A BLOCK ALGORITHM FOR MULTI-INPUT EIGENVALUE ASSIGNMENT

João B Carvalho [*,1]
Biswa Nath Datta, Fellow of IEEE [**,2]

[*] *Department of Mathematical Sciences*
Northern Illinois University, USA.
email: carvalho@math.niu.edu
[**] *Department of Mathematical Sciences*
Northern Illinois University, USA.
email: dattab@math.niu.edu , URL:
http://www.math.niu.edu/˜dattab

Abstract: We present a new block algorithm for the well-known multi-input eigenvalue assignment(EVA) problem. The algorithm has two important features : first, it avoids use of complex arithmetic and is guaranteed to give a real feedback matrix ; and the second, it is suitable for high performance computation. The second feature is the most distinctive one, since high performance algorithms for control problems are rare. The algorithm is also more efficient than the existing well-known algorithms for the problem.

Keywords: pole-assignment, feedback, Arnoldi-method

1. INTRODUCTION

The eigenvalue assignment (or the so-called pole-assignment) problem (EVA) is a well-known problem in control and systems literature. The multi-input EVA problem is defined as follows:

Given a controllable pair of matrices (A, B), $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, and a set $\mathcal{S} = \{\lambda_1, \ldots, \lambda_n\}$, closed under complex conjugation, find a feedback matrix $K \in \mathbb{R}^{m\times n}$ such that $\Omega(A - BK) = \mathcal{S}$. Here $\Omega(M)$ denotes the spectrum of the matrix M.

Several numerically viable algorithms for this important problem have been developed in recent years. The recursive algorithm(Arnold and Datta, 1990), is the most efficient one. The implicit and explicit QR (Patel and Misra, 1984; Miminis and Paige, 1988) and the eigenvector (P.Hr. Petkov and Konstantinov, 1986) algorithms are more expensive than the recursive algorithm. The Schur algorithm (Varga, 1981), and the robust eigenvalue assignment algorithms (J. Kautsky and VanDooren, 1985; Tits and Yang, 1996) are the most expensive ones, but they do more than just assigning eigenvalues. The Schur algorithm can be used as a partial eigenvalue assignment algorithm in the sense that it allows users to assign only a part of the spectrum, which is considered to be "bad" , leaving the other "good" unchanged. The robust eigenvalue assignment algorithms not only assign the eigenvalues, but they find feedback matrices K in such a way that some measures of the sensitivity of the closed-loop eigen-problem are minimized. The Sylvester equation based algorithm of (L.H. Keel and Bhattacharyya, 1985) minimizes the norm of the feedback matrix K and

[1] Author partially supported by Brazilian CAPES and NSF grant ECS-0074411.
[2] The corresponding author. Author partially supported by NSF grant under contract no. ECS-0074411 and by project FONDECYT # 7990011/1999.

that of (Varga, 2000) minimizes simultaneously the norm and the condition number of the matrix K (with respect to the Frobenius norm).

In this paper, we present a new algorithm for the multi-input problem. The algorithm has two important features. First, *the algorithm is guaranteed to give a real feedback matrix K.* In this context, it is noted that some of the above algorithms, such as the recursive and the explicit QR algorithms, might give complex feedback . Second, the most distinctive one, is that the algorithm is a block algorithm; it is composed of BLAS-3 operations such as matrix-matrix products, solutions of linear systems with multiple right hand sides and QR factorizations. *Thus, it is suitable for high performance computing.*

Unfortunately, block algorithms are rare in control. This is particularly true for the eigenvalue assignment problem, as most of the algorithms listed above are level-2 based algorithms (matrix-vector operation based). Only recently block algorithms for state estimation and functional observers have been proposed in the literature (C. Bischof and Purkayastha, 1996; Datta and Sarkissian, 2000).

Besides having the above two features, the *proposed algorithm is more efficient than the recursive algorithm, the implicit and explicit QR algorithms and the eigenvector algorithm.* The other algorithms mentioned in the above list, as stated earlier, are more expensive than the recursive QR and the eigenvector algorithms.

At present, there exists only MATLAB implementation of the algorithm and the results of our numerical experiments have been reported in section 4. The performance results on high performance computers using LAPACK (E. Anderson, 1995; J. Dongarra, 1998) should be available soon.

2. ORTHOGONAL REDUCTION TO CONTROLLER HESSENBERG FORM BY BLOCK ARNOLDI-TYPE METHOD.

Given a pair of matrices (A, B), there always exists an orthogonal matrix V such that

$$V^T AV = H = \begin{bmatrix} H_{11} & H_{12} & H_{13} & \dots & H_{1k} \\ H_{21} & H_{22} & H_{23} & \dots & H_{2k} \\ 0 & H_{32} & H_{33} & \dots & H_{3k} \\ \dots & \dots & \ddots & \dots & \dots \\ 0 & \dots & \dots & H_{k,k-1} & H_{kk} \end{bmatrix} (1)$$

$$\tilde{B} = V^T B = \begin{bmatrix} B_{11} \\ 0 \\ \dots \\ 0 \end{bmatrix} (2)$$

where $H_{ij} \in \mathbb{R}^{n_i \times n_j}$, $n_i = rank(n_{i,i-1})$, $i = 2, 3, \dots, k$. The matrices $H_{i,i-1}, i = 2, \dots, k$ and B_{11} are of the form $(0, R)$ with R upper triangular and nonsingular, except possibly the last block $H_{k,k-1}$, which is zero if the system is uncontrollable.

The pair $(H, \tilde{B})$ is called a *controller Hessenberg* pair and *provides a numerically viable test for controllability.* A well-known numerically viable algorithm to compute the controller- Hessenberg pair is the *staircase* algorithm (Boley, 1981; Miminis and Paige, 1988) .

The subdiagonal blocks $H_{i,i-1}, i = 2, \dots, k$ of the Hessenberg matrix H obtained by the stair-case algorithm are of different sizes, in general. In our development, we, however, need them to be of the same size. We achieve this by using the block-Arnoldi method on the pair (A, B). The traditional block-Arnoldi method computes the block-Hessenberg form of a matrix A by orthogonal similarity transformations (Saad, 1992). We adapt this procedure to the pair (A, B), to achieve a block-Hessenberg pair $(H, \bar{B})$ in the form we need.

We use the modified Gram-Schmidt version, which is known to be more numerically stable than the classical Gram-Schmidt version (Datta, 1995; Golub and VanLoan, 1996).

Algorithm 2.1 : A block Arnoldi method for Controller-Hessenberg Form.

Inputs: controllable pair (A, B), where $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$.

Outputs: (i)H - a block upper Hessenberg matrix with upper triangular subdiagonal blocks of equal size

(ii) $\{V_i\}_{i=1,\dots,k}$- a orthonormal set of block column vectors such that

$$V_i^T V_j = 0 (i \neq j), \quad V_i^T V_i = I_{p \times p} (i = 1, \dots, k-1),$$

$$and\ V^T B = \tilde{B} = \begin{bmatrix} B_{11} & 0 \end{bmatrix}^T,$$

where $V = \begin{bmatrix} V_1 & V_2 & \dots & V_{k-1} & V_k \end{bmatrix}$ and B_{11} is nonsingular upper triangular.

Step 1: Find the thin QR factorization of B (Datta, 2001; Golub and VanLoan, 1996): $B = V_1 R_1$, where $V_1 \in R^{n \times p}$ has orthonormal columns, $R_1 \in \mathbb{R}^{p \times m}$ is upper triangular, where $p = rank(B)$.

Step 2: Define k to be the positive integer such that $n \leq pk < n + p$.

Step 3: For $j = 1, 2, \dots, k$ do

3.1: Compute $W_j = AV_j$

3.2: For $i = 1, 2, \dots, j$ do

Compute $H_{ij} = V_i^T W_j$

Update $W_j = W_j - V_j H_{ij}$

End For

3.3: If $j < k$, compute the thin QR factorization $W_j = V_{j+1} H_{j+1,j}$.

End For

Step 4: Form the matrices $V = \begin{bmatrix} V_1 & \dots & V_k \end{bmatrix}$ and $H = (H_{ij})$ from the computed blocks.

Step 5: Resize $V = V(1:n, 1:n)$, $H = H(1:n, 1:n)$.

Return

Flop Count: We ommit the details of the flop-count here. Counting one multiplication/ division or one addition/subtraction as a floating point operation, and using Housolder's method for QR factorization, it is easy to see that Algorithm 2.1 requires roughly $4n^3 + n^2 \left(\frac{k}{2} + 4p\right)$ flops (assuming $kp \approx n$) .

3. A RECURSIVE BLOCK ALGORITHM FOR MULTI-INPUT EVA.

Given the controller Hessenberg pair $(H, \tilde{B})$, defined by (1) and (2), of the controllable pair (A, B), and the set $\mathcal{S} = \{\lambda_1, \dots, \lambda_n\}$, closed under complex conjugation, the algorithm simultaneously constructs a block nonsingular matrix $L = \begin{bmatrix} L_1 & L_2 & \dots & L_n \end{bmatrix}$, computed block columnwise, and a block tridiagonal matrix Λ, with diagonal blocks containing the eigenvalues to be assigned:

$$\Lambda = \begin{bmatrix} \Lambda_{11} & \Lambda_{12} & & & & & \\ \Lambda_{21} & \Lambda_{22} & \Lambda_{23} & & & & \\ & \Lambda_{32} & \Lambda_{33} & \Lambda_{34} & & & \\ & & \Lambda_{43} & \Lambda_{44} & & & \\ & & & \ddots & \ddots & & \\ & & & & & & \Lambda_{k-1,k} \\ & & & & & \Lambda_{k,k-1} & \Lambda_{kk} \end{bmatrix} \quad (3)$$

in such a way that L and Λ satisfy the matrix equation

$$H^T L - L\Lambda = F^T \tilde{B}^T L. \quad (4)$$

The feedback matrix F is easily computed from the above equation, once L and Λ are computed. Each $\Lambda_{ii}, i = 2, \dots, k$ has the same order p of $H_{i+1,i}$, $i = 1, 2, \dots, k-2$ and the order of Λ_{11} is $\tilde{p}$, the same as the order of H_{kk}. We remark that $p = m$, the number of columns of B, if B is full rank. From the construction of L (see below), it follows that L is nonsingular, we then have, from the above equation, that

$$H - \tilde{B}F = (L^T)^{-1} \Lambda^T L^T \quad (5)$$

showing that $\Omega(H - \tilde{B}F) = \mathcal{S}$. Once F is known, the feedback matrix K for the original pair (A, B) is easily recovered from

$$K = FV^T. \quad (6)$$

The above discussion leads to the following algorithm. A careful assingment of the eigenvalues to the diagonal blocks of Λ is important in achieving a real feedback matrix F and the details are shown in the algorithm.

Algorithm 3.1: A recursive Multi-Input EVA Algorithm.

Inputs: A- an $n \times n$ real matrix

B - an $n \times m$ real matrix ($m \leq n$).

$\mathcal{S}$ - a set of n complex numbers, closed under complex conjugation.

Output: The feedback matrix K such that $\Omega(A - BK) = \mathcal{S}$.

Assumption: (A, B) is controllable.

Step 1: Transform the pair (A, B) to controller-Hessenberg pair using Algorithm 2.1:

$V^T AV = H$, a block unreduced upper Hessenberg matrix

and $V^T B = \tilde{B} = \begin{bmatrix} R \\ 0 \end{bmatrix}$

where R is upper triangular with rank p.

Step 2: Set $k = \lceil n/p \rceil$ and find $\tilde{p}$ such that $\tilde{p} + (k-1)p = n$.

Step 3: Define $D = (d_{ij}) \in \mathbb{R}^{p \times p}$ by $d_{ij} = 1, i + j = p + 1; d_{ij} = 0$, otherwise. Set $L_1 = \begin{bmatrix} 0 & \dots & 0 & D \end{bmatrix}^T$. Set $\Lambda_{i,i+1} = 0 \in \mathbb{R}^{p \times p}, i = 1, 2, \dots, k-2$ and $\Lambda_{k-1,k} = 0 \in \mathbb{R}^{p \times \tilde{p}}$.

Step 4: For $i = 1, 2, \dots, k$ do

Step 4.1: Define $s_i \in \mathbb{C}^p$ to be the $i-th$ segment of the set $\mathcal{S}$. Consider two cases:

Case 1: Step 4.2: If the last assigned subset of $\mathcal{S}$ was closed under conjugation, do Steps 4.2.1 or 4.2.2, and do Step 4.4.

4.2.1: If s_i is closed under conjugation, allocate the numbers in s_i to Λ_{ii}; the real numbers on the main diagonal and the complex conjugate pairs on the 2×2 diagonal blocks.

4.2.2: If s_i is not closed under complex conjugation, allocate the first $p-1$ members of s_i

to the places $\Lambda_{ii}(1:p-1,1:p-1)$ and allocate the real part α of the $p-th$ member of s_i to $\Lambda_{ii}(p,p)$.

Case 2: Step 4.3: If the last assigned subset of $\mathcal{S}$ was not closed under conjugation, define a and b by $a-\mathbf{i}b = s_i(1)$ and do Steps 4.3.1 - 4.3.3 and Step 4.4:

4.3.1: Compute m_{12} and m_{22} in the matrix $M = \begin{bmatrix} \alpha & m_{12} \\ \beta & m_{22} \end{bmatrix}$ such that $\Omega(M) = \{a-\mathbf{i}b, a+\mathbf{i}b\}$, where $\alpha = \Lambda_{i-1,i-1}(p,p)$ and $\beta = \Lambda_{i,i-1}(1,p)$.

4.3.2: Allocate m_{12} to $\Lambda_{i-1,i}(p,1)$ and m_{22} to $\Lambda_{i,i}(1,1)$.

4.3.3: Allocate $\tilde{s}_i = s_i(2:p)$ to $\Lambda_{ii}(2:p,2:p)$ as in Step 3.3.1 or Step 3.2.2, whatever may be appropriate.

Step 4.4: If $i < k$ compute $W_i = H^T L_i - L_i\Lambda_{ii} - L_{i-1}\Lambda_{i-1,i}$ and compute the thin QR factorization: $W_i = L_{i+1}\Lambda_{i+1,i}$ exploiting the structure of the blocks L_i (see Remark 3 below).

End For

Step 5: Let

$\hat{L} = \left(L_{k-1}(:,\tilde{p}+1:p)) \; L_k(:,1:\tilde{p}) \right)$, $\hat{W} = \begin{pmatrix} \Lambda_{k-1,k-1}(\tilde{p}+1:p,\tilde{p}+1:p) & \Lambda_{k-1,k}(\ldots) \\ \Lambda_{k,k-1}(\ldots) & \Lambda_{kk} \end{pmatrix}$

and compute

$$W_k = H^T\hat{L} + \hat{L}\hat{W}$$

Step 6: Set $F = 0 \in \mathbb{R}^{m\times n}$ and solve the triangular linear system for $F(1:p,:)$:

$$[\hat{L}(1:p,:)]^T RF(1:p,:) = W_k^T.$$

Step 7: Recover the feedback matrix K by forming the matrix- matrix product $K = FV^T$.

Return

Remarks:

(1) If $p=1$, then we have single-input assignment and certain steps in the algorithm, which are not appropriate, must be eliminated or properly adjusted.
(2) If $\tilde{p} < p$ then L_k must be resized properly (see Example 4.1).
(3) It can be shown that the blocks L_i, $i = 1,2,\ldots,k-1$ will always have zeros in its upper part, more precisely, $L_i(1:n-ip,:) = 0$ and therefore $W_i(1:n-(i+1)p,:) = 0$. Thus, some work can be saved in Step 3.4, either in the matrix multiplications or in the skinny QR factorization.
(4) It can be shown that $\left[L_{k-1}(1:p,\tilde{p}+1:p) \; L_k(1:p,:) \right]$ will be lower triangular with respect to its anti-diagonal. Therefore this structure can be conveniently exploited in the solution of the linear system in Step 6.
(5) In Step 4.3.1, if $u=0$ or if $|u|$ is small, we can go back to Step 4.2.2 and reallocate to $\Lambda_{i-1,i}$ a different number than the real part of the complex element $s_{i-1}(p)$, then recompute W_{i-1}, Λ_i and $\Lambda_{i,i-1}$ and return to Step 4.3 to get a better value of u, so that the assignment in Step 4.3.1 can be done with higher numerical accuracy (Varga, 1981).

Flop Count: Algorithm 3.1 requires approximately $\dfrac{14n^3}{3} + n^2\left(\dfrac{k}{2}+11p\right)$ flops. This count includes the orthogonal reduction to controller-Hessenberg form by Block Arnoldi method (Algorithm 2.1). The Householder method for QR factorization (Datta, 1995; Golub and VanLoan, 1996) is used in the implementation.

Step 1: Reduction to Controller-Hessenberg form

$$4n^3 + n^2\left(\frac{k}{2}+4p\right) \text{ flops}$$

Step 4.4 and Step 5: Neglecting the flop count for the product $L_{i-1}\Lambda_{i-1,i}$ (since $\Lambda_{i-1,i}$ has only one nonzero element), using $pk \approx n$ and regarding L_i as a $ip\times p$ nonzero matrix and W_i as a $(i+1)p\times p$ nonzero matrix, Step 4.4 and Step 5 require

$$\sum_{i=1}^{k}\left(i+1)p^2(2ip) + (ip^2)(2p) + 4(i+1)p(p^2)\frac{4p^3}{3}\right) \approx \sum_{i=1}^{k} 2p^3 i(i+1) + 6p^3 i = 2p^3\sum_{i=1}^{k}(i^2 + 4i) = 2p^3\left(\frac{k^3}{3}+\frac{5k^2}{2}+\frac{13k}{6}\right) \approx \frac{2n^3}{3} + 5n^2p \text{ flops}$$

Step 6: Solution of linear system with dimension p and n right hand sides: $n\cdot p^2$ flops

Step 7: Computation of K : $2n^2p$ flops

Total (approximately):

$$4n^3 + n^2\left(\frac{k}{2}+4p\right) + \frac{2n^3}{3} + 5n^2p + np^2 + 2n^2p \approx \frac{14n^3}{3} + n^2\left(11p+\frac{k}{2}\right) \text{ flops}$$

Comparision of Efficiency with other algorithms.

The recursive algorithm, the explicit and implicit QR algorithms, the eigenvector algorithm require, respectively, $19n^3/3$, $9n^3$, $9n^3$ and $40n^3/3$ flops (Datta, 2001). Thus, *the proposed algorithm is faster than all the above algorithms.*

Furthermore, within the above operations, the matrix L that transforms the reduced Hessenberg closed-loop system to the block matrix Λ by similarity, is obtained.

4. A NUMERICAL EXAMPLE AND COMPARISONS.

In this section we present a numerical example to illustrate the main algorithm (Algorithm 3.1). As expected, the obtained feedback is real.

We also plot the zero-input responses of the two systems

$$\dot{x}_1(t) = (A - BK_1)x_1(t) \qquad (7)$$
$$\dot{x}_2(t) = (A - BK_2)x_2(t) \qquad (8)$$

where K_1 and K_2 were obtained, respectively, by the proposed algorithm and the explicit QR algorithm (Miminis and Paige, 1988) , with initial condition $x_1(0) = x_2(0) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix}^T$.

Example 4.1: $n = 5$, $p = 2$. Let

$$A = \begin{pmatrix} -8.99 & 9.80 & 9.20 & 3.67 & 2.60 \\ -1.69 & 5.78 & 4.53 & -5.75 & -2.59 \\ -3.90 & -1.23 & -1.76 & 6.78 & 1.50 \\ 7.49 & -0.03 & 4.89 & 2.58 & -0.97 \\ -9.70 & -5.72 & -4.64 & -7.32 & -9.12 \end{pmatrix},$$

$$B = \begin{pmatrix} 9.08 & 6.41 \\ 2.32 & 1.91 \\ 2.39 & 8.44 \\ 0.50 & 1.74 \\ 0.78 & 1.71 \end{pmatrix}$$

$$\mathcal{S} = \{-1, -2-\mathbf{i}, -2+\mathbf{i}, -3+\mathbf{i}, -3-\mathbf{i}\}.$$

Step 1. The reduction to the controller Hessenberg form, using Algorithm 2.1, gives

$$H = \begin{pmatrix} -4.6334 & 11.7586 & 1.7960 & 5.9595 & -9.8077 \\ -3.0787 & -2.3902 & 3.0875 & 2.0430 & 5.9449 \\ 14.9965 & 5.6554 & -0.2563 & 6.9984 & 1.3222 \\ 0 & -0.9322 & 1.6141 & -10.0297 & -2.5196 \\ 0 & 0 & 2.1119 & 1.9852 & 5.7996 \end{pmatrix},$$

$$V = \begin{pmatrix} -0.9345 & 0.2623 & -0.0081 & 0.1239 & -0.2059 \\ -0.2388 & 0.0266 & -0.0957 & -0.4108 & 0.8743 \\ -0.2460 & -0.9335 & -0.0074 & 0.2490 & 0.0774 \\ -0.0515 & -0.1915 & -0.6054 & -0.6663 & -0.3876 \\ -0.0803 & -0.1496 & 0.7901 & -0.5567 & -0.1924 \end{pmatrix},$$

$$V^T B = \begin{pmatrix} R \\ 0 \end{pmatrix} = \begin{pmatrix} -9.7159 & -8.7495 \\ 0 & -6.7356 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

Step2 - Step 4 gives

$$L = \begin{pmatrix} 0 & 0 & 0 & 0 & -0.4858 \\ 0 & 0 & 0 & -0.1222 & -0.1548 \\ 0 & 0 & -0.2857 & 0.3619 & -0.3259 \\ 0 & 1 & -0.2686 & -0.9113 & -0.1610 \\ 1 & 0 & -0.9199 & 0.1537 & -0.7797 \end{pmatrix}$$

$$\Lambda = \begin{pmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & -2 & -0.2492 & 0 & 0 \\ -7.3916 & 4.0132 & -2 & 0 & 0 \\ 0 & 7.6284 & 0 & -3 & 0.1392 \\ 0 & 0 & 8.8194 & -7.1826 & -3 \end{pmatrix}.$$

Step 5. Computing W_3:

$$W_3 = H^T \begin{pmatrix} 0.0000 & -0.4858 \\ -0.1222 & -0.1548 \\ 0.3619 & -0.3259 \\ -0.9113 & -0.1610 \\ 0.1537 & -0.7797 \end{pmatrix} + \begin{pmatrix} 0.0000 & -0.4858 \\ -0.1222 & -0.1548 \\ 0.3619 & -0.3259 \\ -0.9113 & -0.1610 \\ 0.1537 & -0.7797 \end{pmatrix} \begin{pmatrix} -3.0000 & 0.1392 \\ -7.1826 & -3.0000 \end{pmatrix} = \begin{pmatrix} 2.3140 & -3.6175 \\ 1.7096 & -7.4832 \\ -2.8718 & -4.2016 \\ 7.8379 & -5.7816 \\ -2.1995 & -3.0629 \end{pmatrix}$$

Step 6. Solving the triangular system

$$\begin{pmatrix} 0.0000 & -45.8616 \\ -5.5438 & -76.9548 \end{pmatrix}^T RF = W_3^T$$

gives

$$F = \begin{pmatrix} -27.9021 & -27.8699 & 21.5198 & -84.7490 & 16.7164 \\ 20.0151 & 14.7877 & -24.8397 & 67.7950 & -19.0251 \end{pmatrix}$$

Step 7. Compute

$$K_1 = \begin{pmatrix} 0.6525 & 7.4856 & 1.8137 & 6.1434 & 9.4637 \\ -0.3238 & -6.5305 & -0.4404 & -3.7397 & -8.0802 \end{pmatrix}$$

Verification:

$$\Omega(A - BK_1) = \left\{ \begin{matrix} -1.00000000000043 \\ -3.00000000000019 + 0.99999999999984i \\ -3.00000000000019 - 0.99999999999984i \\ -1.99999999999955 + 0.99999999999982i \\ -1.99999999999955 - 0.99999999999982i \end{matrix} \right\}$$

The explicit QR algorithm by Miminis and Paige, when run on the above example produced the feedback matrix:

$$K_2 = \begin{pmatrix} -0.4976 & 2.6069 & 1.0614 & 1.9788 & 9.9367 \\ 0.5992 & -2.0978 & 0.2006 & -0.1562 & -8.5216 \end{pmatrix}.$$

which verifies

$$\Omega(A - BK_2) = \left\{ \begin{matrix} -0.99999999997130 \\ -1.99999999999616 + 1.00000000003781\mathbf{i} \\ -1.99999999999616 - 1.00000000003781\mathbf{i} \\ -3.00000000001819 + 0.99999999994485\mathbf{i} \\ -3.00000000001819 - 0.99999999994485\mathbf{i} \end{matrix} \right\}.$$

We now plot the norms of the zero-input responses with the two feedbacks. The solid line in Figure 1 corresponds to feedback K_1, and the dotted line in Figure 2 corresponds to feedback K_2.

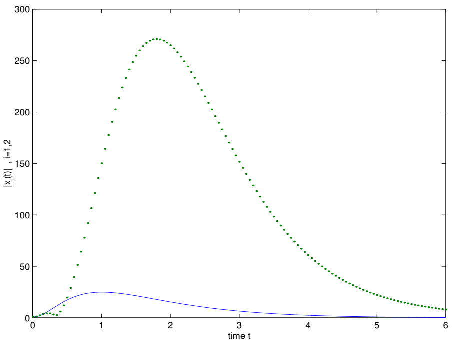

Fig. 1. Transient responses due to K_1 and K_2.

The plots show that the largest norm in transient due to K_2 is much larger than the one due to K_1.

5. CONCLUSION

A new block algorithm for the multi-input eigenvalue assignment problem has been presented in this paper. The proposed algorithm is suitable for high performance computing and is guaranteed to give a real feedback matrix. These features are distinctive because high performance algorithms in control are rare and several existing methods for the multi-input EVA problem might give feedback matrices with complex entries.

The algorithm is faster than the existing ones for this problem. The algorithm is composed of numerically viable computations such as reduction to block Hessenberg form, solution of block linear systems and QR factorizations. The algorithm will soon be implemented in some of the today's high performance computers using LAPACK and the performance results will be reported at the conference.

Future research includes numerical reliability analysis of the algorithm and implementations of the algorithm on practical control problems. Also, a comparative study of the algorithm, and the other well-known existing ones for the problem, with respect to computer timing, numerical effectiveness and accuracy is also in order.

6. REFERENCES

Arnold, M. and B.N. Datta (1990). An algorithm for the multi input eigenvalue assignment problem. *IEEE Trans. Automat. Contr.* **35**, no. 10, 1149–1152.

Boley, D.L. (1981). Computing the controllability/observability decomposition of a linear time-invariant dynamic system, a numerical approach, phd. thesis. *Report STAN-CS-81-860, Dept. of Computer Science, Stanford University, Stanford, CA.*

C. Bischof, B.N. Datta and A. Purkayastha (1996). A parallel algorithm for the sylvester observer equation. *SIAM J. Sci. Computing* **17**, 686–698.

Datta, B.N. (1995). *Numerical Linear Algebra and Applications.* Brooks/Cole Publishing Company. Pacific Grove.

Datta, B.N. (2001). *Numerical Methods for Control Systems Design and Analysis.* Academic Press. to appear.

Datta, B.N. and D. Sarkissian (2000). Block algorithms for state estimation and functional observers. *Proc. IEEE Internation Symposium on Computer-Aided Control System Design* pp. 19–23.

E. Anderson, et al (1995). *LAPACK User's Guide.* SIAM. Philadelphia, PA.

Golub, G. and C. VanLoan (1996). *Matrix Computations, 3rd. ed.*. The Johns Hopkins Univ. Press.

J. Dongarra, et al (1998). *Numerical Linear Algebra for High-Performance computers.* SIAM. Philadelphia, PA.

J. Kautsky, N.K. Nichols and P. VanDooren (1985). Robust pole assignment in linear state feedback. *Int. J. Control* **41**, no. 5, 1129–1155.

L.H. Keel, J.A. Fleming and S.P. Bhattacharyya (1985). Pole assignment via sylvester's equation. *Contemp. Math.* **47**, 265–265.

Miminis, G.S. and C.C. Paige (1988). A direct algorithm for pole assignment of time-invariant multi-input linear systems using state feedback. *Automatica* **24**, no. 3, 343–356.

Patel, R.V. and P. Misra (1984). Numerical algorithms for eigenvalue assignment by state feedback. *Proc. IEEE* **72**, no. 12, 1755–1764.

P.Hr. Petkov, N.D. Christov and M.M. Konstantinov (1986). A computational algorithm for pole assignment of linear multi input systems. *IEEE Trans. Automat. Contr.* **AC-31**, no. 11, pp 1044–1047.

Saad, Y. (1992). *Numerical Methods for Large Eigenvalue Problems.* Wiley. New York.

Tits, A.L. and Y. Yang (1996). Globally convergent algorithms for robust pole assignment by state feedback. *IEEE Trans. Automat. Control* **41**, 1432–1452.

Varga, A. (1981). A schur method for pole assignment. *IEEE Trans. Automat. Control* **AC-26**, no. 2, pp 517–519.

Varga, A. (2000). Robust pole assignment for descriptor systems. *Proc. MTNS'2000 Symposium, Perpignan, France.*