	1.0802	1.0975	$1.6_{10}3$
λ_3	-8.6659..	$1/c_3$	1.6653	1.1128	1.0703	$3.4_{10}5$
λ_4	-8.6659..	$1/c_4$	1.2675	1.7273	1.7538	$3.4_{10}5$
$\kappa(S)$	3.2934	$\kappa(X)$	3.2122	3.1974	3.1969	$6.8_{10}5$
		$\|F\|$	1.4039	1.4035	1.3970	164.92
		flops	5212	5166	81064	7602
		sweeps	5	3	34	-

Table 3A-E

Errors

λ_i	Method 3		Method O(2)		Method O(1)		Method 1	
	Abs.	%	Abs.	%	Abs.	%	Abs.	%
-0.2	0.0005	0.25	0.0056	2.8	0.0126	6.3	0.0030	1.5
-0.5	0.0012	0.24	0.0020	0.40	0.0034	0.69	0.0025	0.50
-5.0566..	0.0015	0.03	0.0002	0.004	0.0030	0.06	0.0034	0.07
-8.6659..	0.0028	0.03	0.0063	0.07	0.0064	0.07	0.0026	0.03

Table 3B-E

Errors

λ_i	Method 3		Method O(2)		Method O(1)		Method 1	
	Abs.	%	Abs.	%	Abs.	%	Abs.	%
-0.2	0.0019	0.95	0.0020	1.0	0.0096	4.8	3.65	1825
-0.5	0.0013	0.26	0.0004	0.08	0.0014	0.28	12.1	2420
-8.6659..	0.0028	0.03	0.0010	0.01	0.0017	0.02	3.9	45
-8.6659..	0.0013	0.02	0.0029	0.03	0.0107	0.12	.0017	.02

Table 4A-E

Errors

λ_i	Method 3		Method O(2)		Method O(1)		Method 1	
	Abs.	%	Abs.	%	Abs.	%	Abs.	%
-1	0.0081	0.81	0.0024	0.24	0.0080	0.80	0.0411	4.1
-2	0.0080	0.40	0.0180	0.90	0.0017	0.01	0.0674	3.3
-3	0.0012	0.04	0.0046	0.15	0.0254	0.84	0.0189	0.63
-4	0.0392	0.98	0.0308	0.77	0.0076	0.19	0.0316	0.79

Table 4B-E

Errors

λ_i	Method 3		Method O(2)		Method O(1)		Method 1	
	Abs.	%	Abs.	%	Abs.	%	Abs.	%
-0.1	0.0018	1.8	0.0034	3.4	0.0032	3.2	0.00007	0.07
-1.0	0.0002	0.02	0.0002	0.02	0.0004	0.04	0.0157	1.6
-3.0	0.0253	0.84	0.0137	0.46	0.0005	0.02	0.0092	0.31
-64.795	0.002	0.003	0.018	0.02	0.002	0.004	0.0314	0.05

Example 4 (EX7) NUCLEAR ROCKET ENGINE : $\|A\|_2 = 95.975$ $\sigma_m(B) = 0.4$

Table 4A

			Method 3	Method 0(2)	Method 0(1)	Method 1
λ_1	-1	$1/c_1$	1.6180	18.457	18.035	24.243
λ_2	-2	$1/c_2$	77.395	2.9675	3.3544	24.307
λ_3	-3	$1/c_3$	77.392	18.483	18.414	7.8961
λ_4	-4	$1/c_4$	2.3567	2.7602	3.0060	7.8198
$\kappa(S)$	42.506	$\kappa(X)$	154.79	36.957	36.904	48.982
		$\|F\|$	133.18	64.943	62.675	43.712
		flops	4972	4955	9293	7242
		sweeps	4	3*	3*	-

Table 4B

			Method 3	Method 0(2)	Method 0(1)	Method 1
λ_1	-0.1	$1/c_1$	1.0002	1.0002	1.0002	1.0070
λ_2	-1.0	$1/c_2$	1.0692	1.0692	1.0692	1.2279
λ_3	-3.0	$1/c_3$	1.0001	1.0001	1.0001	1.8163
λ_4	-64.795	$1/c_4$	1.0692	1.0692	1.0692	1.9077
$\kappa(S)$	1.7655	$\kappa(X)$	1.4478	1.4477	1.4478	3.5436
		$\|F\|$	122.16	121.44	121.76	151.74
		flops	7947	21586	29036	7687
		sweeps	7	16	12	-

Example 5 (EX12) DRUM BOILER : $\|A\|_2 = 0.62810$ $\sigma_m(B) = 0.02490$

Table 12A

			Method 3	Method O(2)	Method O(1)	Method 1
λ_1	-0.01	$1/c_1$	26.917	25.683	25.683	24.553
λ_2	-0.02	$1/c_2$	26.617	26.346	26.346	92.136
λ_3	-0.03	$1/c_3$	50.554	41.428	41.428	99.274
λ_4	-0.04	$1/c_4$	37.070	23.911	23.911	42.989
λ_5	-0.05	$1/c_5$	22.078	23.467	23.467	35.728
$\kappa(S)$	106.89	$\kappa(X)$	113.63	88.564	88.564	226.85
		$\|F\|$	6.1610	5.1424	5.1424	10.845
		flops	18569	61722	101822	13697
		sweeps	6	27	26	-

Table 12B

			Method 3	Method O(2)	Method O(1)	Method 1
λ_1	-0.01	$1/c_1$	20.271	19.748	19.777	19.340
λ_2	-0.02	$1/c_2$	18.191	17.400	17.250	48.837
λ_3	-0.03	$1/c_3$	26.467	24.675	24.700	55.824
λ_4	-0.05	$1/c_4$	19.460	16.227	16.075	19.351
λ_5	-0.06	$1/c_5$	11.189	11.664	11.755	15.978
$\kappa(S)$	67.036	$\kappa(X)$	58.131	51.413	51.219	128.12
		$\|F\|$	2.3754	2.181	2.2469	6.6642
		flops	22025	55930	36782	13930
		sweeps	7	25	5*	-

Example 6 (EX5) AIRCRAFT CONTROL PMF SYSTEM :

$\|A\|_2 = 1.5663 \quad \sigma_m(B) = 0.00369$

Table 5

			Method 3	Method O(2)	Method O(1)	Method 1
λ_1	-1	$1/c_1$	4.1839	4.1875	4.1876	4.2079
λ_2	-2	$1/c_2$	9.3192	9.2938	9.2938	9.3212
λ_3	-3	$1/c_3$	1.2775	1.0199	1.0193	1.0181
λ_4	-4	$1/c_4$	6.7676	6.7610	6.7609	6.7911
$\kappa(S)$	24.251	$\kappa(X)$	19.033	18.974	18.974	19.021
		$\|F\|$	813.79	808.64	808.74	934.68
		flops	5011	16634	22231	7594
		sweeps	5	11	10	-

APPENDIX II Solutions at Convergence: Feedback and System Matrices.

Example 1 (EX.4) TEST

A. $\Delta = \{1, 1, 3\}$

Method 2

F =

-1.6053E+00	3.0941E+00	-1.4887E+00
-2.0941E+00	4.2907E+00	-2.1966E+00

A + B*F

=

-1.6053E+00	4.0941E+00	-1.4887E+00
-2.0941E+00	4.2907E+00	-1.1966E+00
2.3006E+00	-3.6153E+00	2.3146E+00

X =

8.5211E-01	1.8142E-01	-6.4205E-01
4.3819E-01	4.3599E-01	-5.1606E-01
-2.8616E-01	8.8148E-01	5.6697E-01

Method 0(2)

F =

-1.6073E+00	3.0972E+00	-1.4899E+00
-2.0972E+00	4.2955E+00	-2.1984E+00

A + B*F

=

-1.6073E+00	4.0972E+00	-1.4899E+00
-2.0972E+00	4.2955E+00	-1.1984E+00
2.2955E+00	-3.6073E+00	2.3117E+00

X =

-3.9185E-01	7.6355E-01	-6.4254E-01
8.3805E-02	5.8518E-01	-5.1683E-01
9.1620E-01	2.7304E-01	5.6571E-01

Method 1

F =

-1.6073E+00	3.0972E+00	-1.4899E+00
-2.0972E+00	4.2955E+00	-2.1984E+00

A + B*F

=

-1.6073E+00	4.0972E+00	-1.4899E+00
-2.0972E+00	4.2955E+00	-1.1984E+00
2.2955E+00	-3.6073E+00	2.3117E+00

X =

3.1494E-01	7.9835E-01	-6.4254E-01
4.9491E-01	3.2330E-01	-5.1683E-01
8.0986E-01	-5.0804E-01	5.6571E-01

B. $\Delta = (-0, 2, -0, 2, -10)$

Method 2

F =

-6.7934E+00	1.2869E+01	-5.9113E+00
2.0676E+00	-4.5490E+00	8.5369E-01

A + B*F

=

-6.7934E+00	1.3869E+01	-5.9113E+00
2.0676E+00	-4.5490E+00	1.8537E+00
1.2742E+00	-2.6803E+00	9.4241E-01

X =

4.3895E-01	7.4596E-01	9.3836E-01
-1.6757E-01	5.2777E-01	-2.9426E-01
-8.8275E-01	4.0620E-01	-1.8135E-01

Method 0(2)

F =

-6.7866E+00	1.2855E+01	-5.9053E+00
2.0781E+00	-4.5713E+00	8.6316E-01

A + B*F

=

-6.7866E+00	1.3855E+01	-5.9053E+00
2.0781E+00	-4.5713E+00	1.8632E+00
1.2915E+00	-2.7166E+00	9.5789E-01

X =

-2.6776E-01	-8.7621E-01	9.3741E-01
2.6726E-01	-4.6711E-01	-2.9576E-01
9.2568E-01	-1.1859E-01	-1.8381E-01

Method 1

F =

-6.7866E+00	1.2855E+01	-5.9053E+00
2.0781E+00	-4.5713E+00	8.6316E-01

A + B*F

=

-6.7866E+00	1.3855E+01	-5.9053E+00
2.0781E+00	-4.5713E+00	1.8632E+00
1.2915E+00	-2.7166E+00	9.5789E-01

X =

3.1494E-01	8.6038E-01	-9.3741E-01
4.9491E-01	2.1136E-01	2.9576E-01
8.0986E-01	-4.6376E-01	1.8381E-01

Example 2 (EX.1) AIRCRAFT CONTROL

$\Delta = \{-1, -2, -3, -4\}$

Method 2

F =

7.9689E-01	3.5594E-01	-5.4029E-01	-8.9527E-02
5.9292E+00	-3.1747E+00	-6.0894E+00	-2.6640E+01
-7.8300E-01	3.9039E+00	3.3350E+00	1.1934E+00

A + B*F

=

.0000E+00	1.0000E+00	.0000E+00	.0000E+00
-4.0000E+00	-5.0000E+00	-1.4988E-15	1.0472E-14
-1.9597E-17	-4.9960E-16	-2.0000E+00	-2.0643E-16
3.6277E-14	2.9990E-14	1.3878E-17	-3.0000E+00

X =

7.0711E-01	8.8471E-17	-4.5935E-15	2.4254E-01
-7.0711E-01	2.8449E-16	1.4135E-14	-9.7014E-01
-4.8572E-17	-1.0000E+00	1.3878E-17	1.3010E-16
2.0955E-15	-6.9389E-17	1.0000E+00	2.0456E-14

Method 0(2)

F =

7.9587E-01	3.5519E-01	-7.1280E-01	-1.6213E-01
5.8299E+00	-3.2436E+00	-1.3013E+01	-2.1104E+01
-7.6505E-01	3.9174E+00	6.6160E+00	2.7849E+00

A + B*F

=

.0000E+00	1.0000E+00	.0000E+00	.0000E+00
-4.0000E+00	-5.0000E+00	-1.1884E-03	-2.7457E-03
-3.1912E-03	-2.4814E-03	-2.7156E+00	-4.5113E-01
-8.9775E-03	-6.1342E-03	-4.5113E-01	-2.2844E+00

X =

-7.0711E-01	8.4447E-04	1.2348E-03	-2.4253E-01
7.0711E-01	-1.6989E-03	-3.7044E-03	9.7014E-01
-1.3117E-04	-5.3329E-01	8.4593E-01	2.2523E-03
1.6114E-03	8.4593E-01	5.3328E-01	2.7919E-03

Method 0(1)

F =

7.9679E-01	3.5584E-01	-7.3152E-01	-1.4604E-01
5.9523E+00	-3.2264E+00	-1.2767E+01	-2.0057E+01
-7.8287E-01	3.9047E+00	6.9908E+00	2.4896E+00

A + B*F

=

.0000E+00	1.0000E+00	.0000E+00	.0000E+00
-4.0000E+00	-5.0000E+00	1.4357E-04	-2.3856E-03
8.7602E-04	4.2240E-04	-2.8095E+00	-3.9266E-01
-8.1413E-03	-5.4150E-03	-3.9266E-01	-2.1905E+00

X =

-7.0711E-01	-1.1046E-03	4.5598E-04	-2.4254E-01
7.0711E-01	2.2091E-03	-1.3679E-03	9.7014E-01
-5.6941E-04	4.3641E-01	8.9975E-01	4.6519E-04
1.8072E-03	-8.9974E-01	4.3641E-01	1.9129E-03

Method 1

F =

4.0852E-01	-3.2429E-02	-8.7928E-01	2.9466E-03
5.7200E+00	-3.3839E+00	-7.0145E+00	-2.8610E+01
-5.2576E-01	4.1611E+00	1.0080E+01	-7.0015E-01

A + B*F

=

.0000E+00	1.0000E+00	.0000E+00	.0000E+00
-2.0000E+00	-3.0000E+00	-6.5503E-15	-6.1331E-15
4.7613E-15	8.6042E-15	-3.6280E+00	4.8334E-01
-1.3350E-14	-7.0360E-15	4.8334E-01	-3.3720E+00

X =

7.0711E-01	4.4721E-01	1.3860E-15	2.1988E-16
-7.0711E-01	-8.9443E-01	-1.8943E-15	7.1731E-16
-1.4265E-15	-4.0496E-15	6.0993E-01	-7.9245E-01
-2.1088E-15	-1.0943E-15	7.9245E-01	6.0993E-01

Method 2 WE = <5, 25, 5, 1>

F =

2.3504E+00	7.4432E-01	-5.4029E-01	-1.7996E-02
6.7659E+00	-2.9655E+00	-6.0894E+00	-6.9971E+00
-1.8119E+00	3.6466E+00	3.3350E+00	2.4002E-01

A + B*F

=

.0000E+00	1.0000E+00	.0000E+00	.0000E+00
-1.2000E+01	-7.0000E+00	7.7716E-16	-1.0562E-14
-3.8336E-13	-1.2573E-13	-2.0000E+00	-2.3419E-16
-1.1371E-12	-3.7947E-13	2.0817E-16	-1.0000E+00

X =

2.0817E-16	-1.0374E-15	3.1623E-01	-2.4254E-01
-1.7347E-16	2.3107E-15	-9.4868E-01	9.7014E-01
.0000E+00	1.0000E+00	2.5709E-15	1.4246E-14
-1.0000E+00	.0000E+00	-2.7756E-17	3.0462E-14

Example 3 (EX.13) CHEMICAL REACTOR

A. Λ = {-0.2, -0.5, -5.0566.., -8.6659..}

Method 3

F =

1.0277E-01	-6.3333E-01	-1.1872E-01	1.4632E-01
8.3615E-01	5.2704E-01	-2.5775E-01	5.4269E-01

A + B*F

=

1.3800E+00	-2.0770E-01	6.7150E+00	-5.6760E+00
2.2062E-03	-7.8867E+00	-6.7420E-01	1.5059E+00
-1.4468E+00	1.8955E+00	-5.9780E+00	4.3519E+00
1.6474E-01	3.5535E+00	1.2081E+00	-1.9378E+00

X =

9.2772E-01	3.4733E-01	-7.2357E-01	2.3164E-01
1.4477E-03	1.0448E-01	-1.7246E-01	-8.6297E-01
-3.4291E-01	5.4497E-01	6.6793E-01	6.2845E-03
-1.4749E-01	7.5595E-01	-2.4017E-02	4.4899E-01

Method 0(2)

F =

9.4786E-02	-2.8080E-02	-8.1182E-02	4.1628E-02
1.1070E+00	1.5187E-01	8.1057E-01	-2.8314E-01

A + B*F

=

1.3800E+00	-2.0770E-01	6.7150E+00	-5.6760E+00
-4.3111E-02	-4.4495E+00	-4.6103E-01	9.1141E-01
-2.3080E+00	3.7633E+00	-9.2963E+00	6.8310E+00
1.5568E-01	4.2411E+00	1.2508E+00	-2.0567E+00

X =

-8.8333E-01	4.5921E-01	4.7613E-02	6.1555E-01
1.5071E-02	1.0622E-01	-5.9704E-01	-1.2015E-01
4.0597E-01	4.9291E-01	4.7860E-01	-7.5148E-01
2.3385E-01	7.3136E-01	6.4205E-01	2.0481E-01

Method 0(1)

F =

2.0429E-01	-1.4015E-01	2.8451E-01	-2.6517E-01
1.0657E+00	-2.5893E-01	6.2955E-01	-1.5248E-01

A + B*F

=

1.3800E+00	-2.0770E-01	6.7150E+00	-5.6760E+00
5.7875E-01	-5.0859E+00	1.6158E+00	-8.3092E-01
-2.0536E+00	4.9284E+00	-8.3113E+00	6.0715E+00
2.8007E-01	4.1138E+00	1.6662E+00	-2.4052E+00

X =

-9.7976E-01	-6.0748E-02	3.7497E-01	-5.4154E-01
-5.9267E-02	8.6697E-02	-5.6557E-01	-2.7311E-01
8.3210E-02	6.5333E-01	2.1640E-01	7.9504E-01
-1.7212E-01	7.4963E-01	7.0193E-01	-7.9072E-03

Method 1

F =

1.9527E-01	-4.4729E-02	1.9147E-01	-1.6056E-01
1.2117E+00	-2.5003E-01	8.0236E-01	-3.2953E-01

A + B*F

=

1.3800E+00	-2.0770E-01	6.7150E+00	-5.6760E+00
5.2753E-01	-4.5554E+00	1.0074E+00	-2.3691E-01
-2.5232E+00	5.0043E+00	-8.9607E+00	6.7473E+00
2.6982E-01	4.2179E+00	1.5605E+00	-2.2864E+00

X =

-4.0114E-01	-8.6615E-01	3.2980E-01	5.8027E-01
-1.4104E-01	-8.3625E-03	-5.7694E-01	1.4118E-01
-5.3009E-01	4.4096E-01	2.5862E-01	-7.9335E-01
-7.3362E-01	2.3510E-01	7.0106E-01	7.7359E-02

B. Δ = {-0.2, -0.5, -8.6659.., -8.6659..}

Method 3

F =

1.4165E-01	-6.9031E-01	5.5633E-02	1.4301E-02
1.1012E+00	1.7627E-01	8.0195E-01	-2.8312E-01

A + B*F

=

1.3800E+00	-2.0770E-01	6.7150E+00	-5.6760E+00
2.2302E-01	-8.2103E+00	3.1594E-01	7.5621E-01
-2.2366E+00	2.9343E+00	-9.1137E+00	6.7999E+00
2.0891E-01	3.4888E+00	1.4062E+00	-2.0878E+00

X =

8.9977E-01	4.2291E-01	-6.0138E-01	-8.3812E-02
-9.4798E-03	1.0583E-01	3.4456E-01	-8.3676E-01
-3.8516E-01	5.1105E-01	6.5388E-01	4.0389E-01
-2.0485E-01	7.4080E-01	-3.0342E-01	3.6011E-01

Method O(2)

F =

1.4120E-01	-6.9049E-01	5.5267E-02	1.4191E-02
1.1009E+00	1.7547E-01	8.0146E-01	-2.8448E-01

A + B*F

=

1.3800E+00	-2.0770E-01	6.7150E+00	-5.6760E+00
2.2048E-01	-8.2113E+00	3.1386E-01	7.5559E-01
-2.2361E+00	2.9366E+00	-9.1126E+00	6.8041E+00
2.0841E-01	3.4886E+00	1.4058E+00	-2.0879E+00

X =

-8.9811E-01	4.1021E-01	3.7499E-02	6.1504E-01
1.0070E-02	1.0565E-01	-8.7463E-01	-1.0659E-01
3.8738E-01	5.1710E-01	2.6081E-01	-7.5561E-01
2.0792E-01	7.4376E-01	4.0693E-01	1.9852E-01

Method 0(1)

F =

1.1481E-01	-6.9232E-01	5.7045E-02	6.7912E-03
1.0955E+00	1.7290E-01	7.9695E-01	-2.8656E-01

A + B*F

=

1.3800E+00	-2.0770E-01	6.7150E+00	-5.6760E+00
2.4099E-01	-8.2217E+00	3.2396E-01	7.2493E-01
-2.2149E+00	2.9426E+00	-9.0961E+00	6.8045E+00
2.1251E-01	3.4865E+00	1.4078E+00	-2.0940E+00

X =

9.2142E-01	3.5544E-01	1.5339E-01	-5.9678E-01
-1.1299E-03	1.0466E-01	-8.7898E-01	-6.1121E-02
-3.5306E-01	5.4156E-01	1.1287E-01	7.9135E-01
-1.6110E-01	7.5460E-01	4.3718E-01	-1.1780E-01

Method 1

F =

2.7941E+01	2.6128E+01	2.8570E+01	3.6818E+01
7.1303E+01	6.7870E+01	7.2798E+01	9.2601E+01

A + B*F

=

1.3800E+00	-2.0770E-01	6.7150E+00	-5.6760E+00
1.5810E+02	1.4409E+02	1.6225E+02	2.0976E+02
-1.9151E+02	-1.7956E+02	-2.0322E+02	-2.4360E+02
3.1789E+01	3.3954E+01	3.3799E+01	3.9721E+01

X =

-8.5127E-01	-8.4070E-01	1.7934E-01	1.7949E-01
2.4781E-02	-2.4564E-03	-8.7547E-01	-8.7544E-01
4.4079E-01	4.6699E-01	7.8161E-02	7.7949E-02
2.8361E-01	2.7411E-01	4.4191E-01	4.4194E-01

Example 4 (EX.7) NUCLEAR ROCKET ENGINE

A. Δ = {-1, -2, -3, -4}

Method 3

F =

9.2232E-01	-1.8648E+00	3.6799E-01	-2.9848E-01
-1.1577E+00	-1.3272E+02	4.4800E+00	-9.8945E+00

A + D*F

=

-5.0462E+00	-5.6213E+01	5.1932E-01	9.8999E-02
1.0000E-01	-1.0000E-01	.0000E+00	.0000E+00
1.0000E+00	.0000E+00	-5.0000E-01	-1.0000E+00
-4.6307E-01	-5.3086E+01	2.1920E+00	-4.3538E+00

X =

2.3322E-01	-7.1566E-01	6.9681E-01	-6.5628E-01
-2.5913E-02	3.7666E-02	-2.4028E-02	1.6828E-02
-9.4249E-01	1.2220E-02	7.9961E-03	3.7458E-01
-2.3802E-01	-6.9733E-01	7.1680E-01	6.5476E-01

Method 0(2)

F =

9.2663E-01	-1.4090E+00	2.7411E-01	-2.8675E-01
1.8607E+00	-6.3553E+01	7.8108E+00	-1.0577E+01

A + B*F

=

-4.7692E+00	-2.6582E+01	-1.6831E+00	8.6101E-01
1.0000E-01	-1.0000E-01	.0000E+00	.0000E+00
1.0000E+00	.0000E+00	-5.0000E-01	-1.0000E+00
7.4426E-01	-2.5421E+01	3.5243E+00	-4.6308E+00

X =

-6.9839E-01	5.2727E-01	-7.2286E-01	-6.4432E-01
7.7599E-02	-2.7751E-02	2.4926E-02	1.6521E-02
-2.5306E-02	-6.8561E-01	1.2975E-02	3.7454E-01
-7.1105E-01	-5.0115E-01	-6.9042E-01	6.6656E-01

Method 0(1)

F =

9.2597E-01	-1.4203E+00	2.7384E-01	-2.8664E-01
1.8609E+00	-6.1276E+01	7.6165E+00	-1.0471E+01

A + B*F

=

-4.8117E+00	-2.7320E+01	-1.7003E+00	8.6852E-01
1.0000E-01	-1.0000E-01	.0000E+00	.0000E+00
1.0000E+00	.0000E+00	-5.0000E-01	-1.0000E+00
7.4435E-01	-2.4511E+01	3.4466E+00	-4.5883E+00

X =

-6.9139E-01	5.3918E-01	7.0482E-01	6.6248E-01
7.6821E-02	-2.8378E-02	-2.4304E-02	-1.6987E-02
-5.0443E-02	-6.8528E-01	1.6608E-03	-3.7456E-01
-7.1661E-01	-4.8874E-01	7.0897E-01	-6.4849E-01

Method 1

F =

9.3819E-01	-1.2747E+00	2.4749E-01	-2.7303E-01
4.7066E+00	-4.0231E+01	1.2202E+01	-1.0000E+01

A + B*F

=

-4.6480E+00	-1.7057E+01	-3.4128E+00	1.7010E+00
1.0000E-01	-1.0000E-01	.0000E+00	.0000E+00
1.0000E+00	.0000E+00	-5.0000E-01	-1.0000E+00
1.9146E+00	-1.6092E+01	5.2809E+00	-4.7500E+00

X =

7.0237E-01	-7.1104E-01	6.1805E-01	6.2949E-01
-7.9041E-02	3.7423E-02	-2.1312E-02	-1.4141E-02
1.0168E-02	5.9401E-03	-4.9225E-01	-3.7434E-01
7.0745E-01	-7.0213E-01	-6.1258E-01	-6.8070E-01

B. Δ = {-0.1, -1.0, -3.0, -64.795..}

Method 3

F =

4.8043E-01	-1.0115E+00	5.4817E-02	1.7818E-01
7.8027E+01	1.9807E+00	3.9424E+01	-8.5309E+01

A + B*F

=

-3.3772E+01	-7.5073E-01	-1.5937E+01	3.1081E+01
1.0000E-01	-1.0000E-01	.0000E+00	.0000E+00
1.0000E+00	.0000E+00	-5.0000E-01	-1.0000E+00
3.1211E+01	7.9227E-01	1.6170E+01	-3.4524E+01

X =

-1.9245E-19	2.3238E-01	-7.1170E-01	-7.0192E-01
9.9959E-01	-2.5820E-02	2.4542E-02	1.0850E-03
-2.6452E-02	-9.4248E-01	3.8662E-03	2.1990E-02
1.0581E-02	-2.3886E-01	-7.0204E-01	7.1192E-01

Method 0(2)

F =

4.7559E-01	-1.0129E+00	5.3354E-02	1.7917E-01
7.7875E+01	2.1874E+00	3.9172E+01	-8.4522E+01

A + B*F

=

-3.4087E+01	-8.3898E-01	-1.6032E+01	3.1146E+01
1.0000E-01	-1.0000E-01	.0000E+00	.0000E+00
1.0000E+00	.0000E+00	-5.0000E-01	-1.0000E+00
3.1150E+01	8.7496E-01	1.6069E+01	-3.4209E+01

X =

9.4689E-21	2.3412E-01	7.0816E-01	-7.0622E-01
9.9950E-01	-2.6013E-02	-2.1419E-02	1.0916E-03
-2.9433E-02	-9.4249E-01	-1.0139E-03	2.1990E-02
1.1773E-02	-2.3713E-01	7.0563E-01	7.0765E-01

Method 0(1)

F

```
 4.7611E-01  -1.0125E+00   5.4034E-02   1.7214E-01
 7.7877E+01   2.1001E+00   3.9287E+01  -8.4932E+01
```

A + B*F

=

```
-3.3923E+01  -8.3174E-01  -1.5988E+01   3.1144E+01
 1.0000E-01  -1.0000E-01   .0000E+00    .0000E+00
 1.0000E+00   .0000E+00   -5.0000E-01  -1.0000E+00
 3.1151E+01   8.7205E-01   1.4115E+01  -3.4373E+01
```

X =

```
-2.0961E-16  -2.3231E-01   7.1047E-01   7.0434E-01
-9.9950E-01   2.5813E-02  -2.4479E-02  -1.0887E-03
 2.9226E-02   9.4240E-01  -2.8740E-03  -2.1790E-02
-1.1690E-02   2.3993E-01   7.0329E-01  -7.0952E-01
```

Method 1

F =

```
 2.2552E-01  -9.9402E-01  -1.0208E+00   1.2035E-01
 7.4292E+01  -5.7690E-01   1.2481E+02  -4.3986E+01
```

A + B*F

=

```
-5.0341E+01   3.8064E-01  -8.3854E+01   2.7323E+01
 1.0000E-01  -1.0000E-01   .0000E+00    .0000E+00
 1.0000E+00   .0000E+00   -5.0000E-01  -1.0000E+00
 2.9717E+01  -2.3076E-01   5.0324E+01  -1.7955E+01
```

X =

```
-2.2549E-18  -7.4054E-01  -4.1732E-01  -8.5443E-01
 9.9999E-01   8.2282E-02   1.4390E-02   1.3207E-03
 4.0156E-03   2.2630E-01   4.7642E-01   2.1363E-02
-1.6062E-03  -6.2739E-01   7.7373E-01   5.1912E-01
```

Example 5 (EX.12) DRUM BOILER

A. $\Lambda = \{-0.01, -0.02, -0.03, -0.04, -0.05\}$

Method 3

F =

1.2203E+00	-2.6992E-01	1.5668E-02	1.0396E+00	-5.8364E+00
4.7694E-01	-3.6438E-02	-1.4004E+00	-7.2173E-01	-1.1475E+00

A + B*F

=

-6.2705E-02	-5.0648E-03	2.0134E-01	1.4968E-01	-1.5759E-01
5.0022E-02	-1.3081E-03	-5.8064E-02	-1.3710E-02	-6.6220E-01
-3.9990E-02	3.6037E-03	3.8500E-02	7.2266E-02	7.4987E-02
3.4495E-02	-6.7210E-03	3.9013E-04	-5.6315E-02	-1.4533E-01
9.6312E-04	1.9458E-04	1.0978E-02	8.1140E-03	-6.8172E-02

X =

6.2821E-01	5.8561E-01	-7.6563E-01	-5.0878E-01	3.1917E-01
-6.3502E-01	7.8631E-01	-9.0023E-02	-8.4931E-01	-1.1348E-01
-1.3561E-01	5.5538E-02	2.4969E-01	-1.1119E-01	-5.2590E-01
4.2654E-01	1.7987E-01	-5.8158E-01	-3.7964E-02	7.7884E-01
4.2180E-02	5.7838E-02	-7.1589E-02	-7.7522E-02	4.5754E-02

Method 0(2)

F =

1.0123E+00	-2.9017E-01	3.2876E-01	1.3501E+00	-4.7714E+00
4.5164E-01	-3.9641E-02	-1.3673E+00	-6.8918E-01	-9.9265E-01

A + B*F

=

-6.6221E-02	-5.5101E-03	2.0594E-01	1.5420E-01	-1.3607E-01
4.9114E-02	-1.4231E-03	-5.6877E-02	-1.2542E-02	-6.5664E-01
-3.7488E-02	3.9205E-03	3.5227E-02	6.9047E-02	5.9673E-02
2.9315E-02	-7.2253E-03	8.1861E-03	-4.8584E-02	-1.1881E-01
1.0982E-03	2.1168E-04	1.0801E-02	7.9402E-03	-6.8999E-02

X =

6.9799E-01	-4.5759E-01	-6.6133E-01	3.5069E-01	4.6427E-01
-5.3507E-01	-8.7872E-01	2.2955E-01	8.8783E-01	2.4401E-01
-1.3244E-01	-1.0298E-01	3.2667E-01	2.3653E-01	-4.4570E-01
4.5460E-01	-7.4044E-02	-6.3256E-01	-1.7057E-01	7.2128E-01
4.8007E-02	-4.8752E-02	-5.5689E-02	6.1159E-02	7.7606E-02

Method 0(1)

F

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

A + B*F

=

[illegible]	-5.5101E-03	2.0594E-01	1.5420E-01	-1.3607E-01
4.9111E-02	-1.4231E-03	-5.4877E-02	-1.2542E-02	-6.5664E-01
-3.7180E-02	3.9205E-03	3.5227E-02	6.9047E-02	5.9673E-02
2.9315E-02	-7.2253E-03	8.1862E-03	-4.8584E-02	-1.1881E-01
[illegible]	2.1169E-04	1.0801E-02	7.9402E-03	-6.8999E-02

X =

-6.9779E-01	-4.5759E-01	6.6133E-01	-3.5068E-01	[illegible]
5.3207E-01	-3.7972E-01	-2.2755E-01	-8.8783E-01	2.4401E-01
[illegible]	-1.0290E-01	-3.2467E-01	-2.3453E-01	-4.4570E-01
-4.5165E-01	-7.1044E-02	6.3957E-01	1.7057E-01	7.2123E-01
[illegible]	-4.8752E-02	5.5689E-02	-6.1159E-02	7.7604E-02

Method 1

F =

1.8565E+[illegible]	-1.1178E-01	-5.2767E-01	5.8679E-01	-1.0515E+01
5.3242E-01	-2.5585E-02	-1.4073E+00	-7.3473E-01	-1.6133E+00

A + B*F

=

-5.4994E-02	-3.5563E-03	2.0038E-01	1.4787E-01	-2.2234E-01
5.2014E-02	-9.1851E-04	-5.8314E-02	-1.4177E-02	-6.7892E-01
-4.5476E-02	2.5304E-03	3.9186E-02	7.3552E-02	1.2106E-01
5.0336E-02	-3.2813E-03	-1.3139E-02	-6.7589E-02	-2.6182E-01
6.6688E-04	1.3662E-04	1.1015E-02	8.1835E-03	-6.5685E-02

X =

6.2026E-01	6.9776E-01	-6.6492E-01	-5.1670E-01	-1.4611E-01
7.3583E-01	-3.9312E-01	2.2191E-01	-8.4479E-01	4.1840E-01
1.1878E-02	-2.3365E-01	3.2524E-01	-1.0354E-01	5.2675E-01
2.6667E-01	5.4899E-01	-6.3223E-01	-5.0099E-02	-7.2526E-01
5.0774E-02	5.1013E-02	-5.6169E-02	-7.8276E-02	-1.1045E-02

B. $\Delta = \{-0.01, -0.02, -0.03, -0.05, -0.06\}$

Method 3

F =

2.1179E-01	-3.7342E-01	2.0333E-01	1.0306E+00	-1.6529E+00
4.1379E-01	-4.9913E-02	-1.3364E+00	-6.7545E-01	-5.5222E-01

A + B*F

=

-7.1193E-02	-6.9354E-03	2.1163E-01	1.5611E-01	-7.4240E-02
4.7755E-02	-1.7920E-03	-5.5407E-02	-1.2047E-02	-6.4502E-01
-3.3714E-02	4.9368E-03	3.1178E-02	6.7689E-02	1.6114E-02
2.7510E-02	-9.2902E-03	5.0629E-03	-5.6552E-02	-4.6133E-02
1.3504E-03	2.6656E-04	1.0503E-02	7.0669E-03	-7.1351E-02

X =

7.2230E-01	5.0204E-01	-7.4010E-01	-3.5459E-01	1.6387E-01
-4.9271E-01	8.5186E-01	1.9803E-02	-8.9360E-01	1.7903E-01
-1.3063E-01	8.7668E-02	2.7995E-01	-2.0892E-01	-5.5033E-01
4.2363E-01	1.0950E-01	-6.0745E-01	1.5457E-01	7.9114E-01
5.0991E-02	5.1978E-02	-6.7065E-02	-7.8687E-02	6.1136E-02

Method O(2)

F =

9.7749E-01	-3.5266E-01	-1.2069E-01	8.5764E-01	-1.6118E+00
4.1724E-01	-4.7933E-02	-1.3668E+00	-6.9836E-01	-5.0514E-01

A + B*F

=

-7.1004E-02	-6.6627E-03	2.0602E-01	1.5293E-01	-6.8305E-02
4.7879E-02	-1.7208E-03	-5.6856E-02	-1.2871E-02	-6.3913E-01
-3.4085E-02	4.7406E-03	3.5172E-02	6.9954E-02	1.1459E-02
2.8450E-02	-8.7812E-03	-3.0053E-03	-6.0845E-02	-4.0133E-02
1.2819E-03	2.5596E-04	1.0798E-02	7.9892E-03	-7.1603E-02

X =

6.8794E-01	-4.7511E-01	-6.5015E-01	2.9478E-01	2.3677E-01
-5.5105E-01	-8.6869E-01	2.5262E-01	8.7274E-01	4.6079E-01
-1.3307E-01	-9.7079E-02	3.3089E-01	2.7723E-01	-4.7001E-01
4.5074E-01	-8.7866E-02	-6.3329E-01	-2.6430E-01	7.0928E-01
4.7157E-02	-5.0030E-02	-5.4209E-02	6.8668E-02	8.7282E-02

Method 0(1)

F =

1.0009E+00	-3.4303E-01	-2.2369E-01	7.8704E-01	-1.6647E+00
4.2024E-01	-4.6775E-02	-1.3796E+00	-7.0741E-01	-5.1011E-01

A + B*F

=

-7.0587E-02	-6.5018E-03	2.0423E-01	1.5167E-01	-6.9996E-02
4.7986E-02	-1.6792E-03	-5.7319E-02	-1.3196E-02	-6.3931E-01
-3.4381E-02	4.6261E-03	3.6445E-02	7.0850E-02	1.1950E-02
2.9032E-02	-8.5414E-03	-5.5700E-03	-6.2603E-02	-4.1452E-02
1.2659E-03	2.4978E-04	1.0867E-02	8.0376E-03	-7.1576E-02

X =

-6.5819E-01	-4.9320E-01	6.5321E-01	-2.8202E-01	2.4024E-01
5.2501E-01	-9.5755E-01	-2.4642E-01	-8.6657E-01	4.7557E-01
1.3456E-01	-9.0907E-02	-3.2977E-01	-2.8991E-01	-4.6396E-01
-4.3873E-01	-1.0234E-01	6.3314E-01	2.8467E-01	7.0216E-01
-4.4663E-02	-5.1340E-02	5.4612E-02	-6.6550E-02	8.8519E-02

Method 1

F =

1.6022E+00	-1.3701E-01	-7.3165E-01	3.7489E-01	-6.2927E+00
4.7455E-01	-2.7510E-02	-1.3723E+00	-7.0845E-01	-1.0251E+00

A + B*F

=

-6.3038E-02	-3.8239E-03	2.0526E-01	1.5153E-01	-1.4058E-01
4.9936E-02	-9.8762E-04	-5.7054E-02	-1.3233E-02	-6.5780E-01
-3.9753E-02	2.7208E-03	3.5717E-02	7.0953E-02	6.2886E-02
4.4005E-02	-3.4117E-03	-1.8218E-02	-7.2865E-02	-1.5669E-01
9.7592E-04	1.4690E-04	1.0828E-02	8.0431E-03	-6.8826E-02

X =

5.3444E-01	7.7260E-01	-6.6166E-01	-3.9258E-01	5.0420E-02
8.1717E-01	-2.2048E-01	2.2885E-01	-8.9729E-01	-1.9600E-01
2.9826E-02	-2.1029E-01	3.2654E-01	-1.6239E-01	-5.7445E-01
2.0902E-01	5.5383E-01	-6.3253E-01	8.5340E-02	7.9287E-01
4.4976E-02	5.9377E-02	-5.5733E-02	-8.4291E-02	2.0118E-02

Example 6 (EX.5) AIRCRAFT CONTROL PMF SYSTEM

$\Delta = \{-1, -2, -3, -4\}$

Method 3

F =

6.3619E+01	3.5300E+01	-9.9150E+00	-1.5380E-01
-9.2599E+01	-5.0465E+01	-7.1423E+00	-8.0684E+02

A + B*F

=

.0000E+00	1.0000E+00	.0000E+00	.0000E+00
-1.0943E+01	-6.4899E+00	1.5853E+00	2.2762E-02
-1.5069E+00	1.6380E-01	-5.1547E-01	4.2532E-02
-9.0269E-01	-1.8621E-01	2.7364E-01	-2.9946E+00

X =

2.6886E-01	-3.8992E-01	5.4594E-03	-2.3983E-01
-2.6886E-01	7.7983E-01	-1.6378E-02	9.5933E-01
9.2440E-01	-4.8399E-01	-1.2724E-02	-1.4885E-01
3.0245E-02	7.4719E-02	9.9977E-01	2.8628E-03

Method 0(2)

F =

6.3608E+01	3.5298E+01	-9.9071E+00	-1.5895E-01
3.0763E+01	-2.3721E+01	-2.5240E+01	-8.0731E+02

A + B*F

=

.0000E+00	1.0000E+00	.0000E+00	.0000E+00
-1.0940E+01	-6.4894E+00	1.5838E+00	2.3645E-02
-1.5163E+00	1.6176E-01	-5.1425E-01	4.2692E-02
-4.4748E-01	-8.7530E-02	2.0686E-01	-2.9964E+00

X =

-2.6863E-01	3.9119E-01	4.4272E-03	2.3982E-01
2.6863E-01	-7.8237E-01	-1.3282E-02	-9.5930E-01
-9.2382E-01	4.8458E-01	-1.3606E-02	1.4893E-01
-4.7291E-02	-6.3490E-03	9.9981E-01	-7.4306E-03

Method 0(1)

F =

6.3608E+01	3.5290E+01	-9.9070E+00	-1.6019E-01
3.3145E+01	-2.3068E+01	-2.5705E+01	-8.0732E+02

A + B*F

=

.0000E+00	1.0000E+00	.0000E+00	.0000E+00
-1.0940E+01	-6.4894E+00	1.5838E+00	2.3856E-02
-1.5165E+00	1.6171E-01	-5.1421E-01	4.2722E-02
-4.3662E-01	-8.5122E-02	2.0485E-01	-7.9964E+00

X =

2.6063E-01	3.9119E-01	-4.1850E-03	2.3983E-01
-2.6963E-01	7.8238E-01	1.2557E-02	-9.5930E-01
9.2382E-01	4.8458E-01	1.3813E-02	1.4892E-01
4.7229E-02	-5.7392E-03	-9.9982E-01	-6.9473E-03

Method 1

F =

6.3723E+01	3.5318E+01	-9.9632E+00	-2.0070E-01
4.2806E+02	7.8565E+01	-1.6485E+02	-8.0986E+02

A + B*F

=

.0000E+00	1.0000E+00	.0000E+00	.0000E+00
-1.0957E+01	-6.4922E+00	1.5924E+00	3.0807E-02
-1.5499E+00	1.5331E-01	-5.0205E-01	4.3883E-02
1.0186E+00	2.8990E-01	-3.0829E-01	-3.0058E+00

X =

-2.6894E-01	-3.9118E-01	3.5347E-03	-2.3913E-01
2.6894E-01	7.8235E-01	-1.0604E-02	9.5652E-01
-9.2379E-01	-4.8416E-01	2.0407E-02	-1.4688E-01
4.4289E-02	-2.2237E-02	-9.9973E-01	-7.9471E-02

APPENDIX III Improved Methods for Steps A/0 and F

We consider here new methods for implementing Steps A/0 and F of the numerical algorithms described in Section 3.0-3.2. The methods derive from the following theorem [7], [9].

Theorem A Given $\Lambda = \text{diag}\{\lambda_j\}$ and X, non-singular, then $\exists\ F$, a solution to (3) if and only if

$$U_1^T (A - X\Lambda X^{-1}) = 0 , \tag{III.1}$$

where

$$B = (U_0, U_1) \begin{bmatrix} \Sigma \\ 0 \end{bmatrix} V^T . \tag{III.2}$$

Then F is given explicitly by

$$F = V\Sigma^{-1} U_0^T (X\Lambda X^{-1} - A) . \tag{III.3}$$

It follows that if $\underline{x}_j$ is an eigenvector of $A + BF$ corresponding to the assigned eigenvalue λ_j, then, since $U_1^T(AX - X\Lambda) = 0$, we must have $U_1^T(A - \lambda_j I)\,\underline{x}_j = 0$. Hence $\underline{x}_j$ must belong to the m-dimensional null space $S_j = N\left[U_1^T(A - \lambda_j I)\right]$.

The basic steps of the numerical algorithm now become

Step A/0: Compute the S.V.D. of matrix B, given by (III.3), to determine U_0, U_1, Σ and V, and construct orthonormal bases, comprised by the columns of matrices S_j, $\hat{S}_j$, for the space $S_j = N\left[U_1^T (A - \lambda_j I)\right]$ and its complement $\hat{S}_j$ for $\lambda_j \epsilon \Delta$, $j = 1, 2, \ldots n$.

Step X: As before.

Step F: Find the matrix $M \equiv A + BF$ by solving $MX = X\Lambda$ and compute F explicitly from $F = V\Sigma^{-1} U_0^T (M - A)$.

In Step A/O construction of the bases for S_j and $\hat{S}_j$ can again be accomplished by S.V.D. (Case 1) or QR (Case 2) decompositions, as follows.

Case 1 (SVD) We determine the singular value decomposition of $U_1^T(A - \lambda_j I)$ in the form

$$U_1^T(A - \lambda_j I) = Z_j [\Gamma_j, 0] [\hat{S}_j, S_j]^T,$$

and then the columns of S_j, $\hat{S}_j$ give the required orthonormal bases.

Case 2 (QR) We determine the QR decomposition of $\left(U_1^T (A - \lambda_j I)\right)^T$ partitioned as

$$\left(U_1^T(A - \lambda_j I)\right)^T = [\hat{S}_j, S_j] \begin{bmatrix} R_j \\ 0 \end{bmatrix}.$$

Then S_j, $\hat{S}_j$ are the required matrices.

In Step F the matrix $M = X\Lambda X^{-1}$ is constructed by solving the equation $X^T M^T = (X\Lambda)^T$ for M^T using a direct L-U decomposition (or Gaussian elimination) method. This process is stable for well-conditioned matrix X. The computation of F is then achieved by straightforward matrix multiplication.

The revised algorithm is considerably more efficient than the original. In Step A/O the orthonormal bases for S_j and $\hat{S}_j$ are determined directly using only one matrix decomposition, and there is no need to compute, or store, the matrices $\tilde{V}_j$, $\tilde{S}_j$ or V_j. Furthermore, for $j = 1, 2, ..n$, the decompositions are now applied to matrices of order $(n-m)x\ n$ rather than order $(m+n) \times n$, which considerably decreases the total work. In Step F the computational work is slightly increased due to the extra matrix multiplication involved; this additional work is insignificant, however, in comparison with the savings made in Step A/O.

Results obtained by the revised methods for the test examples of Section 4. are essentially the same as the results achieved by the original procedures. For Methods 2/3 and 1 the change of basis for S_j has no significant effect. For Method 0, the new basis leads to different initial approximations and, as would be expected, to somewhat different solutions. The overall performance of the methods remains very satisfactory, and for systems where $\kappa_2(S)$ is small, the methods all produce robust, well-conditioned solutions to the pole assignment problem. Numerical results are published elsewhere [9].

APPENDIX VI Modifications for complex eigenvalues.

VI.1 Modifications The numerical methods described in Section 3.0 take a modified form in the case where the assigned eigenvalue set Δ contains complex conjugate pairs λ_j and $\bar{\lambda}_j$. The basic theory remains essentially the same, but the subspaces S_j and $\bar{S}_j$ from which the corresponding eigenvectors $\underline{x}_j$ and $\bar{\underline{x}}_j$ are chosen now have complex bases. Using complex arithmetic the two steps Step A/0 and Step F of the numerical algorithm remain the same. The resulting matrix F is in complex form, then, but the imaginary parts of the components of F are negligible within rounding error. With slight modifications to Step F, the gain matrix F may be calculated in real arithmetic only.

The principle modifications occur in Step X. We have implemented such alterations only for Methods 2/3 and Method 0. For Method 0 the changes are fairly straightforward and we describe these first.

VI.1.1. Method 0. The number p of complex pairs of poles to be assigned is specified, and it is assumed that, in the given set Δ of eigenvalues, the complex pairs occur first. An initial set of vectors $\underline{x}_j$, $j = 1,2,..n$, is chosen such that $\underline{x}_{2j} = \bar{\underline{x}}_{2j-1}$ and $\underline{x}_{2j-1} \in S_{2j-1}$ for $j \leq p$, and $\underline{x}_j \in S_j$ for $j > 2p$. We note that $S_{2j} = \bar{S}_{2j-1}$ for $j \leq 2p$, and the basis for only one of the subspaces corresponding to a complex eigenvalue pair needs to be computed. We denote the basis obtained for this space by $S_{2j-1} \equiv S_j^R + i\, S_j^I$ and we may store S_j^R, S_j^I separately for $j \leq p$. To implement Method 0 we simply compute the vector $\underline{y}_{2j-1}$ orthogonal to the current space $X_{2j-1} = \langle \underline{x}_i,\ i \neq 2j-1 \rangle$, as previously, and replace old vector $\underline{x}_{2j-1}$ by the projection of $\underline{y}_{2j-1}$ into the subspace S_{2j-1} and old vector $\underline{x}_{2j}$ by the conjugate of this projection, for $j = 1,2,..p$. For $j = 2p+1, 2p+2,...n$, the procedure is carried on as described in Section 3.3.2. The process is repeated in sweeps through the matrix X until changes in the conditioning $\kappa_2(X)$ are

less than the tolerance given.

VI.1.2 Methods 2/3. For these methods we also assume that the number p of complex pairs of poles in the set Δ is specified, and that these pairs are given by λ_{2j-1}, $\overline{\lambda}_{2j-1}$, $j = 1,2,..p$. The objective of Methods 2/3 is to obtain a set of real orthonormal vectors $\tilde{\underline{x}}_j$, $j = 1,2,..n$, such that the projection of these vectors into the subspaces S_j produces a well-conditioned set of eigenvectors $\underline{x}_j$ corresponding to the assigned eigenvalues. In the case of complex conjugate poles it is necessary to modify the projection procedure to produce complex conjugate eigenvectors, and also to modify the measure which is used to determine the set $\tilde{\underline{x}}_j$. For $j \le p$ we let $\tilde{\underline{x}}_{2j-1} \equiv \underline{z}_j^R$ and $\tilde{\underline{x}}_{2j} \equiv \underline{z}_j^I$ represent the real and imaginary parts of a vector $\underline{z}_j$, and define $\underline{x}_{2j-1}$ to be the projection of $\underline{z}_j$ into S_{2j-1} and $\underline{x}_{2j} = \overline{\underline{x}}_{2j-1}$. We require $\underline{z}_j$ to be close to space S_{2j-1} for $j \le p$ and vector $\tilde{\underline{x}}_j$ to be close to space S_j for $j > 2p$. Essentially now we rotate the vectors $\tilde{\underline{x}}_j$ so that the <u>planes</u> corresponding to complex pairs and the <u>vectors</u> corresponding to reals remain orthogonal to each other and close to the appropriate subspaces (of dimensions $2m$ and m, respectively). The measure $\hat{\nu}$ to be minimized (Method 2) now takes the form

$$\hat{\nu} = \sum_{j=1}^{p} \omega_j \left(\| \hat{S}_{2j-1}^H \underline{z}_j \|_2^2 + \| \hat{S}_{2j}^H \overline{\underline{z}}_j \|_2^2 \right) + \sum_{j=2p+1}^{n} \omega_j \| \hat{S}_j^T \tilde{\underline{x}}_j \|_2^2$$

However, since $\hat{S}_{2j-1} \equiv \hat{S}_j^R + i\hat{S}_j^I$ and

$\hat{S}_{2j} = \overline{\hat{S}}_{2j-1} = \hat{S}_j^R - i\hat{S}_j^I$ for $j \le p$, we may write

$$\tfrac{1}{2}\left(\| \hat{S}_{2j-1}^H \underline{z}_j \|_2^2 + \| \hat{S}_{2j}^H \overline{\underline{z}}_j \|_2^2 \right) = \| \hat{S}_j^{RT} \underline{z}_j^R + \hat{S}_j^{IT} \underline{z}_j^I \|_2^2 + \| \hat{S}_j^{IT} \underline{z}_j^R - \hat{S}_j^{RT} \underline{z}_j^I \|_2^2 .$$

The rotations applied to $\tilde{\underline{x}}_j$, $j = 1,2,..n$ are thus chosen to minimize

$$\hat{\upsilon} = \sum_{j=1}^{p} 2\omega_j \left(\| \hat{S}_j^{R^T} \tilde{\underline{x}}_{2j-1} + \hat{S}_j^{I^T} \tilde{\underline{x}}_{2j} \|_2^2 + \| \hat{S}_j^{I^T} \tilde{\underline{x}}_{2j-1} - \hat{S}_j^{R^T} \tilde{\underline{x}}_{2j} \|_2^2 \right)$$

$$+ \sum_{j=2p+1}^{n} \omega_j \| \hat{S}_j^T \tilde{\underline{x}}_j \|_2^2 \ .$$

The complementary measure υ to be maximized (Method 3) takes the corresponding form

$$\upsilon = \sum_{j=1}^{p} 2\omega_j \left(\| S_j^{R^T} \tilde{\underline{x}}_{2j-1} + S_j^{I^T} \tilde{\underline{x}}_{2j} \|_2^2 + \| S_j^{I^T} \tilde{\underline{x}}_{2j-1} - S_j^{R^T} \tilde{\underline{x}}_{2j} \|_2^2 \right)$$

$$+ \sum_{j=2p+1}^{n} \omega_j \| S_j^T \tilde{\underline{x}}_j \|_2^2 \ .$$

The rotations to be applied are now rather more complicated to determine, but explicit expressions for the sines and cosines of the optimal rotation angles have been derived, using arguments similar to those given in Section 3.3.1. Eight different cases are distinguished. For rotations between vectors $\tilde{\underline{x}}_j$ $(j > 2p)$ corresponding to <u>real</u> eigenvectors, the rotations remain the same as those derived in Section 3.3.1. No rotations are applied between the vectors $\tilde{\underline{x}}_{2j-1}$ and $\tilde{\underline{x}}_{2j}$ $(j \leq p)$ corresponding to the real and imaginary parts of a complex pair of eigenvectors. The remaining cases involve rotations between vectors $\tilde{\underline{x}}_j$ corresponding to reals and to the real and/or imaginary parts of complex pairs.

VI.2 <u>Implementation</u>. The procedures for the complex cases have been implemented using MATLAB, and executive files for carrying out the modified steps of the algorithm are available.

VI.3 Numerical Results. The methods perform very much as in the case of real eigenvalue assignment. Method 0 does not necessarily converge, and the results depend heavily upon the initial vectors chosen. The condition numbers $\kappa_2(X)$ tend to oscillate from sweep to sweep, but for examples where $\kappa_2(S)$ is small, good results are generally obtained. The amount of work per sweep is considerably increased, however, since all the decompositions required must be computed in complex arithmetic.

For Methods 2/3 the majority of the computation is carried out in real arithmetic and, although some more complicated logical decisions are required, the methods perform satisfactorily without a large increase in work. These methods are always convergent, and for examples where $\kappa_2(S)$ is small the methods produce well-conditioned solutions to the pole assignment problem.