Improving the Performance of certain Algorithms in Eigenstructure Assignment*

George Miminis
Department of Computer Science
Memorial University of Newfoundland
St. John's, NFLD, CANADA, A1B 3X5
e-mail:miminis@cs.mun.ca

CONTENTS

Abstract— The DEscriptor Multi-input Eigenstructure Assignment problem using State Feedback (DEMESAS) is considered. It is pointed out, by referring to the literature, that in many situations the final step of the DEMESAS is (or can be) the solution of the matrix equation

$$(A - BF)X = EXL \Longleftrightarrow BFX = AX - EXL \quad (1)$$

with respect to F. We show why a "straightforward" algorithm that has been often used for the solution of (1) is *Numerically Unstable*, and we present a new algorithm for the solution of (1). We show why the new algorithm overcomes the numerical problems of the "straightforward" algorithm, and we present a numerical example which supports our results.

*This work was supported by NSERC grant #OGP0944. A copy of the *dvi* file of this paper may be obtained by connecting to *triton.cs.mun.ca* (134.153.1.19) by *anonymous ftp*, change directory to *pub/miminis* and in binary mode get the file *ieeemed3.dvi*. The accompanying software, *PolePack.tar.Z* may also be found in the same directory.

I. NOMENCLATURE

$\mathbb{R}$	= Real numbers
$\mathbb{C}$	= Complex numbers
Greek letters	= Scalars
Upper Roman	= Matrices
Lower Roman	= Vectors and indices
Superscript T	= Transposition
$\chi(A)$	= $\|A\| \left\|A^{-1}\right\|$ the condition number of A with respect to inversion
$\mathcal{R}(A)$	= The column space of A
$\lambda(A)$	= The eigenvalues of A
I, O	= The identity and zero matrices
o	= The zero vector
e_i	= The ith column of I

II. INTRODUCTION

Consider the continuous time-invariant descriptor system

$$E\dot{x}(t) = Ax(t) + Bu(t) \quad (2)$$

where $E \in \mathbb{R}^{n\times n}$, $A \in \mathbb{R}^{n\times n}$ is the open-loop system matrix, $B \in \mathbb{R}^{n\times m}$ is the control influence matrix, $x(t) \in \mathbb{R}^n$ is the state of the system at time t, and $u(t) \in \mathbb{R}^m$ is the input or control of the system. System (2) is said to be completely controllable if and only if

$$\{\forall \lambda \in \mathbb{C} \Rightarrow \mathbf{rank}(B, A - \lambda E) = n\} \wedge \{\mathbf{rank}(B, E) = n\}$$

For definitions on the controllability of descriptor systems see for example [4]. An important problem in control theory is to guarantee the stability of (2) by choosing $u(t)$. This may be accomplished by using the state feedback $u(t) = -Fx(t)$, with $F \in \mathbb{R}^{m\times n}$, which gives the closed-loop system

$$E\dot{x}(t) = (A - BF)\,x(t). \quad (3)$$

It can be proven that when E is not singular, (3) is stable if all the eigenvalues of the pencil $[(A-BF), E]$ have negative real parts. The above discussion also applies to discrete time systems. The only difference is that all the eigenvalues of the corresponding pencil should be less than 1 in absolute value. A method that is concerned only with the placement of the eigenvalues at the right points is that of *Eigenvalue Assignment*. According to this method, we are given a completely controllable system (E, A, B) and a self conjugate set Λ of at most n scalars; an F may then be computed such that $\lambda[(A-BF), E] - \{-\infty, \infty\} \subseteq \Lambda$. This definition takes into consideration the possibility of a singular E and it guarantees that the resulting pencil $[(A-BF), E]$ is regular ($\lambda[(A-BF), E] \neq \mathbb{C}$). It may be shown that when $m > 1$ (multi-input case) there is no unique F that accomplishes eigenvalue assignment. It appears that the extra freedom in F was first identified, for the case $E = I$ in [16], and it was associated with freedom in the selection of the eigenvectors that are assigned along with the given eigenvalues (closed-loop eigenvectors). More elaborate discussions on the subject may be found in [10], and in [11] for the case $E \neq I$. Given now that normally the plant is known with some uncertainty, the question of how to use the extra freedom in order to design a control system that satisfies various stability and performance specifications in the face of plant uncertainty emerges. This gives rise to the *Eigenstructure Assignment* problem. Actually the computation of a feedback that will satisfy various robustness criteria (stability and/or performance) is the central subject of *Robust Control*, where along with eigenstructure assignment, other methods have also been evolved like *Linear Quadratic Regulator* (LQR) and *Linear Quadratic Gaussian* (LQG) optimal control, H_∞ optimal control, *Adaptive Control* etc. Among these, eigenstructure assignment is probably the simplest, and since it also appears to be quite successful, it has naturally become the subject of extensive research, as well as the method of choice for a fair number of applications.

Next we give a brief account of a small sample of the respective literature, including various robustness criteria that researchers have attempted to satisfy via eigenstructure assignment. In [16] the closed-loop eigenvectors are chosen so that a desirable distribution of the modes among the components of the output $y(t) = Cx(t)$ (with $C \in \mathbb{R}^{p\times n}$), is achieved. To see this, let (λ_i, x_i) be a closed-loop eigenpair and z_i the corresponding left closed-loop eigenvector, then the output vector may be given by

$$y(t) = \sum_{i=1}^{n} Cx_i \left(z_i^T x(0)\right) e^{\lambda_i t}.$$

If now we choose x_i such that for example, $Cx_i = (2, 1, 0, \ldots, 0)^T$ the ith mode will appear in the first two components of $y(t)$ and it will be twice as large in the first component than in the second. In [12] the assignment of principal closed loop eigenvectors is considered. [16], [12] are mathematical treatments of the subject and along with [21] include a considerable number of interesting results that were actually rediscovered later. In some occasions (for example, [6], [7], [8]), the eigenvalue assignment problem is solved via the eigenstructure assignment problem. This however is not numerically advisable since eigenvalue assignability may be accomplished without the computation of the closed-loop eigenvectors, (see [2], [13], [14], [17], [18] and for a counterexample see [13]). In [5] a parametric approach to the eigenstructure assignment problem is proposed and good numerical properties are claimed for the algorithms within, however no evidence of the latter is given. In [19] a set of desired closed-loop eigenvectors are given along with the closed-loop eigenvalues. Since however the given eigenvectors may not be feasible, a number of least squares problems is solved so that the "closest" feasible eigenvectors to the corresponding given eigenvectors may be found. In [20] the additional freedom beyond eigenvalue assignability is used to minimize the index

$$J = \sum_{i=1}^{n} \omega_i \left\| Fe_i \right\|_2^2$$

with ω_i being desired "weights". It is interesting to observe that the weights may be chosen so that the state feedback will effectively result into a specific output feedback. To see this consider for example the case (see [20]) $n = 19$ and choose $\omega_i = 1$ for $i \in S = \{1, 2, 7, 12, 14\}$ and $\omega_i = 100$ for $i \notin S$ so that only the 1,2,7,12,14 columns of F are significant. Then with $y(t) = Cx(t)$, where $C = (e_1, e_2, e_7, e_{12}, e_{14})^T$ and $K = FC^T$ we have

$$u = -Fx \approx -Ky.$$

In [1] the following optimization problem is solved with the additional freedom

$$\left\{ \begin{array}{ll} \min\limits_{u=-Fx} \int_{t_0}^{\infty} \left(x^T Q x + u^T R u\right) dt & \\ \text{Subject to} \qquad \lambda(A - BF) = \Lambda & \end{array} \right\}$$

where Q is symmetric positive semidefinite and R symmetric positive definite. By far however, the most extensively studied robustness criterion, is that of optimizing the condition number of the eigenproblem of the pencil $[(A-BF), E]$ (see for example [3], [10], [11], [22], [23]).

In general, when the freedom in the choice of the eigenvectors is to be used in order to solve a problem beyond eigenvalue assignment, first an optimum set of eigenvectors is computed so that the additional problem is solved, and then the optimum set of eigenvectors along with the corresponding given eigenvalues are assigned. Although at first glance it may not be obvious that this two step procedure may be used in some of the above references, often the corresponding methods in these references may be tailored to suit the two step procedure. In this paper we will only be interested in the second step of the two step procedure and the reason is twofold. Initially the second step may be common to a large number of applications despite what the first step attempts to accomplish. Therefore a numerically sound solution of this step would be welcome. The lack of such a numerical solution provides the second reason. Next we formally define the problem that we will address in this paper.

Problem: Given $E \in \mathbb{R}^{n\times n}, A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}$, with $\text{rank}(B) = m$; a desirable self conjugate set of eigenvalues $\Lambda = \{\lambda_1, \ldots, \lambda_r\}$, with multiplicities not exceeding m; and a corresponding set of feasible eigenvectors $X = (x_1, \ldots, x_r)$, with $r \leq \text{rank}(E)$; compute $F \in \mathbb{R}^{m\times n}$ so that

$$(A - BF)\,X = EXL \tag{4}$$

with $L = \text{diag}(\lambda_1, \ldots, \lambda_r)$, and $[(A-BF), E]$ being regular (that is, $\lambda[(A-BF), E] \neq \mathbb{C}$). ■

For ease of reference sometimes we will refer to the problem defined above as the DEMESAS (see abstract). Note that in (4) L can be diagonal only when possible multiplicities in Λ do not exceed m. This is actually the case that is interesting in practice, since a closed-loop eigenvalue with multiplicity greater than m will appear as a defective eigenvalue of the pencil $[(A-BF), E]$, (see [10], [21] for the case $E = I$ and [11] for the $E \neq I$ case). Defective eigenvalues however are sensitive to perturbations in the data. This implies that unless it is absolutely necessary, we should avoid assigning eigenvalues of multiplicity greater than m, and this will be the case we will consider in this paper. In section III we present a new algorithm for the solution of (4), whereas in section IV we point out why the algorithm in III is superior to a straightforward algorithm that is commonly used for the solution of (4). As a result of this discussion some heuristics are also suggested that improve the numerical properties of the algorithm in III. Finally in section IV we also present a numerical example that demonstrates the superiority of the algorithm in III over the straightforward algorithm, and we finish with our conclusion in section V.

III. A Stepwise Algorithm for the DEMESAS

The algorithm begins by separating the uncontrollable part of the system from the completely controllable part. It then assigns an appropriate set of eigenpairs to the resulting completely controllable system. The separation is accomplished by an algorithm presented in [14] where orthogonal matrices U, V, and W are computed such that

$$U^TAV = \left(\begin{array}{c|c} A_1 & \widehat{A} \\ \hline O & \widetilde{A} \end{array}\right), U^TEV = \left(\begin{array}{c|c} E_1 & \widehat{E} \\ \hline O & \widetilde{E} \end{array}\right)$$

and

$$U^TBW = \left(\begin{array}{c} B_1 \\ \hline O \end{array}\right)$$

where all the uncontrollable eigenvalues have been accumulated in $(\widetilde{A}, \widetilde{E})$. It is rather straightforward to show that

$$V^TX = \left(\begin{array}{c|c} X_1 & \widehat{X} \\ \hline O & \widetilde{X} \end{array}\right),$$

with $\lambda(\widetilde{A}, \widetilde{E}) \cap \Lambda = \emptyset \Leftrightarrow \widetilde{X} = O$. If we take now $\left(F_1, \widehat{F}\right) = W^TFV$ and $L = \left(\begin{array}{c|c} L_1 & O \\ \hline O & \widetilde{L} \end{array}\right)$, equation (4) is equivalent to

$$\left(\begin{array}{c|c} (A_1 - B_1F_1)\,X_1 & (A_1 - B_1F_1)\,\widehat{X} + \left(\widehat{A} - B_1\widehat{F}\right)\widetilde{X} \\ \hline O & \widetilde{A}\widetilde{X} \end{array}\right)$$

$$= \left(\begin{array}{c|c} E_1X_1L_1 & E_1\widehat{X}\widetilde{L} + \widehat{E}\widetilde{X}\widetilde{L} \\ \hline O & \widetilde{E}\widetilde{X}\widetilde{L} \end{array}\right). \tag{5}$$

From (5) we see that if $\widetilde{X} = O$ then we need to solve

$$(A_1 - B_1F_1)\,X_1 = E_1X_1L_1 \tag{6}$$

for F_1, and since the desired set of eigenvectors is feasible, we are guaranteed that the rest of (5) will

hold. If however $\widetilde{X} \neq O$, meaning that $\lambda(\widetilde{A}, \widetilde{E}) \cap \Lambda \neq \emptyset$ (some of the uncontrollable eigenvalues are also desired), then once (6) is solved we also need to solve

$$(A_1 - B_1 F_1)\,\widehat{X} + \left(\widehat{A} - B_1 \widehat{F}\right)\widetilde{X} = E_1 \widehat{X} \widetilde{L} + \widehat{E} \widetilde{X} \widetilde{L} \tag{7}$$

with respect to $\widehat{F}$, and since the desired eigenvectors are feasible, the rest of (5) should also hold. It is worth noticing that we may change (assign) the eigenvector of an uncontrollable eigenvalue so long as the desired eigenvector is feasible. This is of course a well known result (see for example [16], [21]) the difference being that here it is apparent in an algorithmic way and thus how to proceed in order to accomplish it is straightforward . However due to lack of space, in this paper we will only consider the case $\lambda(\widetilde{A}, \widetilde{E}) \cap \Lambda = \emptyset$ and thus show how to solve (6) efficiently. The general case will be treated in a full paper ([15]).

It is also worth noting that if the pencil $(\widetilde{A}, \widetilde{E})$ is singular, so will $[(A - BF), E]$ for every F_1. If however (A_1, E_1) is singular, (5) suggests that there may be a way to change this in $[(A_1 - B_1 F_1), E_1]$ by choosing the right F_1. In the subsequent development we will see that this is possible, and we will show how it can be accomplished.

We are now ready to present an algorithm for the solution of (6) with (E_1, A_1, B_1) being completely controllable. For simplicity we will assume that the size of B_1 is $n \times m$. The matrix E_1 will be allowed to be singular. The algorithm assigns real eigenpairs one at a time in a single step, whereas complex eigenpairs are assigned two at a time as conjugate pairs in a double step. In this way complex arithmetic is completely avoided.

To facilitate this we will assume that complex conjugate eigenpairs appear successively in L_1 and X_1. Furthermore we will assume that if $\mu_1 \pm i\nu_1$ are two desired complex conjugate eigenvalues, they will appear in L_1 as a 2×2 diagonal block, say L_{11} of the form

$$L_{11} = \begin{pmatrix} \mu_1 & \nu_1 \\ -\nu_1 & \mu_1 \end{pmatrix}.$$

If in addition, $x_1 \pm i x_2$ are the corresponding desired eigenvectors, then the two vectors x_1, x_2 should appear as columns of X instead. This is clearly justifiable since $\mathcal{R}(x_1, x_2) = \mathcal{R}(x_1 + i x_2, x_1 - i x_2)$.

Let now $\widetilde{V}$, and $\widetilde{U}$ be the orthogonal matrices of the QR decompositions of X_1, and $E_1 \widetilde{V}$ respectively, and consider the partitioning

$$\begin{aligned}
X_1 \equiv \widetilde{V}^T X_1 &= \left(\begin{array}{c|c} X_{11} & X_{12} \\ \hline O & X_3 \end{array}\right) \\
\widetilde{U}^T \left(E_1 \widetilde{V}\right) &= \left(\begin{array}{c|c} E_{11} & E_{12} \\ \hline O & E_3 \end{array}\right) \\
\widetilde{U}^T A_1 \widetilde{V} &= \left(\begin{array}{c|c} A_{11} & A_{12} \\ \hline & A_3 \end{array}\right) \\
B_1 \equiv \widetilde{U}^T B_1 &= \left(\begin{array}{c} B_{11} \\ \hline B_3 \end{array}\right) \\
L_1 &= \left(\begin{array}{c|c} L_{11} & O \\ \hline O & L_3 \end{array}\right),
\end{aligned}$$

with A_{11} and B_{11} being $n \times 2$ and $2 \times m$ respectively, and

$$X_{11} = \begin{pmatrix} \xi_{11} & \xi_{12} \\ 0 & \xi_{22} \end{pmatrix}, \; E_{11} = \begin{pmatrix} \varepsilon_{11} & \varepsilon_{12} \\ 0 & \varepsilon_{22} \end{pmatrix}.$$

The algorithm then attempts to assign the eigenpairs $(\mu_1 \pm i v_1, x_1 \pm i x_2)$ by computing the $m \times 2$ matrix F_{11} in $F_1 \widetilde{V} = (F_{11}, F_3)$ as follows:

$$\begin{aligned}
(A_1 - B_1 F_1)\, X_1 &= E_1 X_1 L_1 \Longleftrightarrow \\
\widetilde{U}^T (A_1 - B_1 F_1)\, \widetilde{V} \widetilde{V}^T X_1 &= \widetilde{U}^T E_1 \widetilde{V} \widetilde{V}^T X_1 L_1 \Longleftrightarrow
\end{aligned}$$

$$\begin{aligned}
&\left(\begin{array}{c|c} (A_{11} - B_1 F_{11})\, X_{11} & \times \\ \hline & (A_3 - B_3 F_3) X_3 \end{array}\right) \\
&= \left(\begin{array}{c|c} E_{11} X_{11} L_{11} & \times \\ \hline O & E_3 X_3 L_3 \end{array}\right). \qquad (8)
\end{aligned}$$

By observing the first two columns of (8) we may consider the equation

$$B_1 F_{11} X_{11} = A_{11} X_{11} - \left(\frac{H}{O}\right), \tag{9}$$

where if, E_{11} is nonsingular, we take $H = E_{11} X_{11} L_{11}$ and by solving (9) with respect to F_{11} we assign $(\mu_1 \pm i v_1, x_1 \pm i x_2)$. If however E_{11} is singular we cannot assign $(\mu_1 \pm i v_1, x_1 \pm i x_2)$. In this case our aim will be to make certain that the computed F_{11} will result in a regular pencil $[(A_1 - B_1 F_1), E_1]$. This may be accomplished by taking H in (9) to be any nonsingular 2×2 matrix with $\|H\|$ being comparable to the magnitude of our data (the latter is essential in order to avoid unnecessary numerical problems). The reason for this choice of H becomes apparent later when we

continue with the next eigenpair assignment. In the assignment we just described we considered a complex conjugate eigenpair; if alternatively we considered the assignment of a real eigenpair (λ_1, x_1) then instead of (9) we would need to solve

$$B_1 f_{11} \xi_{11} = a_{11} \xi_{11} - \begin{pmatrix} \eta \\ o \end{pmatrix} \tag{10}$$

where f_{11}, a_{11} are $m \times 1$ and $n \times 1$ respectively. If now $\varepsilon_{11} \neq 0$ we may take $\eta = \varepsilon_{11} \xi_{11} \lambda_1$ in (10) and solve

$$B_1 f_{11} = a_{11} - \begin{pmatrix} \varepsilon_{11} \lambda_1 \\ o \end{pmatrix} \tag{11}$$

with respect f_{11}. If however $\varepsilon_{11} = 0$, then λ_1 obviously cannot be assigned. In this case take $\eta \neq 0$ with $|\eta|$ comparable to the magnitude of our data and instead solve

$$B_1 f_{11} = a_{11} - \begin{pmatrix} \eta \\ o \end{pmatrix} \tag{12}$$

in order to ascertain that $[(A_1 - B_1 F_1), E_1]$ is regular. Once (9) or (10) is solved, equation (8) takes the form

$$\left(\begin{array}{c|c} H & \times \\ \hline O & (A_3 - B_3 F_3) X_3 \end{array}\right) = \left(\begin{array}{c|c} E_{11} X_{11} L_{11} & \times \\ \hline O & E_3 X_3 L_3 \end{array}\right). \tag{13}$$

It is now clear why a nonsingular H (or a nonzero η) will have the desired effect regarding the regularity of the closed-loop pencil. From (13) we may continue in a similar manner with the next assignment by considering the equation $(A_3 - B_3 F_3) X_3 = E_3 X_3 L_3$. Note that the system (E_3, A_3, B_3) can be proven to be completely controllable (see [14] for a proof).

We may observe from the above process that after every step the number of the states of the next system (for example (E_3, A_3, B_3) above) becomes one or two less than in the previous step (for example (E_1, A_1, B_1) above) depending on whether a real eigenpair or a complex conjugate eigenpair was assigned, respectively. The above process may continue until the number of inputs of the current system becomes greater than or equal to the number of states. At this point the equation that corresponds to (9) will be underdetermined and cannot uniquely determine the remaining columns of F_1, and henceforth some modifications must be made. In order to simplify the presentation we will assume we are at the beginning of the process, allocating $(\mu_1 \pm \nu_1, x_1 \pm i x_2)$, and that $n = m$. Let us now consider the partitioning we used in the $n > m$ case, along with the following partitioning of a QR-like decomposition of $\widetilde{U}^T B_1$,

$$\widetilde{U}^T B_1 \widetilde{W} = \left(\begin{array}{c|c} B_{11} & B_{12} \\ \hline O & B_3 \end{array}\right),$$

and let

$$\widetilde{W}^T F_1 \widetilde{V} = \left(\begin{array}{c|c} F_{11} & F_{12} \\ \hline & F_3 \end{array}\right),$$

where this time B_{11} is a 2×2 upper triangular matrix, and F_{12} is $2 \times (n-2)$. Once F_{11} is computed by solving (9) the following equation may be formed

$$\left(\begin{array}{c|c} H & H X_{11}^{-1} X_{12} + (A_{12} - B_{11} F_{12} - B_{12} F_3) X_3 \\ \hline O & (A_3 - B_3 F_3) X_3 \end{array}\right) = \left(\begin{array}{c|c} E_{11} X_{11} L_{11} & (E_{11} X_{12} + E_{12} X_3) L_3 \\ \hline O & E_3 X_3 L_3 \end{array}\right). \tag{14}$$

We may observe from (14) that we first need to compute F_3, and subsequently F_{12} may be computed from the equation

$$B_{11} F_{12} = \left[H X_{11}^{-1} X_{12} - (E_{11} X_{12} + E_{12} X_3) L_3\right] X_3^{-1} + A_{12} - B_{12} F_3. \tag{15}$$

The computation of F_3 will also involve steps like the above. Thus when $m \geq n$ the algorithm has in fact two parts. The forward part, where all the elements of F_{ii} are computed by solving equations of the kind (9) or (10) and the backward part, where the elements of $F_{i,i+1}$, are computed by solving equations of the kind (15) (F_{12} in our presentation). Note that no inverses need be computed in (15) (for example $Y = H X_{11}^{-1} \Leftrightarrow Y X_{11} = H$ and Y may be computed by solving $Y X_{11} = H$).

Once all the equations of the kind (9), (10) and (15) are solved, we need to apply the history of orthogonal transformations back, in order to eventually compute F. To see how we may accomplish this, assume for the sake of presentation, that we begin the process with $n > m$, and after a specific number of steps we reach F_r for some r, with F_r being square. Assume also that throughout the process we applied double steps only. We may now compute $\widetilde{F}_r = \widetilde{W}(F_{rr}, \ldots, F_{n-1,n-1})$ and finally F is obtained by

$$F = W \left[\left(F_{11}, F_{33}, \ldots, F_{r-2,r-2}, \widetilde{F}_r\right) \widetilde{V}^T, \widehat{F}\right] V^T,$$

where, as we may recall, $\widehat{F}$ comes from the separation of the uncontrollable part of the system during the initial transformation, and in the case $\lambda\left(\widetilde{A},\widetilde{E}\right)\cap\Lambda=\emptyset$ it can assume any convenient value (for example $\widehat{F}=O$ which minimizes the Frobenius norm of F with respect to $\widehat{F}$).

IV. Discussion of the Algorithm, Heuristics and a Numerical Example

If equation (1) appears to be too simple and thus unworthy of special attention, it certainly has not received much. Thus far the methods employed for its solution are variations of the following general scheme.

(i) Compute $G=AX-EXL$.

(ii) Solve the system $BY=G$ with respect to Y.

(iii) Solve $FX=Y$ with respect to F.

Clearly, the above process makes the accuracy of F depend on the condition numbers $\chi(B)$, $\chi(X)$ (see for example [9] pp. 79-81). Another point about the above approach is that, if E is singular the approach works in such a way that the closed loop pencil $[(A-BF),E]$ is always singular, even if the system is completely controllable. To overcome this, the following two-step method has been recommended in [8].

(i) Compute an orthonormal matrix N that spans the null space of E. Then compute a matrix D such that $E+ANN^T+BDN^T$ is nonsingular.

(ii) Solve $BF(X,N)=(EXL-AX,D)$ with respect to F.

The advantage of the algorithm presented in section III, over the above three-step approach along with the two-step fix, is twofold. First, the stepwise manner which the new algorithm employs, makes the task of producing a regular closed loop pencil $[(A-BF),E]$ (when this is mathematically possible) rather straightforward (see equations (9), (12)). Second, the accuracy of the computed F, by the new algorithm, does not depend on the condition number of the entire X. Instead, it may be apparent from equations (9), (11), (14), (15) that the computed F depends on the condition numbers of some 2×2 diagonal blocks of X_1, as well as the condition number of the $m\times m$ south-east block of X_1. For example if $n=9$, $m=3$ and $\Lambda=\{c,\bar{c},r,r,c,\bar{c},\times,\times,\times\}$, where $(c,\bar{c})$ represents a complex conjugate pair of eigenvalues, r a real eigenvalue and $\times$ any eigenvalue, then with

$$X_1=\begin{pmatrix} \bullet & \bullet & \times & \times & \times & \times & \times & \times & \times \\ & \bullet & \times & \times & \times & \times & \times & \times & \times \\ & & \times & \times & \times & \times & \times & \times & \times \\ & & & \times & \times & \times & \times & \times & \times \\ & & & & \bullet & \bullet & \times & \times & \times \\ & & & & & \bullet & \times & \times & \times \\ & & & & & & \bullet & \bullet & \bullet \\ & & & & & & & \bullet & \bullet \\ & & & & & & & & \bullet \end{pmatrix}$$

the computed F will depend on the condition numbers of only those blocks with $\bullet$ elements. In view of this, computing a QR decomposition of X so that the relevant blocks of X_1 are as well-conditioned as possible even if X is ill-conditioned, seems quite attractive. We will therefore aim to hide any possible closeness of X to rank deficiency, or if this is impossible, we will attempt to associate any relatively small diagonal elements of X_1 with real eigenpairs, so that they are cancelled (see equation (11)). To this end, we may compute the desired QR decomposition of X in the following two-stage process:

In the first stage we compute the QR decomposition of X with *minimum* norm column pivoting. Unlike the maximum norm column pivoting which is quite successful in revealing the rank of a matrix (see for example [9] p. 233), the minimum norm column pivoting should have the opposite effect. Consider for example the well known $n\times n$ matrix

$$X=\begin{pmatrix} 1 & -\gamma & -\gamma & \cdots & -\gamma \\ & \sigma & -\gamma & \cdots & -\gamma \\ & & \sigma^2 & \cdots & -\gamma \\ & & & \ddots & \vdots \\ & & & & \sigma^{n-1} \end{pmatrix}, \qquad (16)$$

where $\gamma^2+\sigma^2=1$. If we take $n=10$ and $\gamma=0.7$ then $\chi(X)\approx 10^7$. QR with minimum norm pivoting however, does not change X and since, for the given case, the smallest diagonal element of X is $\sigma^{n-1}\approx 0.048$, it appears to be fine for our intent. Note that maximum norm column pivoting produces a triangular matrix with smallest diagonal element $\approx 10^{-7}$. It is worth pointing out at this point, that since some column pairs of X span 2-dimensional eigenspaces corresponding to complex conjugate eigenvectors, it is desirable for these

columns to remain in consecutive positions if complex arithmetic is to be avoided. Column pivoting however may destroy this structure. Therefore if such a column is to be relocated, its companion column should follow as well. This can be accomplished (from a data structure point of view), if instead of pivoting, the columns in-between are shifted one position eastward.

In the second stage we check the magnitudes of the diagonal elements of X_1, looking for an element smaller than a specific tolerance. If such an element is found outside the $m \times m$ south-east diagonal block and it is associated with a real eigenvalue, we do nothing. If however it is associated with a complex eigenvalue, we find the columns which the current column is linearly dependent with, within the given tolerance. If one of these columns is associated with a real eigenvalue then we make an exchange similar to the one in the first stage, and we update the slightly distorted upper triangular form of X_1. As a result the small diagonal element is now associated with a real eigenvalue, and thus it will be cancelled out. If a small diagonal element is found within the $m \times m$ south-east diagonal block we proceed as above, in an attempt to bring it outside the $m \times m$ block and to associate it with a real eigenvalue.

Next we give a numerical example to demonstrate the performance of the new algorithm, equipped with the "heuristics" we just described. The accuracy of the computed solution will be compared with that of the three-step approach given at the beginning of this section. Consider the case $n = 10$, $m = 2$ and take X as in (16) with $\gamma = 0.7$. Compute random B, E and L of appropriate forms, and compute A so that it satisfies the equation $AX = EXL$. Then from $(A - BF)X = EXL$ obviously $F = O$. The choice of X suggests that the three-step approach should demonstrate a loss of accuracy, up to seven significant decimal digits (recall that $\chi(X) \approx 10^7$). Using MATLAB (uses an accuracy of approximately 16 significant decimal digits) on a Zenith Notebook (Z-NOTE 325L) which has a 387-math coprocessor equipped with the IEEE floating point standard of arithmetic, the new algorithm produced an F with $\|F\|_2 \approx 10^{-15}$, whereas the three-step approach produced an F with $\|F\|_2 \approx 10^{-10}$. The new algorithm performed as well as possible within MATLAB's accuracy, whereas the three-step approach demonstrated an unnecessary loss of five significant digits of accuracy, which clearly indicates numerical instability. The MATLAB programs implementing the new algorithm are part of the MATLAB package PolePack developed by the author. PolePack may be obtained via *anonymous ftp* (see footnote in front page).

V. Conclusion and Future Work

We have pointed out, through reference to the literature, that the solution of the matrix equation $(A - BF)X = EXL$ with respect to F, is (or can be) a key point in eigenstructure assignment. We presented an algorithm for its solution and we demonstrated theoretically as well as experimentally that the new algorithm is superior to a widely used three-step approach. A rigorous rounding error analysis of the new algorithm is under investigation. This will be presented in our future work along with the application of the algorithm to the Partial Eigenstructure Assignment problem as well as to a selection of other Robust Control problems.

References

[1] A. T. Alexandridis, G. D. Galanos, "Optimal Pole-Placement for Linear Multi-Input Controllable Systems", IEEE Trans. on Circuits and Systems, Vol. 34, pp. 1602-1604, 1987.

[2] M. Arnold, B. N. Datta, "An Algorithm for the Multi-Input Eigenvalue Problem", IEEE Trans. on Autom. Control, Vol. 35, pp. 1149-1152, 1990.

[3] R. K. Cavin III, S. P. Bhattacharyya, "Robust and Well-Conditioned Eigenstructure Assignment via Sylvester's Equation", Optimal Control Applications and Methods, Vol. 4, pp. 205-212, 1983.

[4] E. K.-W. Chu, "Controllability of Descriptor Systems", Int. J. Control. Vol. 46, pp. 1761-1770, 1987.

[5] G. R. Duan, "Solutions of the Equation $AV + BW = VF$ and their Application to Eigenstructure Assignment in Linear Systems", IEEE Trans. on Autom. Control, Vol. 38, pp. 276-280, 1993.

[6] M. M. Fahmy, J. O'Reilly, "On Eigenstructure Assignment in Linear Multivariable Systems", IEEE Trans. on Autom. Control, Vol. 27, pp. 690-693, 1982.

[7] M. M. Fahmy, H. S. Tantawy, "Eigenstructure Assignment via Linear State-Feedback Control", Int. J. Control, Vol. 40, pp. 161-178, 1984.

[8] L. R. Fletcher, J. Kautsky, N. K. Nichols, "Eigenstructure Assignment in Descriptor Systems", IEEE Trans. Autom. Control, Vol. 31, pp. 1138-1141, 1986.

[9] G. H. Golub, C. F. Van Loan, "Matrix Computations", Johns Hopkins University Press (2nd Edition) 1989.

[10] J. Kautsky, N. K. Nichols, P. VanDooren, "Robust Pole Assignment in Linear State Feedback", Int. J. Control. Vol. 41, pp. 1129-1155, 1985.

[11] J. Kautsky, N. K. Nichols, E. K.-W. Chu, "Robust Pole Assignment in Singular Control Systems", Linear Algebra and its Applications, Vol. 121, pp. 9-37, 1989.

[12] G. Klein, B. C. Moore, "Eigenvalue Generalized Eigenvector Assignment with State Feedback", IEEE Trans. on Autom. Control, Vol. 22, pp. 140-141, 1977.

[13] G. S. Miminis, C. C. Paige, "A direct algorithm for pole assignment of time-invariant multi-input linear systems using state feedback", Automatica, Vol. 24, pp. 343-356, 1988.

[14] G. S. Miminis, "Deflation in Eigenvalue Assignment of Descriptor Systems using State Feedback", IEEE Trans. Autom. Control, Vol. 38, pp. 1322-1336, 1993.

[15] G. S. Miminis, "Eigenstructure Assignment in Robust Control", in preparation.

[16] B. C. Moore, "On the Flexibility Offered by State Feedback in Multivariable Systems Beyond Closed Loop Eigenvalue Assignment", IEEE Trans. Autom. Control, Vol. 21, pp. 689-692, 1976.

[17] R. V. Patel, P. Misra, "Numerical algorithms for eigenvalue assignment by state feedback", Proc. IEEE, Vol. 72, pp. 1755-1764, 1984.

[18] P. Petkov, N. Christov, M. Konstantinov, "A computational algorithm for pole assignment of linear multi-input systems", IEEE Trans. Autom. Control, Vol.31, pp. 1044-1047, 1986.

[19] S. Pradhan, V. J. Modi, M. S. Bhat, A. K. Misra, "Matrix Method for Eigenstructure Assignment: The Multi-Input Case with Application", AIAA Journal of Guidance, Control and Dynamics, Vol. 17, pp. 983-989, 1994.

[20] G. Roppenecker, "On Parametric State Feedback Design", Int. Journal of Control, Vol. 43, pp. 793-804, 1986.

[21] V. Sinswat, F. Fallside, "Eigenvalue Eigenvector Assignment by State-Feedback", Int. J. Control, Vol. 26, pp. 389-403, 1977.

[22] V. L. Syrmos, F. L. Lewis, "Robust Eigenvalue Assignment for Generalized Systems", Automatica, Vol. 28, pp. 1223-1228, 1992.

[23] C. C. Tsui, "On the Solution to Matrix Equation $TA - FT = LC$ and its Applications", SIAM J. Matrix Anal. Appl., Vol. 14, pp. 33-44, 1993.