NUMERICAL ALGORITHMS FOR CONTROLLABILITY AND EIGENVALUE ALLOCATION

by

George S. Miminis
School of Computer Science
McGill University

May 1981

A thesis submited to the Faculty of Graduate Studies and Research, in partial fulfillment of the requirements for the degree of Master of Science.

Algorithms for Controllability and Eigenvalue Allocation

Contents

CHAPTER 4

(I) Conclusion

APPENDIX

REFERENCES

Abstract

As part of this thesis we are concerned with the numerical properties of some methods for computing controllability of a linear dynamic system with constant coefficients. We show, using counterexamples, why some well known methods for determining the controllability of such a system are often numerically unstable or computationally inefficient. We also show that recently developed algorithms by C. Paige [2] and others [17], [18], [19], work satisfactorily for the same examples.

We also give a different algorithm which determines whether a system is controllable or not. The same algorithm with some changes can be used to compute the distance of the given system from the nearest uncontrollable one, but it is pointed out why this algorithm is computationally inefficient.

We also present an algorithm for solving the problem of eigenvalue allocation for a single-input controllable, linear dynamic system with constant coefficients. This algorithm is based on numerically stable transformations.

These latter two algorithms have been developed jointly with C. Paige.

Resumé

Dans cette thèse, nous examinons les propriétés numériques de certaines méthodes pour le calcul de la commandabilité d'un système linéaire dynamique avec coefficients constants. Nous montrons, utilisant des contre-exemples, pourquoi certaines méthodes très connues pour déterminer la commandabilité sont souvent instables numériquement ou inefficaces à calculer. Nous montrons aussi que des algorithmes recemment developpés par C. Paige [2] et autres [17], [18], [19], sont satisfaisant pour les mêmes exemples.

Nous indiquons aussi un algorithme différent qui détermine si un système est commandable ou non. Ce même algorithme avec quelques changements peut être utilisé pour calculer la distance du système donné au système non commandable le plus rapproché, mais on note pourquoi cet algorithme est inefficace.

De même, nous indiquons un algorithme pour résoudre le problème d'allocation des valeurs propres pour un système linéaire dynamique commandable monovariable avec coefficients constants. Cet algorithme est basé sur des transformations numériquement stables.

Ces deux derniers algorithmes ont été developés conjointement avec C. Paige.

Acknowledgements

I would like to thank Professor C. Paige, not only for his invaluable supervision of the work reported here but also for all that I have learned from him about computational linear algebra. It is always a pleasure to work with him and I am grateful to him for his guidance and encouragement during the entire period of the work.

I also thank Diane Chan for typing this thesis.

Chapter 1

(I) Introduction

(1) Overview

C. Paige [2] points out that a measure of the distance of a given dynamic system from the nearest uncontrollable one would be quite useful because it would not only determine whether a system is controllable or not, but it would also indicate, in the case of a controllable system, if the given system is quite controllable or it is nearly uncontrollable. In the same paper it is mentioned that the above problem appears to be an open problem.

This problem and the fact that apparently there are no numerically reliable and efficient algorithms for the problem of eigenvalue allocation, gave us the impetus to write this thesis. The outline of the thesis is as follows.

In the second section of this chapter we establish our notational conventions. In Chapter 2 we first prove three widely used mathematically equivalent theorems. We also prove another theorem; suggested by C. Paige in [2], using a reduction of a matrix to a block upper Hessemberg matrix, also suggested by C. Paige in [2] as a very useful tool. Then we show, using counterexamples, why three well known methods often fail to determine whether a system is controllable or not, while the new algorithm in [2] succeeds in determining the controllability of the systems in the same examples. Finally in the same chapter we give an algorithm which uses the notion of the distance of a system from the nearest uncontrollable one to determine the controllability of the system. As we men-

tioned in the abstract, the same algorithm with some changes can be used for the computation of the distance of a system from the nearest uncontrollable one when the system is controllable, but it is pointed out why this algorithm is not computationally efficient.

In Chapter 3 we present some new algorithms for the eigenvalue allocation problem. First we describe how we can assign a certain desired set of real engenvalues to a real upper Hessenberg matrix by determining its first row, using an explicit method related to the QR algorithm. Then we do the same thing but using an implicit version of the method. A good description of the QR algorithm is given in [12]. Finally we describe how to assign a certain desired set of eigenvalues which occur in complex conjugate pairs to a real upper Hessenberg matrix by determining its first row, using an implicitly shifted method.

In Chapter 4 we comment on the methods which are developed in this thesis.

In the Appendix we present the subroutine EVA which performs the eigenvalue allocation described in Chapter 3. We also present four examples, which indicate that the methods described in Chapter 3 are very efficient and reliable.

(2) Notation

a) Matrices will be represented by upper case Roman letters. Vectors will be represented by lower case Roman. Scalars will be represented by lower case Greek letters, and indices by lower case Roman.

b) A^T will represent the transpose of the matrix A, while A^H will represent the complex conjugate transpose of A.

c) $\mathbb{R}$ represents the set of real numbers, $\mathbb{C}$ the set of complex numbers and $\mathbb{N}$ the set of natural numbers.

d) $\forall \alpha \in A$ means, for every α element of the set A.

e) e_k with $k \in \mathbb{N}$ represents a column vector, the elements of which are all zero except the kth one which is unity, while e will represent a column vector with all its elements equal to one.

f) $A = (\alpha_{ij})$ means that the matrix A has elements α_{ij}.

g) $i(j)h$ will represent the arithmetical progression with first element i, step j and final element h. j can be either positive or negative, so i can be less than or greater than h.

h) $A \cup B$ means the union of the sets A and B.

i) Δ_n, where $n \in \mathbb{N}$ will represent the set of integers in the interval $[1, n]$, that is, $i \in \Delta_n$ if and only if $i \in \mathbb{N}$ and $1 \leq i \leq n$.

j) Let $x = \begin{pmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_n \end{pmatrix}$ be an n-vector. The numbers $\xi_1, \xi_2, \ldots, \xi_n$ are called the components or the elements of x.

k) $\mathbb{R}^n$ represents the set of all n-vectors with real components. Similarly $\mathbb{C}^n$ represents the set of all n-vectors with complex components.

l) $\mathbb{R}^{m \times n}$ (respectively $\mathbb{C}^{m \times n}$) represents the set of all matrices of dimensions m, n and real (respectively complex) elements.

m) Let $x = (\xi_1, \xi_2, \ldots, \xi_n)^T$, then the 2-norm of x is represented by $||x||_2$ and defined by $||x||_2 = \sqrt{\sum_{i=1}^{n} |\xi_i|^2}$.

n) Let A be an $m \times n$ matrix with elements (α_{ij}), then the 2-norm (respectively the F-norm) of A is represented by $||A||_2$ (respectively $||A||_F$) and is defined by $||A||_2 = \max_{||x||_2 \neq 0} \frac{||Ax||_2}{||x||_2}$ (respectively $||A||_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |\alpha_{ij}|^2}$).

p) A subspace R is called trivial if and only if $R = \{0\}$.

q) A permutation matrix is the unity matrix with rearranged columns or equivalently, rows.

e.g. $P = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$

r) Let $A \in \mathbb{R}^{m \times n}$ then $R(A)$ will represent the subspace spanned by the columns of A.

CHAPTER 2

(I) Introduction

(1) Dynamic systems

A system is called dynamic when it changes states with respect to time and the characteristics of the state at one time are related with those at other times. The dynamical systems considered here can be represented mathematically by either difference or differential equations. The choice of difference or differential equation corresponds to whether the system is observed in discrete or continuous time respectively.

(2) Controllability of dynamic systems

The concept of controllability of a dynamical system is introduced by the following question:

<<Can a dynamical system be transferred from any given initial state to any desired state in finite time by some control action?>>

If we answer <<yes>> then the system is controllable.

(II) The algebraic problem of controllability

Consider the continuous-time system

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

where $t \in \mathbb{R}$ $(t \geq 0)$ is the time; $x \in \mathbb{R}^n$ is the state of the system; $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$ and $u \in \mathbb{R}^m$ is the control vector.

Note: Throughout this thesis n will represent the order of matrix A and m the number of columns of B, unless it is mentioned that they represent something else.

The set of all x is a real n-dimensional vector space, called the state space of the system (1) and it will be denoted by R.

Definition (I): The system (1) is said to be controllable if for any given pair of states $(x_I, x_F) \in \mathbb{R}^n \times \mathbb{R}^n$ there exists a time $t > 0$ and a control u defined on $[0,t]$ such that the solution of (1) which corresponds to the initial value $x(0) \equiv x_I$ gives us $x(t) \equiv x_F$.

The state space R can be decomposed into two parts: the controllable subspace R_1 and the uncontrollable subspace R_2 which is the orthogonal complement of R_1 [16]. So, R is generated by $R_1 \cup R_2$ and we write

$$R = R_1 \oplus R_2 \qquad (1\alpha)$$

From the definition (I) we can observe the following: if the system (1) is controllable then $R_2 = \{0\}$. So $R = R_1$. If (1) is uncontrollable then $R_2 \neq \{0\}$ but R_1 need not be trivial.

We shall now define the Discrete-time system

$$x_{k+1} = Ax_k + Bu_k \quad , \quad k \in \mathbb{N} \qquad (2)$$

where the symbols have meaning similar to the previously defined continuous-time system.

Definition (II): The system (2) is called controllable if for any given pair of states $(x_I , x_F) \in \mathbb{R}^n \times \mathbb{R}^n$ there exists a positive integer q and a sequence of controls $u_1, u_2, \ldots, u_q$ such that using (2) we can transfer state $x_1 \equiv x_I$ to state $x_{q+1} \equiv x_F$ in exactly q steps.

Definition (III): Define

$$\bar{\mu} \triangleq \min_{\delta A, \delta B} ||(\delta A, \delta B)||_\zeta , \quad \zeta \in \{2,F\} , \quad \delta A \in \mathbb{R}^{n\times n} , \quad \delta B \in \mathbb{R}^{n\times m} \tag{3}$$

such that the systems

$$\dot{x} = (A + \delta A)x + (B + \delta B)u \tag{4}$$

and

$$x_{k+1} = (A + \delta A)x_k + (B + \delta B)u_k \tag{4α}$$

are uncontrollable, [2], (another approach is used by Moore in [15],[16]).

Remark (I): μ is a useful measure of the distance of the system (1) (respectively (2)) from the nearest uncontrollable system (4) (respectively (4α)).

Remark (II): Since the controllability of a problem depends only on the matrices A and B (as we will see in the following), we conclude that the distance between the systems (1) and (4) and the distance between the systems (2) and (4α) are identical.

Remark (III): If $\mu = 0$ then (1) and (2) are uncontrollable, otherwise $\mu > 0$ and systems (1) and (2) are controllable.

Note: As remark (II) indicates and as we will see in the following the conditions on A and B for controllability are the same for the discrete and continuous system. So in the following if we want to say that system (1) or (2) is controllable or not we will just say that the pair (A,B) is controllable or not. In general, if we want to derive a result which can be applied to both (1) and (2) we can work with the pair (A,B) .

Proposition (I): Applying orthogonal transformations to (1) leaves μ unchanged.

Proof: After applying the orthogonal transformations to the system (1) we get:

$$Q^T\dot{x} = (Q^TAQ)(Q^Tx) + (Q^TBP)(P^Tu) \qquad (4\beta)$$

where $Q \in \mathbb{R}^{n\times n}$, $P \in \mathbb{R}^{m\times m}$ are orthogonal matrices. Then from definition (III) of μ let

$$\tilde{\mu} = \min \|(Q^T\delta AQ,\ Q^T\delta BP)\|_\zeta , \quad \zeta \in \{2,F\} \qquad (5)$$

such that the system

$$Q^T\dot{x} = [Q^T(A + \delta A)Q](Q^Tx) + [Q^T(B + \delta B)P](P^Tu)$$

is uncontrollable. Now we can see that

$$(5) \Rightarrow \tilde{\mu} = \min \left\| Q^T(\delta A,\delta B)\begin{pmatrix} Q & 0 \\ 0 & P \end{pmatrix} \right\|_\zeta$$

$$= \min \|(\delta A,\delta B)\|_\zeta$$

$$= \mu$$

where μ is given in (3).

Corollary (I): The system (1) is controllable if and only if the system

$$\dot{\tilde{x}} = \tilde{A}\tilde{x} + \tilde{B}\tilde{u} \tag{5α}$$

is controllable, where $\tilde{x} = Q^T x$, $\tilde{u} = P^T u$, $\tilde{A} = Q^T A Q$, $\tilde{B} = Q^T B P$ with $Q \in \mathbb{R}^{n\times n}$ and $P \in \mathbb{R}^{m\times m}$ orthogonal matrices. In other words orthogonal transformations do not alter the controllability of (1).

Proof: Obviously (5α) is the same as (4β). So from proposition (I) since the distance of (5α) from the nearest uncontrollable system is the same as the distance of (1) from the nearest uncontrollable system, this implies that corollary (I) is true.

Note: The arguments of proposition (I) and corollary (I) can also be applied for discrete-time systems.

Before we mention theorems about controllability, we will transform the pair (A,B) by orthogonal transformations to another simpler pair $(\tilde{A},\tilde{B})$. This transformation will be very useful in the following when we will use it to prove theorems about controllability.

It can be shown [2] that (A,B) can be transformed, using orthogonal transformations, to $(\tilde{A},\tilde{B})$ where for some positive integer $k \le n$ and $n_0 = m$

$$\tilde{A} = \begin{pmatrix} A_{11} & A_{12} & \cdot & A_{1,k-1} & A_{1k} \\ A_{21} & A_{22} & \cdot & A_{2,k-1} & A_{2k} \\ & \cdot & \cdot & \cdot & \cdot \\ & & \cdot & \cdot & \cdot \\ & & & A_{k,k-1} & A_{kk} \end{pmatrix}, \quad \tilde{B} = \begin{pmatrix} A_{10} \\ 0 \\ \cdot \\ \cdot \\ 0 \end{pmatrix} \tag{6}$$

where $A_{ij} \in \mathbb{R}^{n_i \times n_j}$, $\text{rank}(A_{i,i-1}) = n_i$ for $i = 1(1)k-1$ and $A_{k,k-1}$ is either zero or has rank n_k. By corollary (I) (A,B) is controllable if and only if $(\tilde{A},\tilde{B})$ is controllable. Thus instead of looking for the controllability of (A,B) we will look for the controllability of $(\tilde{A},\tilde{B})$.

Matrix transformations like (6) form some of the tools of the numerical Analyst's trade, and related algorithms are used by Wilkinson [20], [21] and Van Dooren [22] in analyzing the generalized eigenvalue problem, and by Van Dooren [23] in analyzing general linear systems.

There are four mathematically equivalent theorems on controllability for the pair $(\tilde{A},\tilde{B})$:

Theorem (I) [2] If $A_{k,k-1} = 0$ (in (6)) then $(\tilde{A},\tilde{B})$ is uncontrollable, if $\text{rank}(A_{k,k-1}) = n_k$ then $(\tilde{A},\tilde{B})$ is controllable.

Theorem (II)[4],[5] The pair $(\tilde{A},\tilde{B})$ is controllable if and only if

$$\text{rank}(\tilde{B},\tilde{A}\tilde{B},\ldots,\tilde{A}^{n-1}\tilde{B}) = n$$

Definition (IV): Let $A \in \mathbb{R}^{n\times n}$ and λ_i, $i = 1(1)n$ be its eigenvalues, then we define $\lambda(A)$ to be the set which contains the eigenvalues of A only, $(\lambda(A) = \{\lambda_i | i=1(1)n\})$.

Theorem (III) [1], [6] The pair $(\tilde{A},\tilde{B})$ is controllable if and only if

$$\text{rank}[\tilde{B},\tilde{A}-\lambda_i I] = n \quad \text{for} \quad i = 1(1)n$$

where λ_i, $i = 1(1)n$ are the eigenvalues of A or $\tilde{A}$ (since $\lambda(A) = \lambda(\tilde{A})$, [12]).

Theorem (IV) [8], [9] The pair $(\tilde{A},\tilde{B})$ is controllable if and only if there exists at least one matrix $F \in \mathbb{R}^{m\times n}$ such that

$$\lambda(\tilde{A}) \cap \lambda(\tilde{A} + \tilde{B}F) = \emptyset$$

Proof: We show the equivalence of, and then prove, these four theorems by first showing

(I) If $A_{k,k-1} = 0$ then

(1) $\operatorname{rank}[\tilde{B},\tilde{A}\tilde{B},\ldots,\tilde{A}^{n-1}\tilde{B}] < n$

(2) $\operatorname{rank}[\tilde{B},\tilde{A}-\lambda_i I] < n$ for at least one integer i with $1 \le i \le n$.

(3) For every $F \in \mathbb{R}^{m\times n}$, $\lambda(\tilde{A}) \cap \lambda(\tilde{A} + \tilde{B}F) \neq \emptyset$.

(II) If $\operatorname{rank}(A_{k,k-1}) = n_k$ then

(1) $\operatorname{rank}[\tilde{B},\tilde{A}\tilde{B},\ldots,\tilde{A}^{n-1}\tilde{B}] = n$

(2) $\operatorname{rank}[\tilde{B},\tilde{A}-\lambda_i I] = n$ for $i = 1(1)n$

(3) There exists at least one $F \in \mathbb{R}^{m\times n}$ such that $\lambda(\tilde{A}) \cap \lambda(\tilde{A} + \tilde{B}F) = \emptyset$.

and then proving

(III) (1) The system (1) is controllable if and only if

$$\operatorname{rank}[B,AB,\ldots,A^{n-1}B] = n$$

(2) The system (2) is controllable if and only if

$$\operatorname{rank}[B,AB,\ldots,A^{n-1}B] = n$$

(I) Let $A_{k,k-1} = 0$ then

(1) We want to prove that $\text{rank}[\tilde{B},\tilde{A}\tilde{B},\ldots,\tilde{A}^{n-1}\tilde{B}] < n$. Since $A_{k,k-1} = 0$ the last n_k rows of the matrix $\tilde{A}^{\ell}$, $\ell = 0(1)n-1$ are going to be $(0,0,\ldots,0,A^{\ell}_{k,k})$. So the last n_k rows of the matrix $\tilde{A}^{\ell}\tilde{B}$, $\ell = 0(1)n-1$ are zero. Thus for every non-zero vector z, with $z \in \mathbb{R}^{n_k}$ there exists a non-zero vector $y^T = (0,0,\ldots,0,z^T)$, $y \in \mathbb{R}^n$ such that

$$y^T[\tilde{B},\tilde{A}\tilde{B},\ldots,\tilde{A}^{n-1}\tilde{B}] = 0 \Rightarrow \text{rank}[\tilde{B},\tilde{A}\tilde{B},\ldots,\tilde{A}^{n-1}\tilde{B}] < n .$$

(2) We want to prove that for some integer i with $1 \le i \le n$ and $\lambda_i \in \lambda(\tilde{A})$, $\text{rank}\,[\tilde{B},\tilde{A}-\lambda_i I] < n$. So we have to find a vector $y \in \mathbb{R}^n$ which satisfies the following conditions:

$$\left.\begin{aligned} &y \neq 0 \\ &y^T\tilde{B} = 0 \\ &y^T(\tilde{A}-\lambda_i I) = 0 \end{aligned}\right\} \qquad (7)$$

If we choose λ_i to be a solution of the eigenproblem $z^T A_{kk} = \lambda z^T$ and $z \in \mathbb{R}^{n_k}$ be a left eigenvector of A_{kk} corresponding to λ_i then the vector $y^T = (0,0,\ldots,0,z^T)$ satisfies all the conditions (7).

(3) We want to prove that for every $F \in \mathbb{R}^{m\times n}$, $\lambda(A) \cap \lambda(\tilde{A} + \tilde{B}F) \neq \emptyset$. Let $F \in \mathbb{R}^{m\times n}$ be a matrix such that $F = (F_1,F_2,\ldots,F_k)$ where $F_i \in \mathbb{R}^{m\times n_i}$ for $i = 1(1)k$. Then

$$\tilde{A}+\tilde{B}F = \begin{pmatrix} A_{11}+A_{10}F_1 & A_{12}+A_{10}F_2 & \cdot & A_{1,k-1}+A_{10}F_{k-1} & A_{1k}+A_{10}F_k \\ A_{21} & A_{22} & \cdot & A_{2,k-1} & A_{2k} \\ & \cdot & \cdot & \cdot & \cdot \\ & & \cdot & \cdot & \cdot \\ & & & 0 & A_{kk} \end{pmatrix}.$$

If D is the matrix which is derived from $\tilde{A} + \tilde{B}F$ by omitting its last k rows and its last k columns, then

$$\det[(\tilde{A} + \tilde{B}F) - \lambda I_n] = \det(A_{kk} - \lambda I_{n_k}) \det(D - \lambda I_{n-n_k})$$

So $\lambda(\tilde{A}) \cap \lambda(\tilde{A} + \tilde{B}F) \supseteq \lambda(A_{kk})$ for every $F \in \mathbb{R}^{m\times n}$

(II) Let $\operatorname{rank}(A_{k,k-1}) = n_k$

(1) We want to prove that $\operatorname{rank}[\tilde{B},\tilde{A}\tilde{B},\ldots,\tilde{A}^{n-1}\tilde{B}] = n$. We know that

$$\tilde{B} = \begin{pmatrix} A_{10} \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{pmatrix}, \quad \tilde{A}\tilde{B} = \begin{pmatrix} A_{11}A_{10} \\ A_{21}A_{10} \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{pmatrix}, \ldots, \quad \tilde{A}^{k-1}\tilde{B} = \begin{pmatrix} C_1 \\ C_2 \\ \cdot \\ \cdot \\ \cdot \\ C_{k-1} \\ A_{k,k-1}A_{k-1,k-2}\cdots A_{10} \end{pmatrix}$$

where C_i, $i = 1(1)k-1$ are appropriate $n_i \times n_0$ matrices. Besides we know that $\operatorname{rank}(A_{i,i-1}) = n_i$ for $i = 1(1)k$. That is, the matrices $A_{i,i-1}$, $i = 1(1)k$ are all of full row rank. So the matrices $A_{i,i-1}A_{i-1,i-1}\cdots A_{10}$, $i = 1(1)k$ are of full row rank. Thus there is no non-zero left null vector for the matrix $[\tilde{B},\tilde{A}\tilde{B},\ldots,\tilde{A}^{n-1}\tilde{B}]$.

(2) We want to prove that $\text{rank}[\tilde{B},\tilde{A}-\lambda_i I] = n$ for every i with $i = 1(1)n$ and $\lambda_i \in \lambda(\tilde{A})$ for $i = 1(1)n$. It is clear that, since $A_{i,i-1}$ is of full row rank for $i = 1(1)k$ then $[\tilde{B},\tilde{A}-\lambda_i I]$ has full row rank for every λ_i.

(3) We want to prove that there is at least one $F \in \mathbb{R}^{m\times n}$ such that $\lambda(\tilde{A}) \cap \lambda(\tilde{A} + \tilde{B}F) = \emptyset$. We know that

$$\tilde{A} = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1,k-1} & A_{1k} \\ A_{21} & A_{22} & \cdots & A_{2,k-1} & A_{2k} \\ & A_{32} & \cdots & A_{3,k-1} & A_{3k} \\ & & \ddots & \vdots & \vdots \\ & & & A_{k,k-1} & A_{kk} \end{pmatrix}, \quad \tilde{B} = \begin{pmatrix} A_{10} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad \text{where } A_{i,j} \in \mathbb{R}^{n_i \times n_j}$$

and the matrices $A_{j,j-1}$ are of full row rank, that is, $\text{rank}(A_{j,j-1}) = n_j$, $j = 1(1)k$.

We are going to give two proofs, the first will be for the simple case $m = n_0 = 1$, that is, when $\tilde{B}$ is a vector and the second will be for the general case.

<u>First Proof</u>: $m = n_0 = 1$, so let $\tilde{B} \equiv b = (\alpha_{10},0,\ldots,0)^T$ and

$$\tilde{A} = \begin{pmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1,n-1} & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2,n-1} & \alpha_{2n} \\ & \alpha_{32} & \cdots & \alpha_{3,n-1} & \alpha_{3n} \\ & & \ddots & \vdots & \vdots \\ & & & \alpha_{n,n-1} & \alpha_{nn} \end{pmatrix} \quad \text{where } \alpha_{j,j-1} \neq 0, \; j = 1(1)n.$$

Let λ_i , $i = 1(1)\ell$ with $\ell \leq n$, be the ℓ distinct eigenvalues of $\tilde{A}$, and x_i , $i = 1(1)\ell$ the eigenvectors of $\tilde{A}$ corresponding to λ_i . Let also $F \equiv f^T$, $f \in \mathbb{R}^n$. Then, if for some λ_i , $(\tilde{A} + bf^T) - \lambda_i I$ is singular there exists $y_f \in \mathbb{R}^n - \{0\}$, where $y_f^T = (\psi_{1f} , y_{2f}^T)$ and $\psi_{1f} \in \mathbb{R}$, such that

$$y_f^T[(\tilde{A} + bf^T) - \lambda_i I] = 0 \Rightarrow \tag{8}$$

$$y_f^T bf^T + y_f^T(\tilde{A} - \lambda_i I) = 0 \Rightarrow$$

$$y_f^T bf^T x_i + y_f^T(\tilde{A} - \lambda_i I)x_i = 0 \Rightarrow$$

$$y_f^T bf^T x_i = 0 \Rightarrow \psi_{1f}\alpha_{10}f^T x_i = 0$$

But $\psi_{1f} \neq 0$, because if $\psi_{1f} = 0$ then from (8) and because of the fact $\alpha_{j,j-1} \neq 0$, $j = 2(1)n$, y_{2f}^T would be zero too, so $y_f^T = 0$, thus $\psi_{1f} \neq 0$ and from $\psi_{1f}\alpha_{10}f^T x_i = 0 \Rightarrow$ $f^T x_i = 0$ since $\alpha_{10} \neq 0$. But if we choose,

$$f^T = e^T(X^T X)^{-1}X \tag{8a}$$

where the columns of X are the eigenvectors of $\tilde{A}$, then $f^T x_i \neq 0$, $i = 1(1)\ell$, which is a contradiction. So for f^T given in (8a) $\lambda(\tilde{A}) \cap \lambda(\tilde{A} + bf^T) = \emptyset$.

Second Proof: For $m \geq 1$. Here we shall prove that for almost all $m \times n$ matrices F we have

$$\lambda(\tilde{A}) \cap \lambda(\tilde{A} + \tilde{B}F) = \emptyset \tag{9}$$

and we shall construct an F for which (9) will be true.

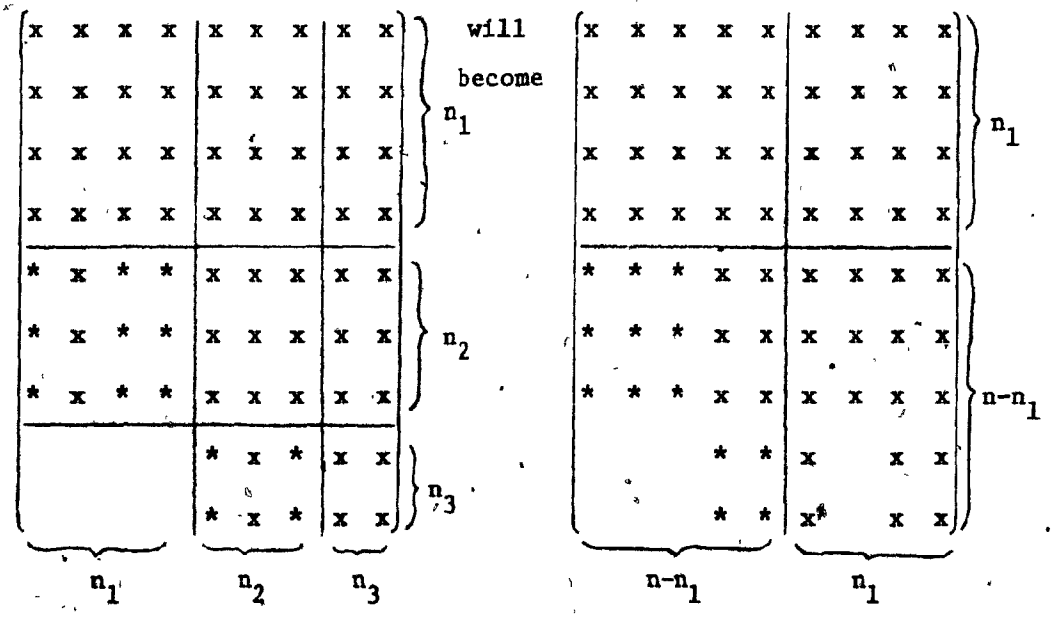
Fact: For any integer $p \geq 1$, almost all $p \times p$ matrices are nonsingular. By this is meant that if the elements of a matrix are independently chosen from a continuous uniform distribution, then with probability 1 the matrix will be nonsingular.

Suppose $\tilde{A}$ has eigenvalues λ_i, $i = 1(1)n$, then we perform the following construction: (Note that $A_{j,j-1}$ has n_j linearly independent columns) for given i, with $i = 1(1)n$ and

for $j = 2(1)k$ choose n_j columns of $\tilde{A}-\lambda_i I$ corresponding to n_j linearly independent columns of $A_{j,j-1}$ and move them to the front of the matrix, that is to positions

$$1 + \sum_{i=2}^{j-1} n_i \quad \text{to} \quad \sum_{i=2}^{j} n_i \text{, (where } \sum_{i=2}^{1} \text{ is ignored).}$$

For example the matrix



where stars represent the n_2 and n_3 linearly independent columns of A_{21} and A_{32} respectively. We may then write the result as, with permutation matrix P,

$$(\tilde{A}-\lambda_i I)P = \left[\begin{array}{c|c} A_{11}^{(i)} & A_{12}^{(i)} \\ \hline A_{21}^{(i)} & A_{22}^{(i)} \end{array}\right] \begin{matrix} \}n_1 \\ \}n-n_1 \end{matrix} \qquad (9\alpha)$$

$$\underbrace{\phantom{A_{21}^{(i)}}}_{n-n_1} \quad \underbrace{\phantom{A_{22}^{(i)}}}_{n_1}$$

where $A_{21}^{(i)}$ is block upper triangular and nonsingular since it has nonsingular $n_j \times n_j$ blocks on its diagonal, $j = 2(1)k$. (These blocks being the same for all choices of λ_i, $i = 1(1)n$). Now write $\tilde{B}FP = \left[\begin{array}{c|c} F_{11} & F_{12} \\ \hline 0 & 0 \end{array}\right] \begin{matrix} \}n_1 \\ \}n-n_1 \end{matrix}$ so

$$\underbrace{}_{n-n_1} \quad \underbrace{}_{n_1}$$

$$[(\tilde{A} + \tilde{B}F) - \lambda_i I]P = \begin{bmatrix} A_{11}^{(i)}+F_{11} & A_{12}^{(i)}+F_{12} \\ A_{21}^{(i)} & A_{22}^{(i)} \end{bmatrix}$$

For this to be singular there exists $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \begin{matrix} \}n-n_1 \\ \}n_1 \end{matrix}$, $x \neq 0$

such that $[(\tilde{A} + \tilde{B}F) - \lambda_i I]Px = 0$ so first

$$A_{21}^{(i)}x_1 + A_{22}^{(i)}x_2 = 0 \Rightarrow$$

$$x_1 = -[A_{21}^{(i)}]^{-1} A_{22}^{(i)}x_2 \qquad (9\beta)$$

From (9β) we can see that $x_2 = 0$ implies $x_1 = 0$ and therefore $x = 0$, thus $x_2 \neq 0$. Next

$$(A_{11}^{(i)} + F_{11})x_1 + (A_{12}^{(i)} + F_{12})x_2 = 0 \Rightarrow$$

$$[A_{12}^{(i)} + F_{12} - (A_{11}^{(i)} + F_{11})[A_{21}^{(i)}]^{-1}A_{22}^{(i)}]x_2 = 0 \qquad (9\gamma)$$

where $x_2 \neq 0$.

Let $C^{(i)} = A_{12}^{(i)} + F_{12} - (A_{11}^{(i)} + F_{11})[A_{21}^{(i)}]^{-1} A_{22}^{(i)}$.

From (9γ) we conclude that $C^{(i)}$ must be singular. But if the elements of F are chosen randomly from the uniform distribution $[0,1]$, then F_{11} and F_{12} will be independent, so given F_{11}, for almost all F_{12} $C^{(i)}$ will be nonsingular. For a given F there are at most n different $C^{(i)}$, so for almost all F, $(\tilde{A} + \tilde{B}F) - \lambda_i I$ is nonsingular for all eigenvalues λ_i of $\tilde{A}$. Then with probability 1 $\tilde{A}$ and $\tilde{A} + \tilde{B}F$ have no eigenvalue in common. Now we shall construct an F such that $\lambda(\tilde{A}) \cap \lambda(\tilde{A} + \tilde{B}F) = \emptyset$. Choose F so that

$$A_{10}FP = (\underbrace{0}_{n-n_1} \mid -\gamma I_{n_1}), \quad \gamma \in \mathbb{C}$$

that is let $F = A_{10}^T(A_{10}A_{10}^T)^{-1}(0, -\gamma I)P^T$.

Note that $(A_{10}A_{10}^T)^{-1}$ exists since A_{10} has full row rank. Also note that F is independent of the choice of λ_i. Then

$$[(\tilde{A} + \tilde{B}F) - \lambda_i I]P = \begin{bmatrix} A_{11}^{(i)} & A_{12}^{(i)} - \gamma I \\ A_{21}^{(i)} & A_{22}^{(i)} \end{bmatrix}$$

So for this choice of F the corresponding relation to (9γ) will be:

$$[(A_{12}^{(i)} - A_{11}^{(i)}[A_{21}^{(i)}]^{-1}A_{22}^{(i)}) - \gamma I]x_2 = 0 \ , \ x_2 \neq 0 \quad (9\delta)$$

So γ must be an eigenvalue of the $n_1 \times n_1$ matrix

$$A^{(i)} = A_{12}^{(i)} - A_{11}^{(i)}[A_{21}^{(i)}]^{-1}A_{22}^{(i)}$$

$A^{(i)}$ has at most n_1 distince eigenvalues, and as there are at most n distinct eigenvalues of $\tilde{A}$, there are at most n distinct $A^{(i)}$, so there are at most nn_1 distinct values of γ for which $\tilde{A} + \tilde{B}F$ has any eigenvalue equal to any eigenvalue of A. Thus for almost all F of even this most restricted form, $\tilde{A}$ and $\tilde{A} + \tilde{B}F$ have completely distince eigenvalues.

(III) For this part the simplest proof was found in [3] (p. 277-281, 282-283) and it is as follows.

Lemma (I): Let $M = [B,AB,\ldots,A^{n-1}B]$ then for any $q \geq n$ we have

$$\text{rank}[B,AB,\ldots,A^{q-1}B] = \text{rank}(M)$$

Proof: As k increases by one unit the rank of the matrix $M_k = [B,AB,\ldots,A^{k-1}B]$ either increases (by at least 1) or remains constant. Suppose that k is an integer such that the rank of M_{k+1} is equal to the rank of M_k. That means that the m columns comprising A^kB are each linearly

dependent on the (previous) columns in M_k . That is, there is a relation of the form

$$A^k B = BD_0 + ABD_1 + \ldots + A^{k-1}BD_{k-1}$$

where each D_i is an $m \times m$ matrix. Now multiplication of this relation by A leads to the new relation

$$A^{k+1}B = ABD_0 + A^2BD_1 + \ldots + A^kBD_{k-1}$$

which shows that the columns comprising $A^{k+1}B$ are linearly dependent on the columns in M_{k+1} . Therefore, the rank of M_{k+2} is the same as the rank of M_{k+1} . By continuing this argument, we see that for all $j > k$ the rank of M_j is equal to that of M_k . Thus we have shown that, in the progression of M_k's , once the rank fails to increase, it will remain constant even as additional columns are adjoined. In view of the above, the rank of M_k increases by at least 1 at each increment of k until it attains its maximum rank. Since the maximum rank is at most n , the maximum rank is attained within n steps (that is, by M_n) .

(1) Suppose first that the rank condition does not hold. For any $t_1 > 0$ and any integrable control $u(t)$ defined on $[0, t_1]$ we have

$$x(t_1) = \int_0^{t_1} e^{A(t_1-t)} B\, u(t)\, dt \Rightarrow \tag{10}$$

$$x(t_1) = B \int_0^{t_1} u(t)\, dt + AB \int_0^{t_1} (t_1-t)\, u(t)\, dt + \ldots$$

when evaluated, the integrals in the above expression are simply constant m-dimensional vectors. Therefore, the expression shows that $x(t_1)$ is a linear combination of the columns of $B, AB, \ldots$. By the earlier lemma, if the rank of M is less than n , then even the infinite set of vectors $B, AB, A^2B, \ldots$ does not contain a full basis for the entire n-dimensional space. Thus, there is a vector x_1 that is linearly independent of all these vectors, and therefore cannot be attained.

Now suppose that the rank condition does hold. We will show that the system is completely controllable and that in fact the state can be transferred from zero to an arbitrary point x_1 within an arbitrary short period of time.

We first show that for any $t_1 > 0$ the matrix

$$K = \int_0^{t_1} e^{-At} BB^T e^{-A^T t} \, dt$$

is nonsingular. To prove this, suppose there is a vector a such that $Ka = 0$. Then

$$a^T K a = 0$$

or more explicitly

$$\int_0^{t_1} a^T e^{-At} BB^T e^{-A^T t} a \, dt = 0 \qquad (11)$$

The integrand above has the form $c(t)^T c(t)$, where $c(t) = B^T e^{-A^T t} a$. It follows that the integrand is always nonnegative. For the integral (11) to vanish, it follows that the integrant must vanish identically for $0 \le t \le t_1$. Therefore

$$a^T e^{-At} B = 0 \qquad \text{for all } t,\ 0 \le t \le t_1$$

Evaluation of this expression, and its successive derivatives, with respect to t, at $t = 0$ leads to the following sequence of equations:

$$a^T B = 0$$
$$a^T AB = 0$$
$$\vdots$$
$$a^T A^{n-1} B = 0$$

This means that the vector a must be orthogonal to all columns of the matrix M. Since it is assumed that this matrix has rank n, it must follow that $a = 0$. Therefore K is non-singular. Now, given x_1, select any $t_1 > 0$ and set

$$u(t) = B^T e^{-A^T t} K^{-1} e^{-At_1} x_1 \qquad (12)$$

Then from (10) $x(t_1) = \int_0^{t_1} e^{A(t_1 - t)} BB^T e^{-A^T t} K^{-1} e^{-At_1} x_1 \, dt \Rightarrow$

$$x(t_1) = e^{At_1} K K^{-1} e^{-At_1} x_1 = x_1$$

Therefore the control (12) transfers the state from zero to x_1, and the system is completely controllable.

(2) Suppose a sequence of controls $u_1, u_2, \ldots, u_q$ is applied to the system (2) with $x_1 = 0$ it follows that

$$x_{q+1} = A^q B\, u_1 + A^{q-1} B\, u_2 + \ldots + B\, u_q$$

From this formula we see that points in state space can be reached if and only if they can be expressed as linear combinations of powers of A times B. Thus the issue of complete controllability rests on whether the infinite sequence $B, AB, A^2B, \ldots$ has a finite number of columns that span the entire n-dimensional space. By the earlier lemma, however, these span the full n-dimensional space if and only if

$$\text{rank}[B, AB, \ldots, A^{n-1}B] = n \ .$$

Remark (IV) From step (III) we can see that (1) is controllable if and only if (2) is controllable. So it is the matrices A and B which determine controllability and not the type of the system (continuous or discrete).

(III) The computational problem of controllability

In the previous section we presented four mathematically equivalent theorems on controllability. These theorems can lead to several different computational approaches, four of which are the following.

C1: Involves transforming A and B to the form in (6) and is described in [2].

C2: Form the matrix $[B,AB,\ldots,A^{n-1}B]$ and then compute its rank.

C3: Compute the eigenvalues λ_i, $i = 1(1)n$ of A and then compute the ranks of the matrices $[B,A-\lambda_i I]$ for $i = 1(1)n$.

C4: For a random matrix $F \in \mathbb{R}^{m\times n}$ form the matrix $A + BF$ and then compute the eigenvalues of $A + BF$ and A.

In this section we will prove that the algorithms which can be made by using C2, C3, and C4 are poor. This will be done by using counter-examples. We will also show that the algorithm which is based on C1 works perfectly for the same examples.

We will show the weakness of C2 first. Before this we give two definitions.

Definition (V) [12], [14]: Let $D \in \mathbb{C}^{n\times m}$ with $n \geq m$, then the singular value decomposition (SVD) of D is $D = USV^H$ where $U \in \mathbb{C}^{n\times n}$, $V \in \mathbb{C}^{m\times m}$, $S = \begin{pmatrix} S_1 \\ 0 \end{pmatrix}$ with $S_1 = \text{diag}\,(\sigma_1,\ldots,\sigma_m)$, $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_m \geq 0$, U and V are unitary matrices. Let $V_1 = V$ and $U = (U_1, U_2)$ where $U_1 \in \mathbb{C}^{n\times m}$. Then $D = U_1 S_1 V_1^H$. The real scalars σ_i, $i = 1(1)m$ are called the singular values of D and the columns of V_1 and U_1 are called the right and left singular vectors, respectively.

Definition (VI): Let $D \in \mathbb{R}^{n\times n}$, then the number

$$\chi(D) \triangleq \frac{\sigma_1(D)}{\sigma_n(D)} \geq 1$$

is called the condition number of D for solution of equations, where $\sigma_1(D)$ and $\sigma_n(D)$ are the largest and the smallest singular values of D. If $\chi(D)$ is large then the problem of the solution of the equation $Dy = d$, where $d \in \mathbb{R}^n$, $y \in \mathbb{R}^n$, is ill-conditioned, for more details about condition numbers see [12, page 192].

Consider now the clearly controllable system

$$\dot{x} = A_n x + B_n u \ , \tag{13}$$

where $A_n = \text{diag}\ (1, 2^{-1}, \ldots, 2^{1-n})$, $B_n = (\underbrace{1,1,\ldots,1}_{n})^T$.

In this case the (i,j) element of the matrix $C_n = [B, AB, \ldots, A^{n-1}B]$ is $2^{(i-1)(1-j)}$ which can be formed and stored with full accuracy on most computers. Even so with $n \in \{8,9,10\}$ we have the following results:

TABLE I

RESULTS OF COMPUTATIONAL TESTS ON (13) USING C2

n	$\sigma_1(C_n)$	$\sigma_8(C_n)$	$\sigma_9(C_n)$	$\sigma_{10}(C_n)$	$\chi(A_n)$	$\chi(C_n)$
8	3.3033	2.0688×10^{-8}	-	-	128	0.1596×10^{9}
9	3.4704	4.7752×10^{-8}	1.5846×10^{-10}	-	256	0.219×10^{11}
10	3.6298	7.1246×10^{-8}	3.6402×10^{-10}	6.1288×10^{-13}	512	0.5922×10^{13}

where $\sigma_i(C_n)$ is the ith singular value of C_n. So the numerical rank of C_n when $n \in \{8,9,10\}$ will be less than n using the SVD on a computer with relative precision no smaller than 10^{-7}. Obviously, in

these cases C2 would fail, indicating that the controllable system (13) is uncontrollable when $n \in \{8,9,10\}$.

This happens because if we transform a problem by matrix multiplication and the transforming matrix is ill-conditioned with respect to solution of equations, that is, it has large condition number, then the transformed problem will usually be more sensitive to changes in data, than the original one. As we can see from table I $\chi(A_n) \gg 1$ when $n \in \{8,9,10\}$ and as a consequence of this the resulting matrix C_n has much larger condition number $\chi(C_n)$ than A_n . So C_n is more sensitive to changes in data than A_n .

In order to compare C1 with C2 we applied C1 to (13) for $n \in \{8,9,10\}$. Since in (13) B is a column vector then in (6) k is equal to 8, 9 or 10 when n is equal to 8, 9 or 10 , respectively, and $A_{i,i-1}$ is scalar for $i = 1(1)k$. Theoretically C1 should give

$$A_{i,i-1} \neq 0 \quad , \quad i = 1(1)k \ .$$

The results which were taken from the actual performance of C1 are presented in table II.

TABLE II

RESULTS OF COMPUTATIONAL TESTS ON (13) USING C1

n	A_{10}	$i = 2(1)\ k-1$	$A_{k,k-1}$
8	-2.828426	$0.024 < \lvert A_{i,i-1} \rvert < 0.323$	-0.010119
9	-3	$0.012 < \lvert A_{i,i-1} \rvert < 0.314$	0.005113
10	-3.162277	$0.006 < \lvert A_{i,i-1} \rvert < 0.305$	-0.002570

So by observing the results presented in table II we can conclude that (13) is controllable when $n \in \{8,9,10\}$, and this conclusion would be reached using a relative precision as low as 10^{-4} . Thus C1 succeeded where C2 failed.

In the rest of this section we will show the weakness of C3 and C4 . We will do this by the same way we showed the weakness of C2 , that is, by considering a counterexample, because we know that a good example is sufficient to condemn a poor algorithm. The approaches in C3 and C4 depend on finding eigenvalues. Unlike singular values the eigenvalues of some matrices can be very ill-conditioned, that is, very sensitive to small changes in the matrix. So we choose a matrix which has already been presented, by Wilkinson [13, p. 90], as an extremely ill-conditioned case regarding the eigenvalues. The matrix Wilkinson considered was the following:

$$\hat{A} = \begin{bmatrix} 20 & 20 & & & & \\ & 19 & 20 & & & \\ & & 18 & 20 & & \\ & & & \ddots & \ddots & \\ & & & & 2 & 20 \\ & & & & & 1 \end{bmatrix}$$

The eigenvalues of $\hat{A}$ are $\lambda_i = i$, $i = 1(1)20$. Consider now the uncontrollable system

$$\dot{x} = Ax + Bu \qquad (14)$$

with $A = Q^T \hat{A} Q$, $B^T = (\underbrace{1,1,\ldots,1}_{19},0)Q$

where Q is a random orthogonal matrix found by finding the SVD of a square matrix whose elements are uniform random numbers on (-1,1) obtained by the VNI(0) random number generator. To test C4 we calculated the eigenvalues of A and A + BF . The sets $\tilde{\lambda}(A)$ of the computed eigenvalues of A and $\tilde{\lambda}(A + BF)$ of A + BF were obtained using single precision (6 significant hexidecimal digits) on the AMDAHL 470/V7 computer at McGill University. The elements of F were random numbers on (-1,1) from the VNI(0) random number generator. To test C3 we calculated the values

$$\rho(B,A-\tilde{\lambda}_i I) \triangleq \text{ratio of smallest to largest singular value of } (B,A-\tilde{\lambda}_i I) . \quad (15)$$

where $\tilde{\lambda}_i \in \tilde{\lambda}(A)$ for $i = 1(1)20$.

The ratio $\rho(B,A-\tilde{\lambda}_i I)$ was the same for both eigenvalues of a complex conjugate pair. The results are given in table III.

TABLE III

RESULTS OF COMPUTATIONAL TESTS ON (14)

$\rho(B, A-\tilde{\lambda}_i I)$	$\tilde{\lambda}(A)$	$\tilde{\lambda}(A + BF)$
2.024×10^{-3}	-0.32985±j1.06242	0.99999
3.979×10^{-3}	0.92191±j3.13716	-8.95872±j 3.73260
7.3203×10^{-3}	3.00339±j4.80414	-5.11682±j 9.54329
1.187×10^{-2}	5.40114±j6.17864	-0.75203±j14.148167
1.846×10^{-2}	8.43769±j7.24713	5.77659±j15.58436
2.5905×10^{-2}	11.82747±j7.47463	11.42828±j14.28694
3.221×10^{-2}	15.10917±j6.90721	13.30227±j12.90197
4.036×10^{-2}	18.06886±j5.66313	18.59961±j14.34739
5.179×10^{-2}	20.49720±j3.81950	23.94877±j11.80677
6.436×10^{-2}	22.06287±j1.38948	28.45618±j 8.45907
		32.68478

Theoretically one of the eigenvalues of A should be the same as an eigenvalue of $A + BF$, this one being unity. But as we can see from table III the computed eigenvalues of A are almost unrelated with the true eigenvalues of A and the computed eigenvalues of $A + BF$. So in this case C4 would tell us a clearly uncontrollable system is controllable. As we can conclude from the results of table III the failure of C4 is due to the ill-conditioned eigenvalues of $\hat{A}$. The approach C3 failed too, because in theory one value of ρ should be zero, but no value of ρ can be considered as zero within the previously mentioned accuracy. So C3 would have indicated that the uncontrollable system (14) was controllable. C3 was also carried out with the true eigenvalue, unity, of A, and this gave

$$\rho(B,A-I) = 5.293 \times 10^{-8}$$

indicating that it is the eigenvalue computation that caused the failure of C3, just as for C4.

In order to compare the ability of approach C1 we applied it to system (14). Since, as in the previous counterexample, B is a column vector then in (6) $k = 20$ and $A_{i,i-1}$ is scalar for $i = 1(1)20$. Theoretically, C1 should give

$$A_{i,i-1} \neq 0 \ , \ i = 1(1)19 \ \text{ and } \ A_{20,19} = 0 \ .$$

In practice C1 gave $A_{1,0} = 4.35887$; $A_{2,1} = 8.30008$; $19 < |A_{i,i-1}| < 22$, $i = 3(1)19$ and $A_{20,19} = 0.000006914$, showing that the system is uncontrollable to this precision of computation. To see if this was just a

coincidence the algorithm was applied to (14) with 50 different random orthogonal matrices Q . In all cases similar good results were obtained.

Some of the ideas and results of this section have already been presented in [2].

(IV) DISTANCE FROM AN UNCONTROLLABLE SYSTEM

As we saw in equation (3) of section (II), μ is defined to be the distance between the system defined by the matrix (A,B) and the nearest uncontrollable system defined by the matrix $(A + \delta A , B + \delta B)$. A knowledge of μ would generally be more useful than just knowing if the system is controllable or not. In this section an algorithm is given which will make use of remark (III) to determine whether a system is controllable or not. The approach could be extended to compute μ , but this is not done here as it appears to be inefficient.

We will relate now μ with another quantity which can be calculated, that is, we are going to prove that

$$\mu = \min_{\lambda} \sigma_n(B, A-\lambda I) \quad , \lambda \in \mathbb{C} \tag{16}$$

where $\sigma_n(B, A-\lambda I)$ is the smallest singular value of $(B, A-\lambda I)$.

Note: In the following we will represent the singular values of $(B,A-\lambda I)$ by either $\sigma_i(B, A-\lambda I)$ or $\sigma_i(\lambda)$ meaning exactly the same thing, that is, $\sigma_i(\lambda) \equiv \sigma_i(B, A-\lambda I)$, $i = 1(1)n$ and $\forall \lambda \in \mathbb{C}$.

Before we go ahead to prove (16) we shall first prove a lemma which will help us in the proof of (16).

Lemma (II): Let $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{m\times n}$, where $m \geq n$ and let $A = U_A \Sigma_A V_A^H$, $B = U_B \Sigma_B V_B^H$ be the singular value decompositions of A and B . Let also $\alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_n$ and $\beta_1 \geq \beta_2 \geq \ldots \geq \beta_n$ be the singular values of A and B respectively. Then for given A we shall prove that:

$$\underset{\text{rank}(B)\leq k}{\text{minimum}} \; ||A - B||_2 = \alpha_{k+1} \;, \; k \leq n \text{ and } \alpha_{n+1} = 0 \qquad (17)$$

Proof: We know ([12], p. 231, th. 6.6) that

$$||A - B||_2 \geq |\alpha_i - \beta_i| \;, \quad i = 1(1)n$$

so obviously $||A - B||_2 \geq \max |\alpha_i - \beta_i|$, $i \in \Delta_n$. Now if $\text{rank}(B) \leq k$ with $k \leq n$ we have that $\beta_i \geq 0$, $i = 1(1)k$ and $\beta_i = 0$, $i = k+1(1)n$.

Let α be the maximum of $|\alpha_i - \beta_i|$ with $i \in \Delta_k$, that is, $\alpha = \max_i |\alpha_i - \beta_i|$, $i \in \Delta_k$. Then we have that

$$||A - B||_2 \geq \max(\alpha \;, \; \alpha_{k+1})$$

So when $\text{rank}(B) \leq k$ we have $||A - B||_2 \geq \alpha_{k+1}$ for every $B \in \mathbb{R}^{m\times n}$.

Now when $\text{rank}(B) \leq k$ there is a B such that $B = U_A \Sigma_B V_A^H$ and $\beta_i = \alpha_i$, $i = 1(1)k$. For this B we obviously have $||A - B||_2 = \alpha_{k+1}$.

So indeed $\underset{\text{rank}(B)\leq k}{\text{minimum}} \; ||A - B||_2 = \alpha_{k+1}$.

Now we are ready to prove (16).

Proof: Let (A,B) define a system and $[(A + \delta A), (B + \delta B)]$ define an uncontrollable system. Also let $D(\lambda) = [(B + \delta B), (A + \delta A) - \lambda I]$, $\lambda \in \mathbb{C}$, and

$$(B,A-\lambda I) = U(\lambda) \begin{bmatrix} \sigma_1(\lambda) & & & \\ & \sigma_2(\lambda) & & 0 \\ & & \ddots & \\ & & & \sigma_n(\lambda) \end{bmatrix} V^H(\lambda) \qquad (18)$$

be the singular value decomposition of $(B, A-\lambda I)$ with

$\sigma_1(\lambda) \geq \sigma_2(\lambda) \geq \ldots \geq \sigma_n(\lambda)$.

From lemma (I) for $k \leq n$ we get ($\sigma_{n+1}(\lambda)$ will be considered zero)

$$\underset{\text{rank}[D(\lambda)]\leq k}{\text{minimum}} \; ||(B,A-\lambda I) - [(B + \delta B), (A + \delta A) - \lambda I]||_2 = \sigma_{k+1}(\lambda) \Rightarrow$$

$$\underset{\text{rank}[D(\lambda)]\leq k}{\text{minimum}} \; ||(\delta A, \delta B)||_2 = \sigma_{k+1}(\lambda) \qquad (19)$$

The smaller the 2-norm of $(\delta A, \delta B)$ is, the nearer the uncontrollable system $[(A + \delta A), (B + \delta B)]$ is to the system (A,B). Since $[(A + \delta A), (B + \delta B)]$ is uncontrollable obviously $k \leq n-1$. We can see that in (19) there are two parameters λ and k. Regarding k, $||(\delta A, \delta B)||_2$ takes its minimum value and at the same time keeps $[(A + \delta A), (B + \delta B)]$ uncontrollable, if and only if $k = n-1$. So we have that:

$$\mu = \min_{\lambda} \left[\underset{\text{rank}[D(\lambda)]\leq n-1}{\text{minimum}} \; ||(\delta A, \delta B)||_2 \right] = \min_{\lambda} \sigma_n(\lambda)$$

Thus $\mu = \min_{\lambda} \sigma_n(B, A-\lambda I)$, $\lambda \in \mathbb{C}$ (20)

Remark (V): An important property of the singular values is that they are not very sensitive to changes in the matrix. In fact if $\sigma_1 \geq \ldots \geq \sigma_n$ are the singular values of the matrix D and $\tilde{\sigma}_1 \geq \ldots \geq \tilde{\sigma}_n$ the singular values of the perturbed matrix $D + \delta D$, then

$$|\sigma_i - \tilde{\sigma}_i| \leq ||\delta D||_2 \quad , \; i = 1(1)n \; . \tag{21}$$

For more information see [12, page 321] or [14, page 24]. Thus, from (21), we conclude that the singular values are well-conditioned with respect to perturbations in the matrix. So, from (20) we can also conclude that μ is well-conditioned with respect to perturbations in the matrix [2, page 137].

Remark (VI): We can easily observe that $\sigma_n(B, A-\lambda I)$ is a function with domain $\mathcal{D} \subseteq \mathbb{C}^{n\times(n+m)}$ and range $P \subseteq \mathbb{R}$, that is,

$$\sigma_n(B, A-\lambda I): \mathcal{D} \longrightarrow P \tag{21a}$$

We are going to present two algorithms, one for $\lambda \in \mathbb{R}$ and one for $\lambda \in \mathbb{C}$. But before we present the algorithms we will make some useful observations about the function $\sigma_n(B, A-\lambda I)$

Remark (VII): Let $\sigma_1' \geq \sigma_2' \geq \ldots \geq \sigma_n'$ be the singular values of (B,A), then using (21) we can see that $(0, \lambda I)$ shifts the singular values of (B,A) by at most $|\lambda|$, because

$$|\sigma_i(\lambda) - \sigma_i'| \leq ||(B, A-\lambda I) - (B,A)||_2 = ||(0,\lambda I)||_2 = |\lambda|$$

for $i = 1(1)n$.

Since we are interested only in the smallest singular value of $(B, A-\lambda I)$, we have

$$|\sigma_n(\lambda) - \sigma_n'| \leq |\lambda| \quad \Rightarrow$$

$$\sigma_n' - |\lambda| \leq \sigma_n(\lambda) \quad . \tag{22}$$

Similarly, (B,A) can shift the singular values of $(0, \lambda I)$ by at most $||(B,A)||_2$, because

$$|\sigma_i(\lambda) - |\lambda|| \leq ||(B, A-\lambda I) - (0,-\lambda I)||_2 = ||(B,A)||_2 = \sigma_1' ,$$

$$i = 1(1)n \ .$$

For $i = n$ we have that

$$|\sigma_n(\lambda) - |\lambda|| \leq \sigma_1' =$$

$$|\lambda| - \sigma_1' \leq \sigma_n(\lambda) \ . \tag{23}$$

From (22) (respectively from (23)) we can easily observe that if $|\lambda| < \sigma_n'$ (respectively $|\lambda| > \sigma_1'$) then $\sigma_n(\lambda) > 0$. Since we are interested in finding if $\sigma_n(\lambda)$ is ever zero we should search in the ring $\sigma_n' \leq |\lambda| \leq \sigma_1'$ (see fig. 1).

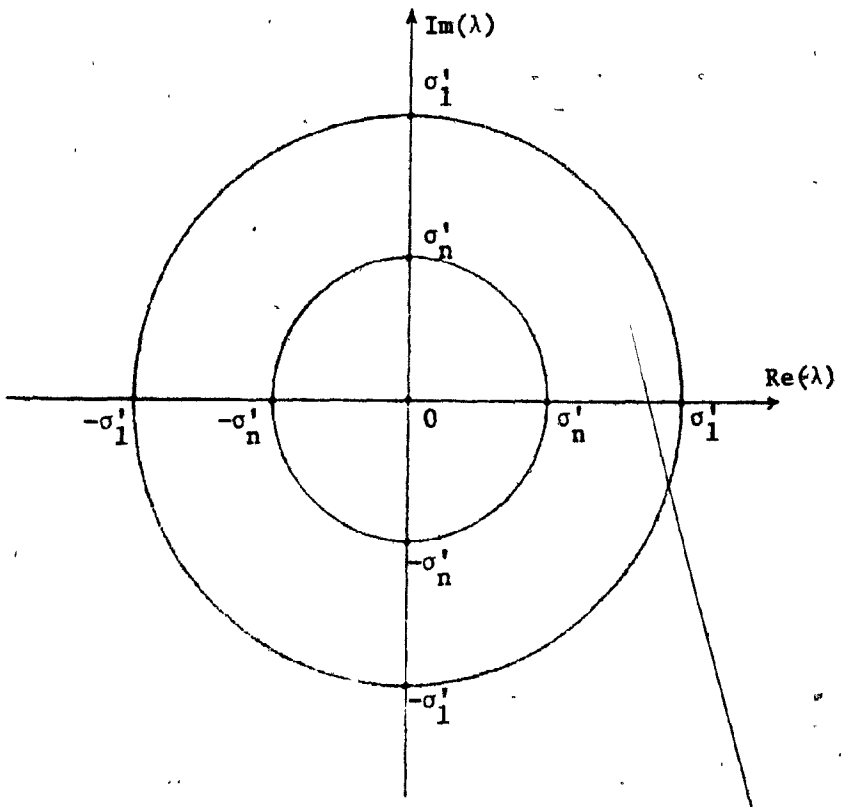


Fig. 1

where $Re(\lambda)$ and $Im(\lambda)$ is the real and the imaginary part of λ, respectively.

Remark (VIII): For A having real eigenvalues we ran some programs in which we plotted the values of $\sigma_n(B, A-\lambda I)$ for λ in the real interval Δ where every eigenvalue λ_i, $i = 1(1)n$ of A was in Δ too, that is, $\lambda_i \in \Delta$, $i = 1(1)n$. The plots we obtained had the general form of fig. 2 when $\mu > 0$ (system controllable) and fig. 3 when $\mu = 0$ (system uncontrollable).

CONTROLLABLE CASE

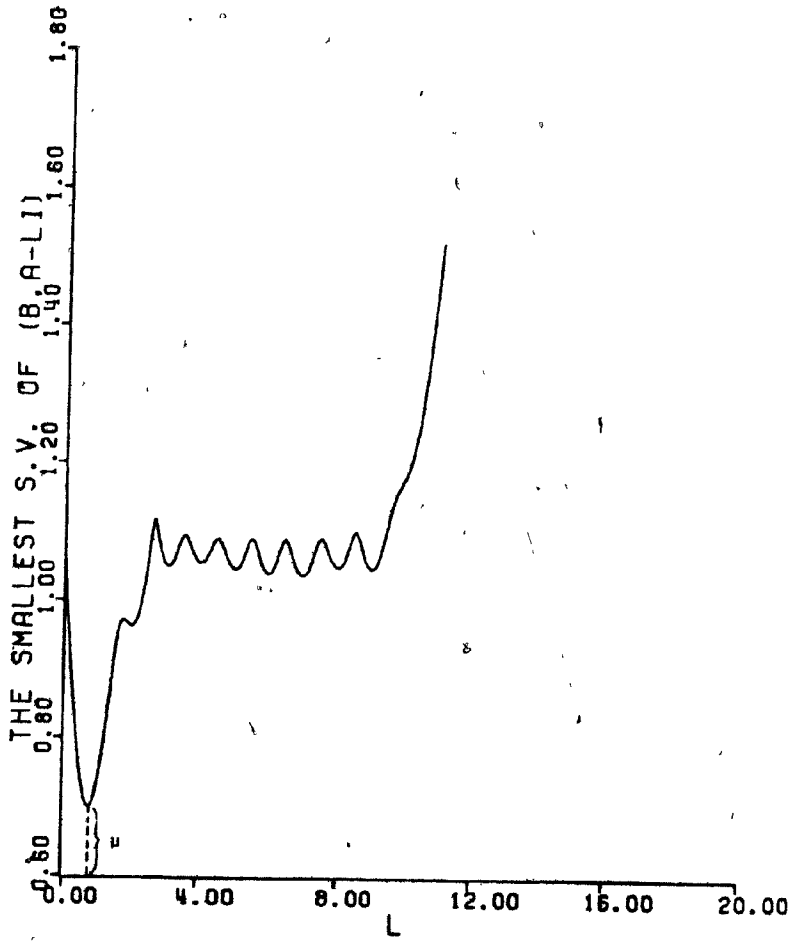


Fig. 2

UNCONTROLLABLE CASE

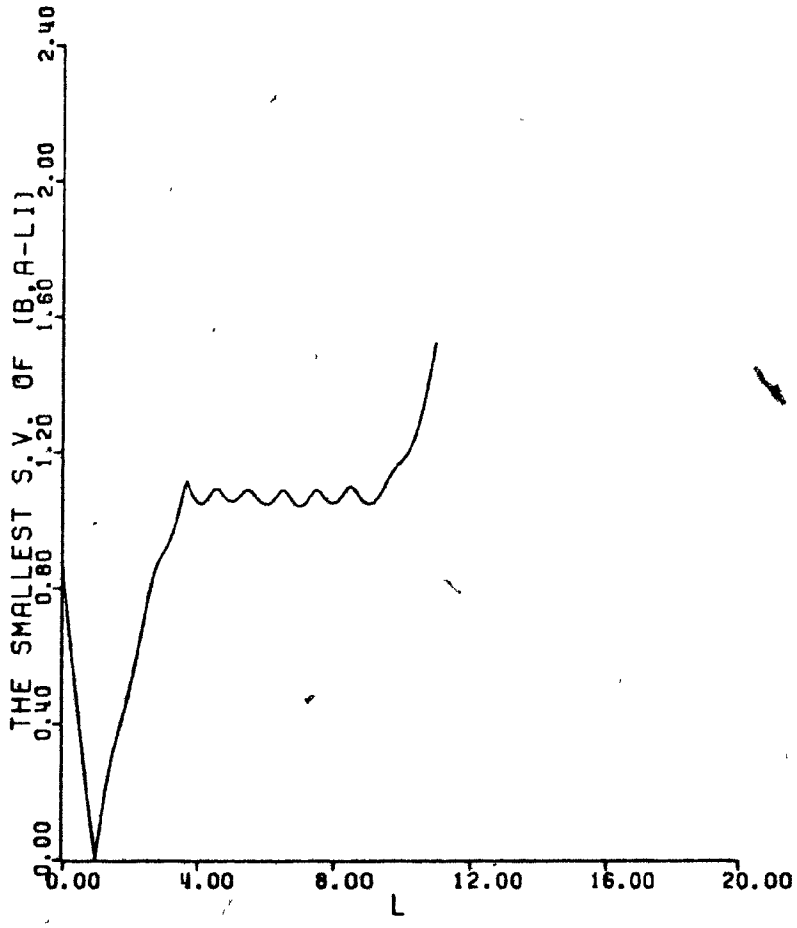


Fig. 3

Note: As we can see from fig. 3 if the system is uncontrollable then the function $\sigma_n(B, A-\lambda I)$ is almost linear in the neighbourhood of λ', where λ' is such that $\sigma_n(B, A-\lambda' I) = 0$.

Remark (IX): Now (21) will help us bound the rate of change in $\sigma_n(B,A-\lambda I)$ with respect to λ. Let λ_1, λ_2 be two distinct complex numbers, then

$$|\sigma_n(B,A-\lambda_1 I) - \sigma_n(B,A-\lambda_2 I)| \leq ||(B,A-\lambda_2 I) - (B,A-\lambda_2 I)||_2$$

$$\leq ||[0, (\lambda_2-\lambda_1)I]||_2$$

$$= |\lambda_1 - \lambda_2| \Rightarrow$$

$$\frac{|\sigma_n(B,A-\lambda_1 I) - \sigma_n(B,A-\lambda_2 I)|}{|\lambda_1 - \lambda_2|} \leq 1 \qquad (24)$$

This means that $|\dot{\sigma}_n(B,A-\lambda I)| \leq 1$ wherever the derivative is defined, in other words, (24) means that at every point of the curve $\sigma_n(B,A-\lambda I)$ the tangent (if it can be defined) makes an angle ω with the horizontal axis such that

$$\omega \in [0, \tfrac{\pi}{4}] \cup [\tfrac{3}{4}, \pi] . \qquad (25)$$

Remark (X): In remark (VII) we mentioned that if $\sigma_n(B,A-\lambda I)$ has a zero, then this zero will be in the ring of fig. 1. So, we should search all the ring in order to find the possible root. Now we will show that we do not have to search all the ring because the plot of $\sigma_n(B,A-\lambda I)$ above the axis $Re(\lambda)$ is completely identical with the

corresponding plot below the axis $Re(\lambda)$. So we should prove that $\sigma_n(B,A-\lambda I) = \sigma_n(B,A-\bar{\lambda}I)$ for every $\lambda \in \mathbb{C}$, where $\bar{\lambda}$ is the complex conjugate of λ . We know that for a matrix D such that $D \in \mathbb{C}^{n\times n}$ we have $\lambda(D^T) = \lambda(D)$ [12, page 267]. Also we know that the singular values of the matrices $(B,A-\lambda I)$ and $(B,A-\bar{\lambda}I)$ are the eigenvalues of the matrices $(B,A-\lambda I)^H(B,A-\lambda I)$ and $(B,A-\bar{\lambda}I)^H(B,A-\bar{\lambda}I)$, respectively. But

$$[(B,A-\lambda I)^H(B,A-\lambda I)]^T = (B,A-\lambda I)^T[(B,A-\lambda I)^H]^T$$

$$= (B,A-\lambda I)^T(B,A-\bar{\lambda}I)$$

$$= (B,A-\bar{\lambda}I)^H(B,A-\bar{\lambda}I)$$

since A and B are real. Thus

$$\sigma_i(B,A-\lambda I) = \sigma_i(B,A-\bar{\lambda}I) \quad , \quad i = 1(1)n \ . \qquad (26)$$

Remark (XI): In this remark we will calculate the $\dot{\sigma}_n(B,A-\lambda I)$, which will be useful for the esitmation of the derivative of the function $\sigma_n(B,A-\lambda I)$ wherever it can be defined. In general let $D \in \mathbb{C}^{m\times n}$ and $D = U\,\Sigma\,V^H$ be the singular value decomposition of D , then we have the following.

$$DV = U\Sigma \Rightarrow \qquad \text{(when these derivatives exist)}$$

$$\dot{D}V + D\dot{V} = \dot{U}\Sigma + U\dot{\Sigma} \Rightarrow$$

$$U^H\dot{D}V + U^HD\dot{V} = U^H\dot{U}\Sigma + U^HU\dot{\Sigma} \Rightarrow$$

$$U^H\dot{D}V + \Sigma V^H\dot{V} = U^H\dot{U}\Sigma + \dot{\Sigma} \qquad (27)$$

But we know that if a vector $x(t)$ has constant magnitude then it is orthogonal to its derivative, that is $\dot{x}^H x = 0$. So in (27) the diagonal elements of the matrices $\Sigma V^H \dot{V}$ and $U^H \dot{U} \Sigma$ are zero. So, from (27), we have

$$\dot{\sigma}_i = u_i^H \dot{D} v_i \tag{28}$$

Thus, in our case since $D = (B, A-\lambda I)$ we have from (28)

$$\dot{\sigma}_i(\lambda) = - u_i^H (0, I_n) v_i \quad , \quad i = 1(1)n \tag{29}$$

where u_i, v_i are the columns of U,V respectively.

We will first present an algorithm for real λ and then we will extend it for complex λ. Acutally our problem is the solution of the equation $\sigma_n(B, A-\lambda I) = 0$ with one unknown, where in the real case the root, if there exists any, will be in the interval

$$\Delta = [-\sigma_1' , -\sigma_n'] \cup [\sigma_n' , \sigma_1'] \tag{30}$$

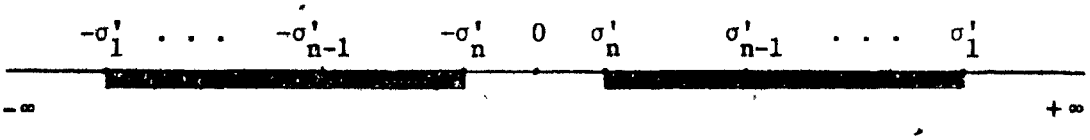


Fig. 4

Of course we do not have any formula for this equation but we can find as many values of the function $\sigma_n(B, A-\lambda I)$ as we want, and numerically in many cases that is what really counts. A first thought is to use Newton's method, but using Newton's method we may face the following problem.

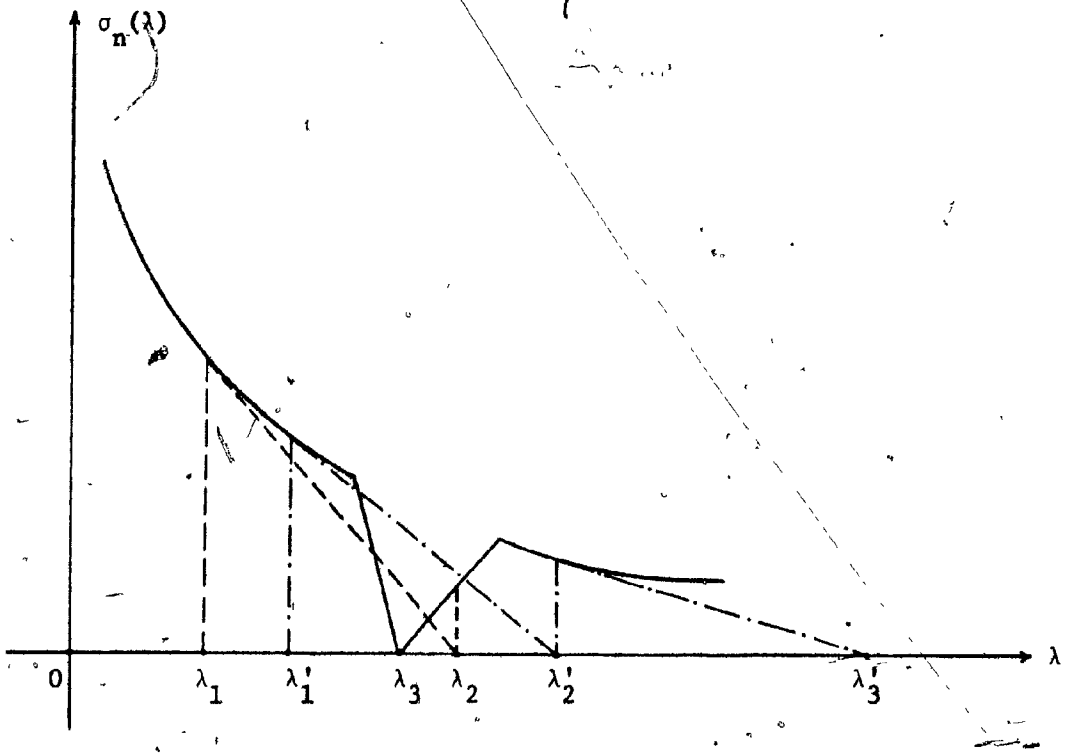


Fig. 5

As we can see from fig. 5, starting from λ_1 we will get the root, but starting from λ_1' we will miss it. So we cannot use Newton's method. We will try an iterative method of the following form $\lambda_{k+1} = \lambda_k + step_k$ where $\lambda_1 = -\sigma_1'$ and $step_k = \sigma_n(B, A-\lambda_k I)$, that is, the step, in order to go from λ_k to λ_{k+1}, is the value of the function, the root of which we are looking for, at λ_k. Using this step size and because of remark (IX) we will never miss the root. This can also be seen from fig. 6.

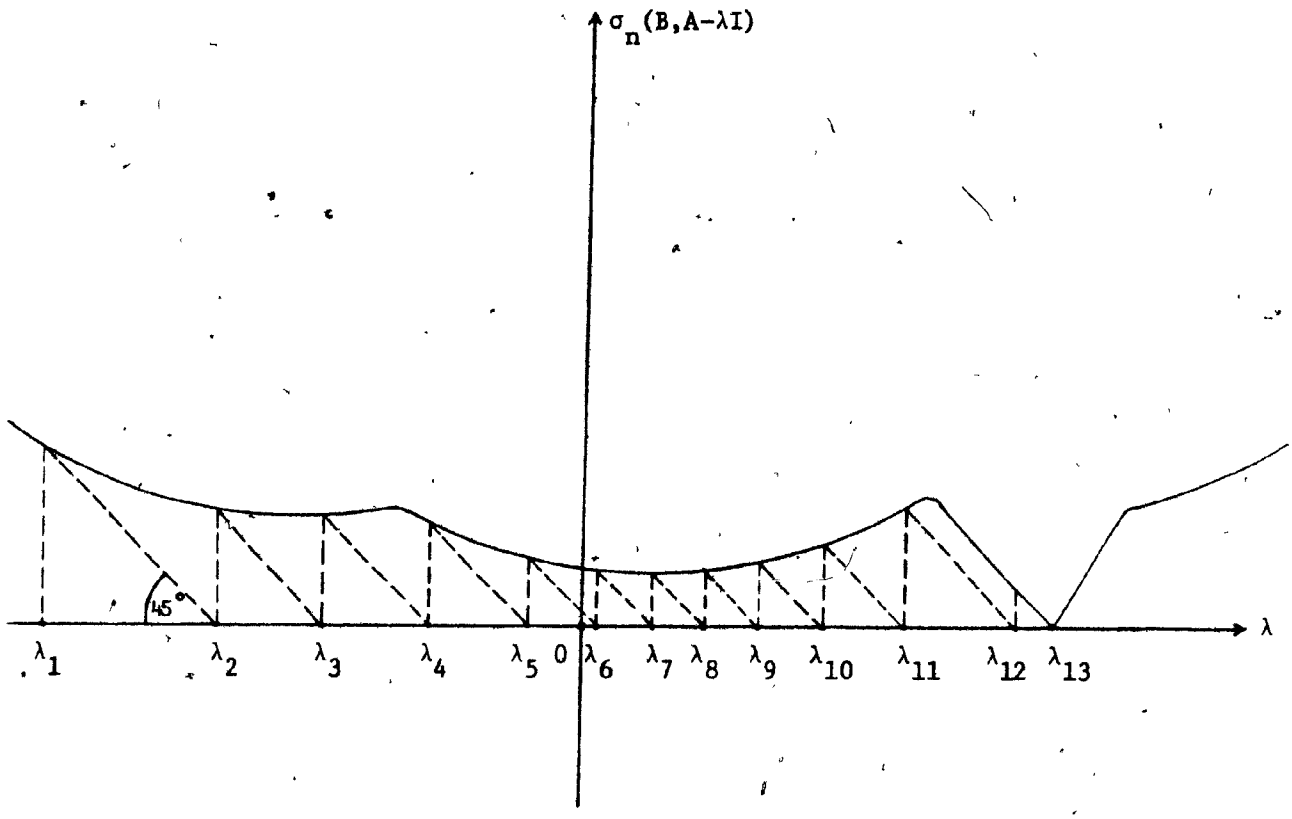


Fig. 6

Since we are looking for a possible root in the interval Δ, this algorithm can be quite fast if the system is quite controllable, that is, if μ is quite greater than zero, this can be observed from fig. 7.

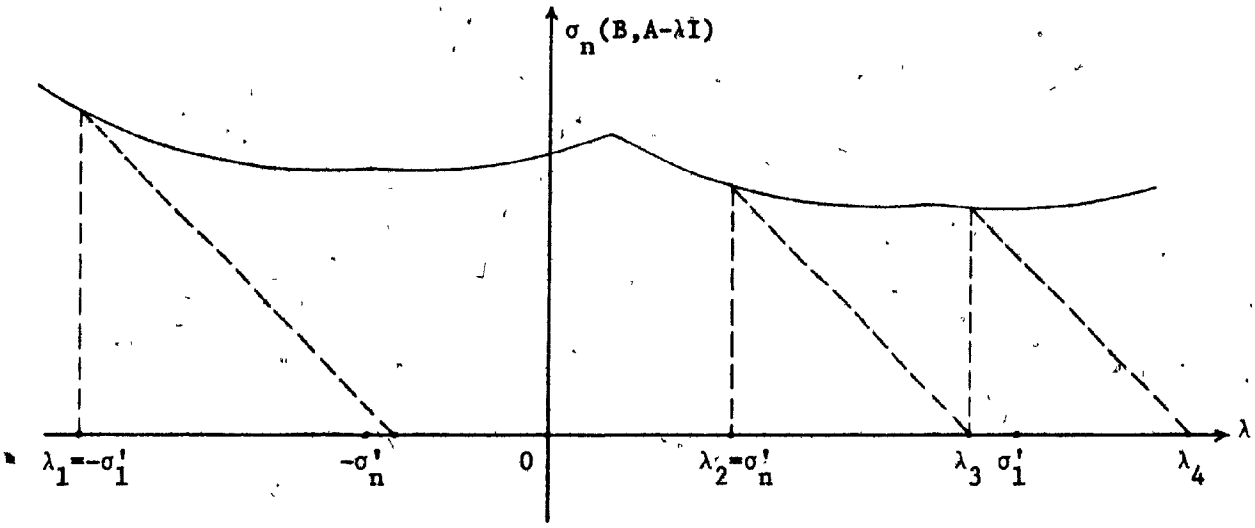


Fig. 7

But the same algorithm can be inefficient if the system is uncontrollable and we have reached the point where the function $\sigma_n(B,A-\lambda I)$ is almost linear, this can be seen from fig. 8.

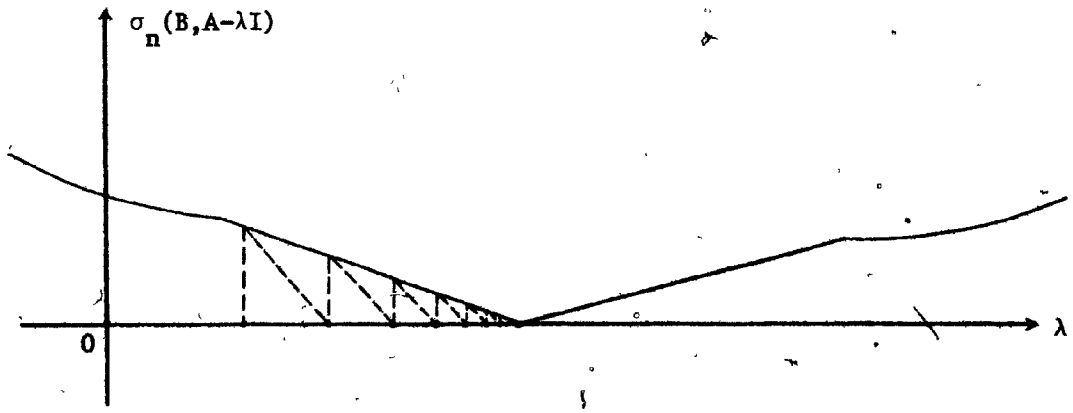


Fig. 8

In order to improve this situation we should take the fact $\dot{\sigma}(\lambda) \equiv \dot{\sigma}(B,A-\lambda I) \equiv \frac{d\sigma(B,A-\lambda I)}{d\lambda}$ into account. We can use the quantity $\frac{\sigma(\lambda_{k+1}) - \sigma(\lambda_k)}{\lambda_{k+1} - \lambda_k}$ as an estimate for $\dot{\sigma}(\lambda_k)$. This quantity will be quite

accurate for the calculation of the derivative when we have reached the point where the function $\sigma_n(B,A-\lambda I)$ is almost linear. At this point two consecutive points will have almost the same derivative as we can see from fig. 9. So this will be the criterion for detecting if we have reached the point where the function $\sigma_n(B,A-\lambda I)$ is almost linear.

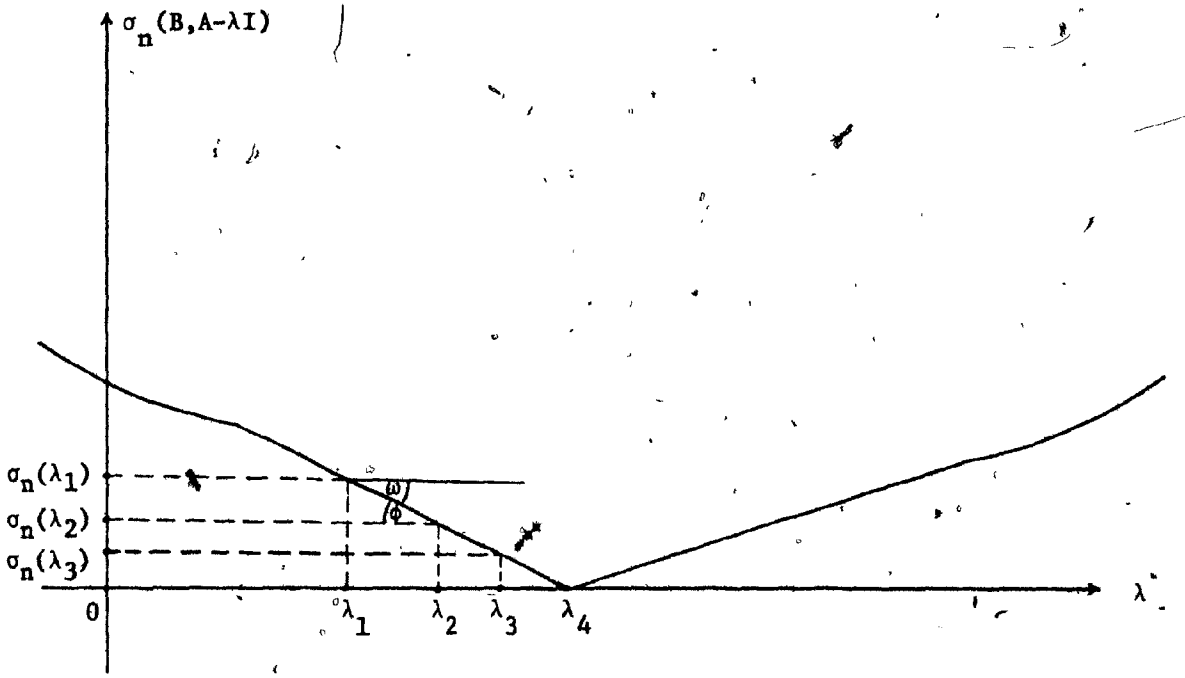


Fig. 9

From fig. 9 we can obviously see that $|\dot{\sigma}_n(\lambda_1) - \dot{\sigma}_n(\lambda_2)| < \delta$ where δ is an appropriate, small enough, positive real number. So from now on, until we reach the root we will use Newton's method, for example in fig. 9 we can have the following:

$$\lambda_4 = \lambda_2 - \frac{\sigma(\lambda_2)}{\dot{\sigma}(\lambda_2)}$$

So the algorithm for $\lambda \in \mathbb{R}$ is as follows:

Part 1 (Initialization)

A1: Calculate the singular values of (B,A) and let them be

$\sigma_1' \geq \sigma_2' \geq \ldots \geq \sigma_n'$;

A2: Set $\lambda := -\sigma_1'$; $\lambda_{old} := \lambda$;

A3: Evaluate $\sigma_n(B,A-\lambda I)$; and set $\sigma_{old} := \sigma_n(B,A-\lambda I)$;

A4: If $\sigma_{old} < \varepsilon\sigma_1'$, where ε is a small positive real number,

then system uncontrollable, stop;

else

A5: $\lambda := \lambda + \sigma_{old}$; $\lambda_{new} := \lambda$;

A6: If $\lambda > -\sigma_n'$ then go to A17

else

A7: Evaluate $\sigma_n(B,A-\lambda I)$; $\sigma_{new} := \sigma_n(B,A-\lambda I)$;

A8: If $\sigma_{new} < \varepsilon\sigma_1'$ then system uncontrollable, stop;

else

A9: $\dot{\sigma}_{old} := \dfrac{\sigma_{new} - \sigma_{old}}{\lambda_{new} - \lambda_{old}}$; $\sigma_{old} := \sigma_{new}$; $\lambda_{old} := \lambda_{new}$;

Part 2 (We search the interval $[-\sigma_1', -\sigma_n']$)

A10: $\lambda := \lambda + \sigma_{old}$; $\lambda_{new} := \lambda$;

A11: If $\lambda > -\sigma_n'$ then go to A17

else

A12: Evaluate $\sigma_n(B,A-\lambda I)$; $\sigma_{new} = \sigma_n(B,A-\lambda I)$;

A13: If $\sigma_n(B,A-\lambda I) < \varepsilon\sigma_1'$ then system uncontrollable, stop;

A14: $\dot{\sigma}_{new} := \dfrac{\sigma_{new} - \sigma_{old}}{\lambda_{new} - \lambda_{old}}$;

A15: If $|\dot{\sigma}_{new} - \dot{\sigma}_{old}| < \delta$ then go to A18

else

A16: $\sigma_{old} := \sigma_{new}$; $\lambda_{old} := \lambda_{new}$; $\dot{\sigma}_{old} := \dot{\sigma}_{new}$; go to A10;

Part 3 (We search the interval $[\sigma_n', \sigma_1']$)

A17: If $\lambda < \sigma_n'$ then $\lambda := \sigma_n'$; $\lambda_{old} := \lambda$; and go to A3 but instead of steps A6 and A11 we shall have the following step in both cases:

If $\lambda > \sigma_1'$ then system controllable, stop; else go to A10 but instead of step A11 we shall have the above mentioned step.

Part 4 (Newton's method)

A18: $\lambda := \lambda + \frac{\sigma_{new}}{\dot{\sigma}_{new}}$;

A19: If $|\lambda - \lambda_{new}| > M$ (*) then go to A16;

A20: $\lambda_{old} := \lambda_{new}$; $\lambda_{new} := \lambda$;

A21: Evaluate $\sigma_n(B,A-\lambda I)$; $\sigma_{old} := \sigma_{new}$; $\sigma_{new} := \sigma_n(B,A-\lambda I)$;

A22: If $\sigma_n(B,A-\lambda I) < \varepsilon\sigma_1'$ then system uncontrollable, stop

else $\dot{\sigma}_{new} := \frac{\sigma_{new} - \sigma_{old}}{\lambda_{new} - \lambda_{old}}$; go to A18;

Now we will extend the algorithm A for $\lambda \in \mathbb{C}$. As we saw in remark (X), we do not have to search all the ring but only the part of the anulus which is above the real axis. In order to be able to apply an algorithm similar to algorithm A we will first do the following:

(*) There might be some case where $|\dot{\sigma}_{new} - \dot{\sigma}_{old}| < \delta$ in A15 but we are not in a neighborhood of the possible root, so we must have a bound and a check built in for safety (see fig. 5).

Let us imagine a plane P which contains the axis $Re(\lambda)$ and is vertical to the complex plane. Now we take the projection of the surface which is above the upper half and is defined by the function $\sigma_n(B,A-\lambda I)$ onto the plane P. The curve we get on plane P is similar with the curve we had for $\lambda \in \mathbb{R}$ (see fig. 10 and fig. 11). The most important thing is that remark (IX) can be applied in this case too. So algorithm A can also be applied in this case with the following changes.

Let us assume that $\lambda_i = \alpha_i + j\beta_i$, where $j^2 = -1$, then we start with $\alpha_1 = -\sigma_1'$ and $\beta_1 = 0$, so $\lambda_1 = -\sigma_1'$. The corresponding value of the function is $\sigma_n^{(1)} = \sigma_n(B,A-\lambda_1 I)$. At the next step we have the following: $\alpha_2 = \alpha_1 + \sigma_n^{(1)}$, but we do not know β_2. So we will find $\sigma_n^{(2)}$ as the minimum of the curve, which lies on the plane perpendicular to the complex plane and contains the line segment $\alpha_2 k$ (see fig. 10), considering α_2 and k as boundaries of the curve, the minimum of which we are looking for. Thus for calculating $\sigma_n^{(2)}$ we have the following:

$$\sigma_n^{(2)} = \min_{\beta_2} \sigma_n[B,A-(\alpha_2 + j\beta_2)I] \text{ , where } \beta_2 \in [0, \sqrt{(\sigma_1')^2 - \alpha_2^2}\,]$$

see triange $\alpha_2 0k$ fig. 10

In general we have the following:

$$\sigma_n^{(i)} = \min_{\beta_i} \sigma_n[B, A-(\alpha_i + j\beta_i)I] \text{ ,} \tag{31}$$

$$\text{where } \beta_i \in \begin{cases} [\,0, \sqrt{(\sigma_1')^2-\alpha_i^2}\,] & \text{if } \alpha_i \in [-\sigma_1', -\sigma_n'] \cup [\sigma_n', \sigma_1'] \\ [\sqrt{(\sigma_n')^2-\alpha_i^2}\,, \sqrt{(\sigma_1')^2-\alpha_i^2}\,] & \text{if } \alpha_i \in (-\sigma_n', \sigma_n') \end{cases} \tag{32}$$

(see fig. 10).

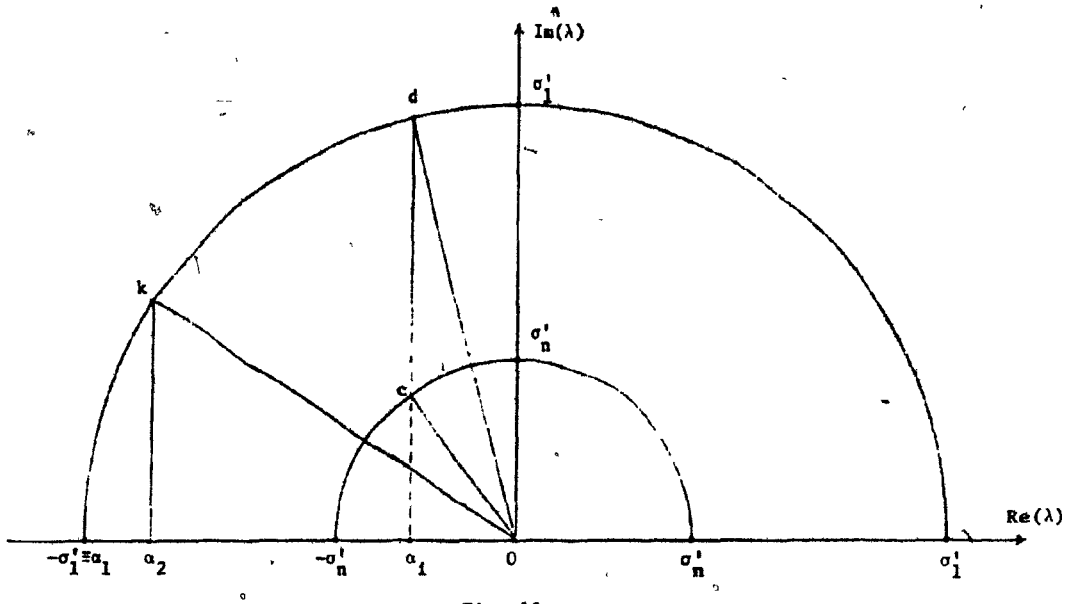


Fig. 10

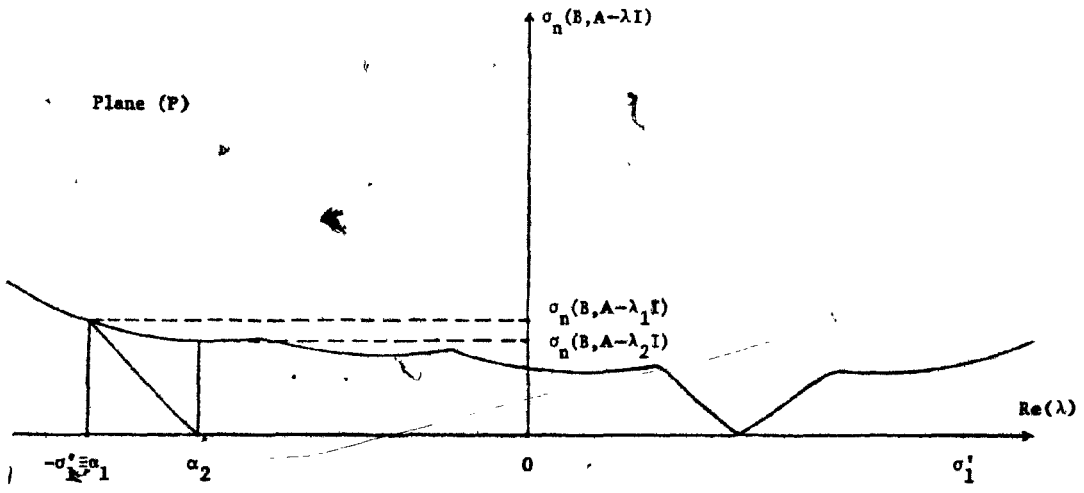


Fig. 11 (uncontrollable case)

So the algorithm for $\lambda \in \mathbb{C}$ is as follows:

Algorithm B

Part 1 (Initialization)

B1: Calculate the singular values of (B,A) and let them be $\sigma_1' \geq \sigma_2' \geq \ldots \geq \sigma_n'$;

B2: Set $\alpha := -\sigma_1'$; $\alpha_{old} := -\sigma_1'$;

B3: Evaluate $\sigma_n(B, A-\alpha I)$; $\sigma_{old} := \sigma_n(B, A-\alpha I)$;

B4: If $\sigma_{old} < \varepsilon\sigma_1'$ then system uncontrollable, stop

else

B5: $\alpha := \alpha + \sigma_{old}$; $\alpha_{new} := \alpha$;

B6: If $\alpha > \sigma_1'$ then system controllable, stop

else

B7: Evaluate $\min_{\beta} \sigma_n[B, A-(\alpha+j\beta)I]$ with $b_1 \leq \beta \leq b_2$, where b_1, b_2 are appropriate bounds (see (32));

Set $\sigma_{new} := \min_{\beta} \sigma_n[B, A-(\alpha+j\beta)I]$

B8: If $\sigma_{new} < \varepsilon\sigma_1'$ then system uncontrollable, stop

else

B9: $\dot{\sigma}_{old} := \dfrac{\sigma_{new} - \sigma_{old}}{\alpha_{new} - \alpha_{old}}$; $\sigma_{old} := \sigma_{new}$; $\alpha_{old} := \alpha_{new}$;

Part 2 (We search in the interval $[-\sigma_1', \sigma_1']$)

B10: $\alpha := \alpha + \sigma_{old}$; $\alpha_{new} := \alpha$;

B11: If $\alpha > \sigma_1'$ then system controllable, stop

else

B12: Evaluate $\min_{\beta} \sigma_n[B, A-(\alpha+j\beta)I]$ with $b_1 \leq \beta \leq b_2$, where b_1, b_2 are appropriate bounds (see (32))

Set $\sigma_{new} := \min_{\beta} \sigma_n[B, A-(\alpha+j\beta)I]$;

B13: If $\sigma_{new} < \varepsilon\sigma_1'$ then system uncontrollable, stop

else

B14: $\dot{\sigma}_{new} := \dfrac{\sigma_{new} - \sigma_{old}}{\alpha_{new} - \alpha_{old}}$;

B15: If $|\dot{\sigma}_{new} - \dot{\sigma}_{old}| < \delta$ then go to B17

else

B16: $\sigma_{old} := \sigma_{new}$; $\alpha_{old} := \alpha_{new}$; $\dot{\sigma}_{old} := \dot{\sigma}_{new}$; go to B10;

Part 3 (Newton's method)

B17: $\alpha := \alpha + \dfrac{\sigma_{new}}{\dot{\sigma}_{new}}$;

B18: If $|\alpha - \alpha_{new}| > M$ then go to B16:

B19: $\alpha_{old} := \alpha_{new}$; $\alpha_{new} := \alpha$;

B20: Evaluate $\min_{\beta}[B, A-(\alpha+j\beta)I]$ with $b_1 \le \beta \le b_2$;

$\sigma_{old} := \sigma_{new}$; $\sigma_{new} := \min_{\beta}[B, A-(\alpha+j\beta)I]$;

B21: If $\sigma_{new} < \varepsilon\sigma_1'$ then system uncontrollable, stop

else $\dot{\sigma}_{new} := \dfrac{\sigma_{new} - \sigma_{old}}{\alpha_{new} - \alpha_{old}}$; go to B17

Note: Obviously the algorithms A and B are quite expensive because we need to evaluate the function $\sigma_n(B,A-\lambda I)$ many times. We feel there may be a much more simple approach to this problem, but have not yet found one.

Note: In page 32 it is pointed out that in order to decide whether a system is controllable or not it is adequate to see whether the equation $\sigma_n(B,A-\lambda I) = 0$ has any root or not, with respect to $\lambda \in \mathbb{C}$. But instead of finding the roots of $\sigma_n(B,A-\lambda I) = 0$, if any, we can solve the optimization problem

$$\min_{\lambda} \sigma_n(B, A-\lambda I) \quad , \quad |\lambda| \in [\sigma_n', \sigma_1'] \tag{32α}$$

which will not only tell us if the system is controllable or not, but in the case of a controllable system it will also tell us how far the system is from the nearest uncontrollable one, that is, it will give us the distance of the system from the nearest uncontrollable one. But we have already pointed out that even the solution of the equation

$$\sigma_n(B, A-\lambda I) = 0 \tag{32β}$$

is quite expensive, so the minimization problem (32α) will be more expensive, at least, with the methods we have so far.

CHAPTER 3

(I) Introduction

(1) Open-loop (Nonfeedback) and closed-loop (Feedback) control systems.

A control action is required to take a controllable dynamic system from a given initial state to any other desired state. The two basic ways for carrying out such a control are the following:

(a) Open-loop control or nonfeedback:

When the determination of the control action is independent of output measurements on the system.

(b) Closed-loop control or feedback:

When the determination of the control action is based on the output behavior of the system.

Closed-loop control is often preferable to open-loop control. The following example will illustrate the difference between closed-loop and open-loop controls and will also illustrate some of the advantages of closed-loop control.

Example: Consider a home heating system. Also consider two different control systems. The first (open-loop) turns the heating on every 30 minutes and keeps it on for 20 minutes. The second (closed-loop) has a thermostat which turns on the heating when the temperature of the home drops under 20°C and turns it off when the temperature exceeds 25°C.

The advantages of the closed-loop system here over the open-loop system are obvious. While the closed-loop system will keep the temperature between

20°C - 25°C the open-loop system may make the home too hot or too cold. For more details and examples see [3], [7].

(2) Stability problem in dynamical systems

Stability in dynamical systems is defined with respect to equilibrium points, so we first need to give the definition of the equilibrium point.

Definition (VII): A vector $\tilde{x}$ is an equilibrium point of a dynamical system when it has the following property; when the state of the system reaches $\tilde{x}$ it remains equal to $\tilde{x}$ forever (as time approaches infinity), when there is no control input.

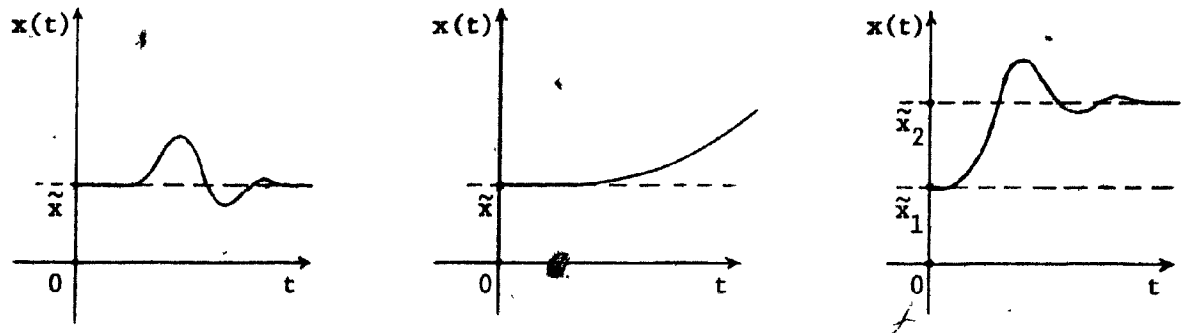
An equilibrium point is stable if when the state vector is moved slightly away from that point, it tends to return to it, or at least does not keep moving further away (marginal stability). The following example will illustrate this point.

Example: Let a stick be perfectly aligned with the vertical. Then the dynamical system which describes the movements of the stick is in an equilibrium point. If the stick is balanced on its bottom end, then the slighest impulse input will destroy the balance and the stick not only is not going to return to the same equilibrium point but it will keep moving further away. So if the stick is balanced on its bottom end, it is in an unstable equilibrium point. If on the other hand it is hanging from a support at the top end and an impulse input moves it from this equilibrium point then the stick tends to return to it. So in this case the stick is in a stable equilibrium point.

Now we will define when an equilibrium point is stable, marginally stable, or unstable.

Definition (VIII): An equilibrium point $\tilde{x}$ of a dynamical system is called stable when in response to an impulse input, if the system is at the equilibrium point $\tilde{x}$, the output of the system tends to return to $\tilde{x}$ as time increases. Otherwise $\tilde{x}$ is called either an unstable or marginally stable equilibrium point according to whether the output of the system goes to infinity or settles at another equilibrium point, see [7].

In fig. 12 we give one example for each case.



a. Stable equilibrium point

b. Unstable equilibrium point

c. Marginally stable equilibrium point

Fig. 12

Remark (XII): Consider the systems

$$\dot{x} = Ax + b \tag{33}$$

$$x_{k+1} = Ax_k + b \tag{34}$$

and let $\tilde{x}$ be an equilibrium point for these two systems, then from (33) we have

$$\left.\begin{aligned} \dot{x} &= Ax + b \\ \dot{\tilde{x}} &= A\tilde{x} + b \end{aligned}\right\} \Rightarrow (\dot{x} - \dot{\tilde{x}}) = A(x - \tilde{x}) \tag{35}$$

from (34), we have

$$\left.\begin{aligned} x_{k+1} &= Ax_k + b \\ \tilde{x} &= A\tilde{x} + b \end{aligned}\right\} \Rightarrow (x_{k+1} - \tilde{x}) = A(x_k - \tilde{x}) \ . \tag{36}$$

Setting $z = x - \tilde{x}$ and $y_k = x_k - \tilde{x}$, the equations (35) and (36) give, respectively

$$\dot{z} = Az \tag{37}$$

and

$$y_{k+1} = Ay_k \ . \tag{38}$$

So, it is clear that the conditions

$$\lim_{t \to +\infty} x(t) = \tilde{x} \qquad \text{and} \qquad \lim_{k \to +\infty} x_k = \tilde{x}$$

are equivalent to the conditions

$$\lim_{t \to +\infty} z(t) = 0 \qquad \text{and} \qquad \lim_{k \to +\infty} y_k = 0$$

respectively, that is, the condition for $x(t)$ and x_k to tend to $\tilde{x}$ in (33) and (34), respectively, is equivalent to the condition for $z(t)$ and y_k to tend to zero in the homogeneous sytems (37), (38), respectively.

So, instead of looking for the stability of (33) and (34), we will look for the stability of (37) and (38) respectively.

(a) Stability for discrete-time systems

Consider the system

$$x_{k+1} = Ax_k \tag{39}$$

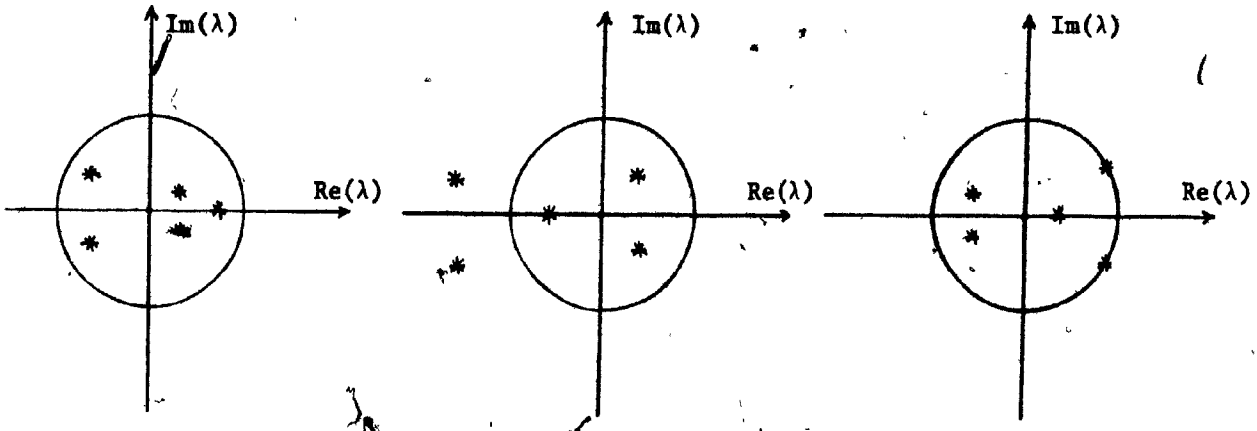
then $x_{k+1} = A^k x_1$ for the given initial state $x_1 \in \mathbb{R}^n$. Therefore $\lim_{k \to +\infty} x_{k+1} = 0 \iff \lim_{k \to +\infty} A^k x_1 = 0 \iff |\lambda_i| < 1$ for $i = 1(1)n$ where $\lambda_i \in \lambda(A)$, $i = 1(1)n$ and for every initial state $x_1 \in \mathbb{R}^n$ [3, page 155]. Thus system (34) is stable if and only if the magnitude of each of the eigenvalues of A is less than one (see figure 13).

(b) Stability for continuous-time systems

Consider the system

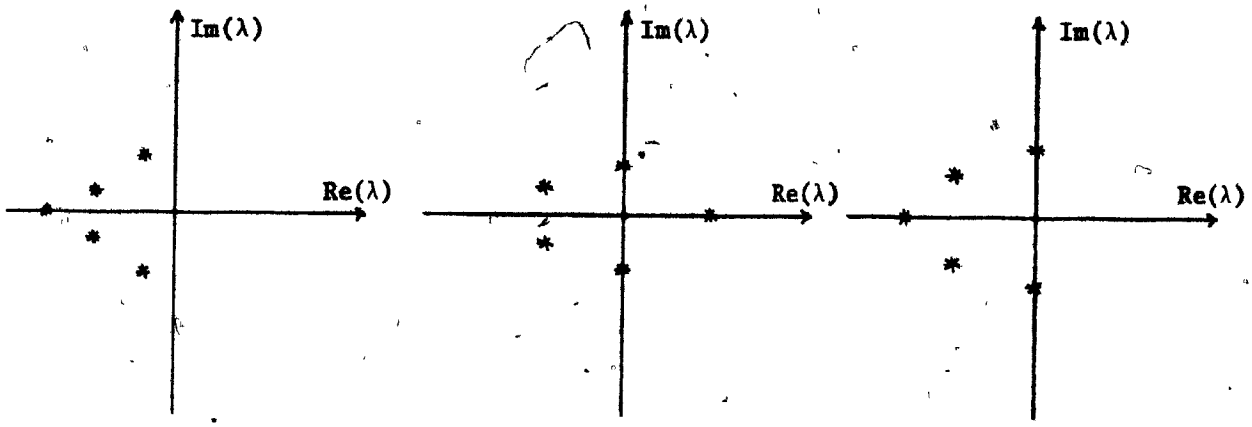
$$\dot{x}(t) = Ax(t) \tag{40}$$

then $x(t) = e^{At}x(0)$ for the given initial stage $x(0) \in \mathbb{R}^n$. So, $\lim_{t \to +\infty} x(t) = 0 \iff \lim_{t \to +\infty} e^{At}x(0) = 0 \iff \mathrm{Re}(\lambda_i) < 0$ for $i = 1(1)n$ where $\lambda_i \in \lambda(A)$, $i = 1(1)n$ and for every initial state $x(0) \in \mathbb{R}^n$ [3, page 157]. Therefore system (33) is stable if and only if the real parts of each of the eigenvalues of A are less than zero (see figure 14).



a. Stable system b. Unstable system c. Marginally stable system

Fig. 13 Discrete-time case



a. Stable system b. Unstable system c. Marginally stable system

Fig. 14 Continuous-time case

(II) On pole assignment in single-input controllable linear systems

The problem of pole assignment or eigenvalue allocation for controllable linear systems arises as follows: consider the controllable system

$$\dot{x} = Ax + Bu \tag{41}$$

which is made into a closed-loop system by defining

$$u = Fx \,, \tag{42}$$

where $F \in \mathbb{R}^{m\times n}$. Then (41), in view of (42), gives

$$\dot{x} = (A + BF)x \,. \tag{43}$$

We want to choose F such that the system (43) is stable at every equilibrium point $\tilde{x}$. So we should choose F such that the real parts of each of the eigenvalues of the matrix $A + BF$ are less than zero.

Note: Instead of a time-continuous system, we can also have a discrete-time system. The only difference from continuous-time systems is that in the discrete case we will choose F such that the magnitude of all the eigenvalues of the matrix $A + BF$ is less than one.

So the pole assignment or eigenvalue allocation probelm is as follows:

> Given $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ find $F \in \mathbb{R}^{m\times n}$ such that the matrix $A + BF$ has a desired set of eigenvalues.

Luenberger in [3] gives a theorem (p. 299) based on a controllability canonical form also discussed in the same text which can be used to derive an

algorithm to solve the above problem when $m = 1$, that is, when we have single-input controllable linear system. But this algorithm is based on the calculation of the coefficients of the characteristic polynomial of A, which can be achieved by calculating the eigenvalues of A. But the calculation of the eigenvalues of a matrix A, when A is not symmetric, can be an ill-conditioned problem [13], [12], [2]. The same problem (single-input) is also considered in [32]. The generalization of the canonical forms, discussed in [3], to multi-input linear dynamical systems are discussed by Luenberger in [10]. One of these forms might be useful in the eigenvalue allocation problem (e.a.p.) for the multi-input case, but it should be pointed out that it is quite difficult to make an algorithm which solves the e.a.p. using this canonical form, and one of the reasons is that in general, there is not a unique way to bring the system to such a canonical form. One particular derivation of a canonical form of a multi-input dynamical system is given in [11] where a different approach from [10] is used. For a more explicit way of bringing a system to its canonical form see Wonham [9], where the fundamental result of pole-assignability was presented. The fact that the e.a.p. is quite difficult for the multi-input case, is emphasized in [24] where a geometric viewpoint of the problem is given. An algorithm which partially solved the e.a.p. was given by Davison [27], where he proved the following:

Let the system

$$\left.\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned}\right\} \qquad (43\alpha)$$

where x, u, A and B are as we have already described, and $C \in R^{r \times n}$, $y \in R^{r}$, $(r \leq n)$. Then if (43α) is controllable and $\operatorname{rank}(C) = r$, there

is a matrix $F \in \mathbb{R}^{m\times r}$ such that r eigenvalues can be assigned to the matrix $A + BFC$. An algorithm is given. This result was further extended by Davison and Chatterjee [28] and by Sridhar and Lindorff [29] as follows: Under the same conditions of [27] $\max(r,m)$ eigenvalues can be assigned to $A + BFC$. Numerical difficulties of the algorithm given in [27], when $n >> 10$ led Davison and Chow to extend the result of [27] to systems where $n >> 10$ [31]. An algorithm is given. The results of [27], [28], [29] are further extended by Davison and Wang [30] where they show that under the same conditions of [27] $\min(n, m+r-1)$ eigenvalues can be assigned to $A + BFC$. An algorithm is also given. A similar extension to [27], [28], [29] is given by Kimura [25] where he proves that under the same conditions of [27] and $n \leq m + r - 1$, an arbitrary set of distinct eigenvalues can be assigned to $A + BFC$. In this paper, different from the conventional (up to that time) approach using the characteristic equation, an approach based on the properties of the eigenvectors is used. This result is further extended by Kimura in [26], where an algorithm is also given.

Although the above mentioned works are important theoretically, the resulting algorithms have usually not been designed with as wide an understanding of numerical difficulties as we have today. The main aim of this thesis is to present algorithms which are reliable numerically, as well as theoretically, and to do this we start with the most simple form of the problem.

Note: In this section we will deal with single-input controllable systems, that is, $m = 1$ or $B \equiv b$, $b \in \mathbb{R}^n$. So that $F \equiv f^T$, $f \in \mathbb{R}^n$. The algorithm we will present will calculate an n-dimensional real vector f such that given an $n\times n$ matrix A and an n-dimensional real vector b the matrix $A + bf^T$ has a desired set of eigenvalues $\lambda_1,\lambda_2,\ldots,\lambda_n$.

Remark (XIII): For a controllable system we can always find an $F \in \mathbb{R}^{m\times n}$ such that the matrix $A + BF$ has any desired set of eigenvalues. A proof for this can be found in [9], [10] or [11] (see also earlier here, page 14).

Remark (XIV): The problem of finding a matrix F such that the matrix $A + BF$ has any desired set of eigenvalues is equivalent to the problem of finding a matrix $\tilde{F}$ such that the matrix $\tilde{A} + \tilde{B}\tilde{F}$ has the same desired set of eigenvalues as $A + BF$, where $\tilde{A} = Q^TAQ$, $\tilde{B} = Q^TBP$, $\tilde{F} = P^TFQ$ with P and Q orthogonal matrices. The above mentioned property holds because

$$\tilde{A} + \tilde{B}\tilde{F} = Q^TAQ + (Q^TBP)(P^TFQ) = Q^T(A + BF)Q .$$

So,

$$\lambda(\tilde{A} + \tilde{B}\tilde{F}) = \lambda(A + BF) .$$

If P, Q are chosen such that $\tilde{A}$ and $\tilde{B}$ are of the form (6), then the problem is much simpler, for example in our case where $m = 1$ we have the following:

$$\tilde{A} = \begin{pmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1,n-1} & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2,n-1} & \alpha_{2n} \\ & \alpha_{32} & \cdots & \alpha_{3,n-1} & \alpha_{3n} \\ & & \ddots & \vdots & \vdots \\ & & & \alpha_{n,n-1} & \alpha_{nn} \end{pmatrix} , \quad \tilde{b} = \begin{pmatrix} \alpha_{10} \\ 0 \\ \vdots \\ 0 \end{pmatrix} , \quad \tilde{f}^T = (\phi_1, \phi_2, \ldots, \phi_n)$$

where $\alpha_{i,i-1} \neq 0$ for $i = 1(1)n$, because the system defined by $(\tilde{A},\tilde{b})$ is considered to be controllable. So,

$$\tilde{A} + \tilde{b}\tilde{f}^T = \begin{pmatrix} \alpha_{11}+\alpha_{10}\phi_1 & \alpha_{12}+\alpha_{10}\phi_2 & \cdots & \alpha_{1,n-1}+\alpha_{10}\phi_{n-1} & \alpha_{1n}+\alpha_{10}\phi_n \\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2,n-1} & \alpha_{2n} \\ & \alpha_{32} & \cdots & \alpha_{2,n-1} & \alpha_{3n} \\ & & \ddots & \vdots & \vdots \\ & & & \alpha_{n,n-1} & \alpha_{nn} \end{pmatrix}.$$

Thus the original problem can be stated as follows:
Given an upper Hessenberg matrix $A \in \mathbb{R}^{n\times n}$ with elements $\alpha_{i,j}$, where $i = 1(1)n$, $j = 1(1)n$, and $\alpha_{i,i-1} \neq 0$, $i = 2(1)n$, we are concerned in choosing real α_{1j}, $j = 1(1)n$ (that is, we are concerned in choosing the first row of A) in order to give A a certain desired set of eigenvalues which may occur in complex conjugate pairs. Thus from now on we will talk about determining the first row of A, rather than determining $\tilde{f}$ in $\tilde{A} + \tilde{b}\tilde{f}^T$.

We shall now describe a method which solves this problem. The description will be divided into two parts: as an easy introduction the first part will describe how to deal with real eigenvalues, while the second will deal with the more difficult case of eigenvalues which occur in complex conjugate pairs. The first part also will be divided into two other parts, the first part will describe an explicitly shifted method, while in the second part we shall show how the same thing can be done implicitly. As in methods for

solving the eigenproblem, the implicitly shifted approach will not only be more accurate in some cases, but will also allow the complex case to be handled in real arithmetic.

From now until the end of this chapter we will use a notation which we describe immediately.

1) Sometimes the elements of the matrices will be represented by either * , x , 0 or blank, meaning that the corresponding element is a variable, a known scalar, or in the last two cases is zero respectively. By "variable" we mean an element that has not yet been described. For example usually the first row of our upper Hessenberg matrix A is "variable".

Definition (IX) [12], [14]: Let $a = (\alpha_1,\ldots,\alpha_i,\ldots,\alpha_j,\ldots,\alpha_n)^T$ be an n-dimensional vector with $\alpha_j \neq 0$, let also

$$R = \begin{array}{c} \\ \\ \\ i \rightarrow \\ \\ j \rightarrow \\ \\ \end{array} \begin{array}{c} \begin{array}{ccccccc} & & & i & & j & \\ \downarrow & & & \downarrow & & \downarrow & \end{array} \\ \begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & & c & & s & \\ & & & & 1 & & \\ & & & & & \ddots & \\ & & & -s & & c & \\ & & & & & & 1 \\ & & & & & & & \ddots \\ & & & & & & & & 1 \end{bmatrix} \end{array}$$

be an orthogonal matrix such that the vector

$$Ra = \begin{pmatrix} \alpha_1 \\ \vdots \\ i \longrightarrow \alpha_i c + \alpha_j s \\ \vdots \\ j \longrightarrow \alpha_j c - \alpha_i s \\ \vdots \\ \alpha_n \end{pmatrix}$$

has its jth element zero. Thus c,s should be chosen such that if $\rho = \sqrt{\alpha_i^2 + \alpha_j^2}$ then $c = \frac{\alpha_i}{\rho}$, $s = \frac{\alpha_j}{\rho}$. Then matrix R is a rotation matrix in the i,j plane.

2) The following notation for a 3×3 matrix

$$\begin{pmatrix} * & * & * \\ x & x & x \\ & \underbrace{x \quad x}_{1} & \end{pmatrix} \rightarrow \begin{pmatrix} * & * & * \\ x & x & x \\ & x & x \end{pmatrix}_{1'} \rightarrow \begin{pmatrix} * & * & * \\ x & x & x \\ (1') & \overset{2}{\curvearrowright} x & x \end{pmatrix} \rightarrow \begin{pmatrix} * & * & * \\ x & x & x \\ & x & x \end{pmatrix}_{2'} \rightarrow \begin{pmatrix} * & * & * \\ * & * & * \\ & x & x \end{pmatrix}$$

means that, first we multiply our matrix from the right by a rotation matrix which combines the second and third columns without eliminating any element, then we apply the transpose of the same rotation from the left combining the second and third row, so a non-zero element is introduced at the place (3,1). After this the element (3,1) is eliminated into the element (3,2) by applying another rotation matrix from the right, and

finally the transpose of the same rotation matrix is applied from the left and it introduces variable elements at the second row of the matrix.

<u>Note</u>: All the methods we will describe in the following have two main parts, the forward sweep and the backward sweep.

(1) <u>The case of real eigenvalues</u>

(a) <u>Explicitly shifted method</u>:

Let A be the $n \times n$ upper Hessenberg real matrix with given real eigenvalues λ_i , $i = 1(1)n$, but unknown first row, also note that if $\alpha_{i,j}$, $i = 1(1)n$, $j = 1(1)n$ are the elements of A , then $\alpha_{i,i-1} \neq 0$, $i = 2(1)n$. Then in the forward sweep we do the following: Let $A_1 \equiv A$ then we start with the first eigenvalue λ_1 . We want the matrix $A_1 - \lambda_1 I$ to be singular, so we choose an orthogonal $n \times n$ matrix Q_1 such that the matrix $Q_1^T(A_1-\lambda_1 I)Q_1$ is clearly singular. Let us consider the following example with $n = 3$ $A_1-\lambda_1 I \equiv \begin{pmatrix} * & * & * \\ x & x & x \\ & x & x \end{pmatrix}$ then we multiply $A_1-\lambda_1 I$ by Q_1 from from the right, where Q_1 is a product of appropriate rotation matrices determined such that $(A_1-\lambda_1 I)Q_1 = R_1$ where R_1 is upper triangular, that is,

$$(A_1-\lambda_1 I)Q_1 \equiv \begin{pmatrix} * & * & * \\ \textcircled{x} \overset{1}{\longrightarrow} & x & x \\ & \textcircled{x} \overset{2}{\longrightarrow} & x \end{pmatrix} \rightarrow \begin{pmatrix} * & * & * \\ & x & x \\ & & x \end{pmatrix} \qquad \text{(M1)}$$

Since $\alpha_{i,i-1} \neq 0$, $i = 2(1)n$, the (i,i) elements of (M1) will be nonzero, $i = 2(1)n$. Thus the element $(1,1)$ of the matrix (M1) must be zero for (M1) to be singular; but since this element is variable we set it equal to zero, thereby using some of our freedom in choosing the first row of A .

Thus (M1) becomes $\begin{pmatrix} 0 & * & * \\ 0 & x & x \\ & 0 & x \end{pmatrix} \equiv (A_1-\lambda_1 I)Q_1$.

Then we multiply $(A_1-\lambda_1 I)Q_1$ from the left by Q_1^T and we get

$$Q_1^T(A_1-\lambda_1 I)Q_1 \equiv \begin{pmatrix} 0 & * & * \\ 0 & x & x \\ & 0 & x \end{pmatrix} \begin{matrix} \\ 2' \\ 1' \end{matrix} + \begin{pmatrix} 0 & * & * \\ 0 & * & * \\ 0 & x & x \end{pmatrix} \qquad (M2)$$

We set the row vector which is formed by the elements $(1,2)$, $(1,3)$ of (M2) equal to b_1^T and the matrix which is formed by the elements $(2,2)$, $(2,3)$, $(3,2)$, $(3,3)$ equal to $A_2-\lambda_1 I$. So we have the following:

$$Q_1^T(A_1-\lambda_1 I)Q_1 = \begin{pmatrix} 0 & b_1^T \\ 0 & A_2-\lambda_1 I \end{pmatrix} \Rightarrow Q_1^T A_1 Q_1 = \begin{pmatrix} \lambda_1 & b_1^T \\ 0 & A_2 \end{pmatrix} \qquad (44)$$

From relation (44) we can obviously see that $\lambda(A_1) = \{\lambda_1\} \cup \lambda(A_2)$. Note that the rotations in (M1) were nontrivial since $\alpha_{i,i-1} \neq 0$, $i = 2(1)n-1$ and the $(2,2)$ element of (M1) is nonzero, as a result the $(3,2)$ element in (M2) is nonzero and A_2 has the same form as A_1 (the general case will be proved in remark (XV)).

In the next step we will work with A_2, the first row of which can be chosen arbitrarily and we will choose an orthogonal $(n-1)\times(n-1)$ matrix Q_2 such that the matrix $Q_2^T(A_2-\lambda_2 I)Q_2$ is singular. At the kth step, we choose an orthogonal $(n-k+1)\times(n-k+1)$ matrix, Q_k, such that the matrix $Q_k^T(A_k-\lambda_k I)Q_k$ is singular, that is,

$$Q_k^T(A_k-\lambda_k I)Q_k = \begin{pmatrix} 0 & b_k^T \\ 0 & A_{k+1}-\lambda_k I \end{pmatrix} \Rightarrow Q_k^T A_k Q_k = \begin{pmatrix} \lambda_k & b_k^T \\ 0 & A_{k+1} \end{pmatrix}, \quad k=1(1)n-1 \tag{45}$$

so that $\lambda(A_k) = \{\lambda_k\} \cup \lambda(A_{k+1})$ for $k = 1(1)n-1$.

At the final step, when $k = n-1$, we choose an orthogonal 2×2 matrix Q_{n-1} such that

$$Q_{n-1}^T(A_{n-1}-\lambda_{n-1}I)Q_{n-1} = \begin{pmatrix} 0 & \beta_{n-1} \\ 0 & A_n-\lambda_{n-1}I \end{pmatrix} \Rightarrow$$

$$Q_{n-1}^T A_{n-1} Q_{n-1} = \begin{pmatrix} \lambda_{n-1} & \beta_{n-1} \\ 0 & A_n \end{pmatrix} \tag{46}$$

But the matrix A_n is well known because it is a scalar, that is, $A_n \in \mathbb{R}$ and since it is a variable we set it equal to λ_n.

Now we will go backwards and we will be calculating one by one the first row of the matrices A_k for $k = n-1(-1)1$. Finally we will calculate the first row of A, since $A \equiv A_1$.

Now we will describe the kth step of the backward sweep. At the kth step of the backward sweep we have from (45)

$$Q_k^T(A_k-\lambda_k I)Q_k = \begin{pmatrix} 0 & b_k^T \\ 0 & A_{k+1}-\lambda_k I \end{pmatrix} \tag{47}$$

From the previous k-1 steps in the backward sweep we will know the first row of A_{k+1} and so the first row a_{k+1}^T of $A_{k+1}-\lambda_k I$ and we want to evaluate the first row a_k^T of $A_k-\lambda_k I$ and from that the first row of A_k. Let

$$Q_k = P_1 P_2 \cdots P_{n-k} \tag{48}$$

where P_i, $i = 1(1)n-k$ are the known rotation matrices (see Definition (IX)). We know that first we applied the transformation Q_k from the right, and since this transformation combines the columns of $A_k-\lambda_k I$, we see that every one of the rotation matrices P_i, $i = 1(1)n-k$ affects the first row of $A_k-\lambda_k I$. But the same thing does not happen with Q_k^T, which combines the rows of $A_k-\lambda_k I$. As we can see, the only rotation matrix of Q_k^T which affects the first row of $A_k-\lambda_k I$ is the last one, that is P_{n-k}^T, because P_{n-k}^T combines the first with the second row of (the altered) $A_k-\lambda_k I$. Let

$$\begin{pmatrix} e_1^T \\ e_2^T \end{pmatrix} P_{n-k-1}^T \cdots P_1^T(A_k-\lambda_k I)Q_k = \begin{pmatrix} 0 & \bar{a}_k^T \\ 0 & a'^T_k \end{pmatrix} \tag{49a}$$

Note that a'^T_k will be available because at the corresponding stage of the forward sweep we will not combine it with the first row which contains variable elements.

Let $$P^T_{n-k} = \begin{pmatrix} c_k & s_k & & \\ -s_k & c_k & & \\ & & 1 & \\ & & & \ddots \\ & & & & 1 \end{pmatrix}$$, then we have (49β)

$$\begin{pmatrix} c_k & s_k \\ -s_k & c_k \end{pmatrix} \begin{pmatrix} 0 & \bar{a}^T_k \\ 0 & a'^T_k \end{pmatrix} = \begin{pmatrix} 0 & b^T_k \\ 0 & a^T_{k+1} \end{pmatrix} \triangleq \begin{pmatrix} e^T_1 \\ e^T_2 \end{pmatrix} \begin{pmatrix} 0 & b^T_k \\ 0 & A_{k+1} - \lambda_k I \end{pmatrix} \Rightarrow \quad (50)$$

$$c_k a'^T_k - s_k \bar{a}^T_k = a^T_{k+1} \quad \Longleftrightarrow$$

$$\bar{a}^T_k = \frac{c_k a'^T_k - a^T_{k+1}}{s_k} \quad , \text{ since } \quad s_k \neq 0 \ , \qquad (51)$$

which determines the variable row $\bar{a}^T_k$. But from (49α) and the form of the P_i ,

$$e^T_1 (A_k - \lambda_k I) Q_k = a^T_k Q_k = (0 \ , \ \bar{a}^T_k)$$

So

$$a^T_k = (0 \ , \ \bar{a}^T_k) Q^T_k \qquad (52)$$

From (52) the first row of A_k is

$$e_1^T A_k = (0 \ , \ \bar{a}_k^T) Q_k^T + \lambda_k e_1^T \qquad (52\alpha)$$

Thus from knowing the first row of A_{k+1} we can find the first row of A_k. But we know the first (only) row of A_n in (46), and so we can go backwards to finally produce the first row of $A \equiv A_1$, our desired result.

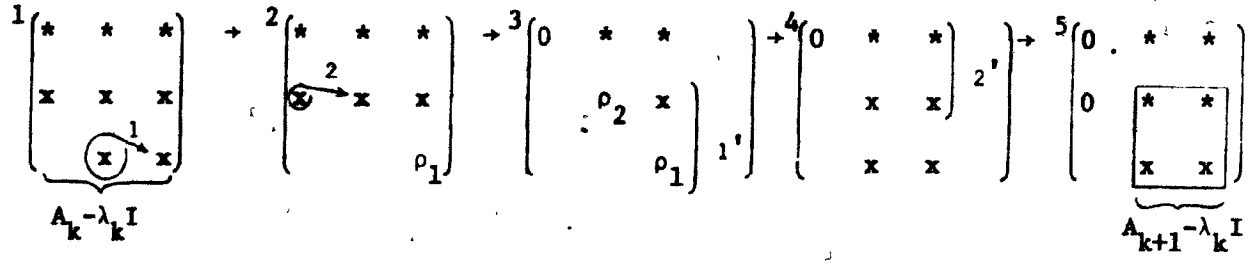
Comment: From (51) we can see that if s_k is relatively small then we may get into numerical trouble. So from now until the end of this section we will comment on this possible trouble.

Definition (X): Let $A \in \mathbb{C}^{n\times n}$ be an upper Hessenberg matrix with elements α_{ij}, $i = 1(1)n$, $j = 1(1)n$. Then we say A is unreduced if $\alpha_{i+1,i} \neq 0$ for $i = 1(1)n-1$.

Remark (XV): If we write the result of the kth step of the forward sweep of the explicitly shifted method as

$$Q_k^T A_k Q_k = \begin{pmatrix} \lambda_k & b_k^T \\ 0 & A_{k+1} \end{pmatrix},$$

then if A_k is unreduced, A_{k+1} will also be unreduced. This result follows because a zero $(i,i-1)$ element of A_{k+1} implies that either the corresponding $(i,i-1)$ element of A_k is zero or the corresponding $(i+1,i)$ element of A_k is zero. This can be seen from the following example of a 3×3 matrix. In the forward sweep of the explicitly shifted method, one step is described by



Let α_{ij}^k represent the (i,j)-element of the kth matrix in this example. Suppose the first matrix is unreduced, so neither rotation is trivial, and suppose $\alpha_{32}^5 = 0$. Then we have either

$$\alpha_{32}^5 = 0 \Rightarrow \alpha_{32}^4 = 0 \Rightarrow \alpha_{22}^3 = 0 \Rightarrow \alpha_{21}^2 = 0 \Rightarrow \alpha_{21}^1 = 0 ,$$

or

$$\alpha_{32}^5 = 0 \Rightarrow \alpha_{32}^1 = 0 \quad \text{since rotation } 1 \text{ will not exist.}$$

But in both cases we have a contradiction, so $\alpha_{32}^5 \neq 0$. More formally, after some trivial calculations we get: First, from rotation 1

$$\alpha_{32}^1 c_1 + \alpha_{33}^1 s_1 = 0 , \quad \frac{c_1^2}{s_1^2} = \frac{(\alpha_{33}^1)^2}{(\alpha_{32}^1)^2} , \quad \text{so if } \rho_1^2 = (\alpha_{33}^1)^2 + (\alpha_{33}^1)^2 , \quad s_1 = -\frac{\alpha_{32}^1}{\rho_1}$$

Similarly from rotation 2

$$\rho_2^2 = (\alpha_{21}^2)^2 + (\alpha_{22}^2)^2 = (\alpha_{21}^1)^2 + (\alpha_{22}^2)^2$$

Now we can calculate α_{32}^5 as follows:

$$\alpha_{32}^5 = \alpha_{32}^4$$

$$= -s_1 \alpha_{22}^3$$

$$= \frac{\alpha_{32}^1}{\rho_1} \rho_2$$

$$= \frac{\alpha_{32}^1}{\rho_1} [(\alpha_{21}^1)^2 + (\alpha_{22}^2)^2]^{1/2} \qquad (52\beta)$$

were ρ_1, ρ_2 are well known numbers associated with rotations 1 and 2 respectively (see definition (IX)), and

$$\alpha_{22}^2 = \frac{1}{\rho_1} \det \begin{pmatrix} \alpha_{22}^1 & \alpha_{23}^1 \\ \alpha_{32}^1 & \alpha_{33}^1 \end{pmatrix} \qquad (53)$$

From (52β) we can see that if $\alpha_{32}^5 = 0$ then either $\alpha_{32}^1 = 0$ or both $\alpha_{21}^1 = 0$, $\alpha_{22}^2 = 0$.

So if A_k is unreduced then $A_k - \lambda_k I$ is unreduced which implies that $A_{k+1} - \lambda_k I$ and so A_{k+1} are unreduced too. But we know that A is unreduced because of the controllability of the system $\dot{x} = Ax + bu$ and theorem (I) therefore $A_1, A_2, \ldots, A_{n-1}$ are all unreduced.

Now since each matrix $A_k - \lambda_k I$ is unreduced its (2,1)-element is non zero, so the rotation P_{n-k} which is made to eliminate this element is non trivial, thus $s_k \neq 0$. But s_k can be relatively small if some of the subdiagonal

elements of $A_k-\lambda_k I$ are small (this can be seen from (52β), if α^1_{32} is small in comparison with α^1_{33} the α^5_{32} may be small and so will be the corresponding s used to eliminate α^5_{32} if α^5_{32} is not small, since $s = \dfrac{\alpha^5_{32}}{\sqrt{(\alpha^5_{32})^2 + (\alpha^5_{33})^2}}$.

Now from (51) we have

$$s_k \bar{a}_k^T + a_{k+1}^T = c_k a'^T_k \quad \langle = \rangle$$

$$s_k(0, \bar{a}_k^T) + (0, a_{k+1}^T) = c_k(0, a'^T_k) \quad \langle = \rangle$$

$$s_k(0, \bar{a}_k^T)Q_k^T + (0, a_{k+1}^T)Q_k^T = c_k(0, a'^T_k)Q_k^T \tag{54}$$

Let $\tilde{a}_k^T$ be the first row of A_k, k = 1(1)n. Then from (52α) we know that

$$(0, \bar{a}_k^T)Q_k^T = \tilde{a}_k^T - \lambda_k e_1^T \tag{54α}$$

We also know that from the second row of (47)

$$(0, a_{k+1}^T) = (0, \tilde{a}_{k+1}^T) - \lambda_k e_2^T \tag{54β}$$

Now (54) in view of (54α) and (54β) gives us

$$s_k(\tilde{a}_k^T - \lambda_k e_1^T) + (0, \tilde{a}_{k+1}^T)Q_k^T - \lambda_k e_2^T Q_k^T = c_k(0, a'^T_k)Q_k^T \quad \langle = \rangle$$

$$s_k \tilde{a}_k^T + (0, \tilde{a}_{k+1}^T)Q_k^T = c_k(0, a'^T_k)Q_k^T + s_k\lambda_k e_1^T + \lambda_k e_2^T Q_k^T \tag{54γ}$$

Let

$$c_k(0, a'^T_k)Q_k^T + s_k\lambda_k e_1^T + \lambda_k e_2^T Q_k^T = d_k^T, \tag{54δ}$$

which we see is wholly available from the forward sweep then (54γ) in view of (54δ) gives us

$$s_k\,\tilde{a}_k^T + (0,\ \tilde{a}_{k+1}^T)Q_k^T = d_k^T \qquad k = n\text{-}1(-1)1 \qquad (54\varepsilon)$$

Now let $\hat{a}_k^T = (\underbrace{0,\ldots,0}_{k-1},\tilde{a}_k^T)$, $\hat{d}_k^T = (\underbrace{0,\ldots,0}_{k-1},d_k^T)$, $\hat{Q}_k = \begin{bmatrix} I_{k-1} & 0 \\ 0 & Q_k \end{bmatrix}$

for $k = 1(1)n$, so $\hat{a}_k^T$, $\hat{d}_k^T$ are n-dimensional row vectors and $\hat{Q}_k$ is an $n\times n$ orthogonal matrix, then instead of (54ε) we have the following:

$$s_k\hat{a}_k^T + \hat{a}_{k+1}^T\,\hat{Q}_k^T = \hat{d}_k^T \quad <=>$$

$$s_k\hat{a}_k^T\hat{Q}_k + \hat{a}_{k+1}^T = \hat{d}_k^T\,\hat{Q}_k \quad , \quad k = n\text{-}1(-1)1 \qquad (54\zeta)$$

Now from (54ζ) we can observe the following:

For $k = n\text{-}1$ we have $(s_{n-1}\ ,\ 1)\begin{pmatrix} \hat{a}_{n-1}^T\,\hat{Q}_{n-1} \\ \hat{a}_n^T \end{pmatrix} = \hat{d}_{n-1}^T\,\hat{Q}_{n-1}$

For $k = n\text{-}2$ we have $(s_{n-2}\ ,\ 1)\begin{pmatrix} \hat{a}_{n-2}^T\,\hat{Q}_{n-2}\,\hat{Q}_{n-1} \\ \hat{a}_{n-1}^T\,\hat{Q}_{n-1} \end{pmatrix} = \hat{d}_{n-2}^T\,\hat{Q}_{n-2}\,\hat{Q}_{n-1}$

So finally we get the system:

$$\begin{pmatrix} s_1 & 1 & & & & \\ & s_2 & 1 & & & \\ & & \ddots & \ddots & & \\ & & & s_{n-2} & 1 & \\ & & & & s_{n-1} & 1 \\ & & & & & s_n \end{pmatrix} \begin{pmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_{n-2}^T \\ x_{n-1}^T \\ x_n^T \end{pmatrix} = \begin{pmatrix} g_1^T \\ g_2^T \\ \vdots \\ g_{n-2}^T \\ g_{n-1}^T \\ g_n^T \end{pmatrix} \qquad (54\eta)$$

$$\left.\begin{array}{l} \text{where} \quad x_i^T = \hat{a}_i^T \hat{Q}_i \hat{Q}_{i+1} \cdots \hat{Q}_{n-1} \hat{Q}_n \\ \text{and} \quad g_i^T = \hat{d}_i^T \hat{Q}_i \hat{Q}_{i+1} \cdots \hat{Q}_{n-1} \hat{Q}_n \end{array}\right\} , \; i = 1(1)n \text{ with } Q_n = I , \qquad (54\theta)$$

and since $\tilde{a}_n^T = \lambda_n$ (see (46)), the first (and only) row of A_n, we have taken $s_n = 1$, $g_n^T = (0,\ldots,0,\lambda_n)$, giving $x_n^T = g_n^T$. Note that $\hat{a}_1^T$ is the desired first row, and knowing x_1^T we know $\hat{a}_1^T = x_1^T \hat{Q}_n^T \hat{Q}_{n-1}^T \cdots \hat{Q}_1^T$. In the system of equations (54η) the unknowns are effectively the rows $\tilde{a}_k^T$, $k = 1(1)n-1$. So system (54η) can be solved from $\tilde{a}_n^T$ to $\tilde{a}_1^T \equiv \hat{a}_1^T$. It should be pointed out that this effective solution of equations appears to be as late in the computation as possible, and probably reffects the ill-condition of the problem. Thus there is some reason to believe that this algorithm will be numerically stable, since there is no other solution of equations, and all other computations are nice stable rotations. The ratio of the largest to smallest singular values of the matrix in (54η) may well give a measure of the sensitivity of the eigenvalue allocation problem.

(b) Implicitly shifted method:

Here we will describe a method in which every step of the forward sweep is achieved without having to subtract the eigenvalue from the diagonal and then restore it. We first describe and prove a theorem which will help us develop the implicitly shifted method.

Theorem (V): Let A, Q and H be real $n \times n$ matrices with $Q^TQ = I$, and H upper Hessenberg with elements $\eta_{i+1,i} > 0$ for $i = m+1(1)n-1$ where $1 \le m \le n-2$.

Write $Q = (\underbrace{Q_1}_{n\times m}, \underbrace{Q_2}_{n\times n-m})$, $H = \left[\begin{array}{c|c} H_1 & H_{12} \\ \hline \underbrace{\eta_{m+1,m} e_1 e_m^T}_{m} & \underbrace{H_2}_{n-m} \end{array}\right] \begin{matrix} \} m \\ \} n-m \end{matrix}$

where e_1 and e_m have $n-m$ and m elements respectively.

If $AQ = QH$ then Q_2, H_2 and $R(Q_1)$ are uniquely determined by A and the last column of Q_2. If $\eta_{m+1,m} = 0$ then $R(Q_1)$ is an eigensubspace of A. If $m = 1$ then apart from the sign of the first column of Q, Q and H are uniquely determined.

Proof: Let $Q = (q_1, q_2, \ldots, q_n)$ be partitioned into columns. Suppose that we already know q_n. Then we have

$$HQ^T = Q^T A \tag{55}$$

Suppose we have already calculated the last $n-k$ columns of Q, that is, we have calculated the q_i for $i = k+1(1)n$. Also suppose that we have already calculated the last $n-k-1$ rows of H. Now we will calculate the $(k+1)$th row of H and the kth column of Q. From (55) and the fact that H is upper Hessneberg, we obtain, with $H = (\eta_{ij})$

$$\eta_{k+1,k} q_k^T + \eta_{k+1,k+1} q_{k+1}^T + \ldots + \eta_{k+1,n} q_n^T = q_{k+1}^T A \ , \ k = n-1(-1)1 \ . \tag{56}$$

Since Q is orthogonal, we multiply (56) by q_i from the right to obtain

$$\eta_{k+1,i} = q_{k+1}^T A q_i \quad , \quad i = n(-1)k+1 \ . \tag{57}$$

Now we will calculate q_k and $\eta_{k+1,k}$. From (56), we obtain

$$q_k^T = \eta_{k+1,k}^{-1} \left(q_{k+1}^T A - \sum_{i=k+1}^{n} \eta_{k+1,i}\, q_i^T\right) , \quad k = n-1(-1)m+1 \tag{58}$$

Here (58) and the requirements $||q_k||_2 = 1$ and $\eta_{k+1,k} > 0$ uniquely determine q_k and $\eta_{k+1,k}$. This way we obtain all but the first m columns of Q and the first $m+1$ rows of H. The rest of H_2 is given by $H_2 = Q_2^T A Q_2$, and $R(Q_1)$ is that space orthogonal to Q_2. Now

$$AQ = QH \tag{59}$$

so if $\eta_{m+1,m} = 0$ we have $AQ_1 = Q_1 H_1$, $Q_1^T Q_1 = I$, and $R(Q_1)$ is an eigensubspace of A. If $m = 1$ then $Q_1 = q_1$ is uniquely determined but for its sign, so $H = Q^T A Q$ is uniquely determined to this extent. If $m=1$ and $\eta_{2,1} = 0$ then

$$Aq_1 = q_1 \eta_{1,1} , \tag{59α}$$

so q_1 is an eigenvector of A corresponding to the eigenvalue $\lambda = \eta_{11}$.

Remark (XVI): The requirement that $\eta_{k+1,k}$ be positive in the statement of the theorem was necessary only to tie down uniqueness. Actually $\eta_{k+1,k}$, $k = m+1(1)n-1$ and q_k, $k = m+1(1)n$ are determined up to a constant factor of absolute value unity simply by the requirement that $\eta_{k+1,k} \neq 0$, $k = m+1(1)n-1$. It is this essential uniqueness of H and Q that we shall use.

To see how theorem (V) can be applied to improve the explicitly shifted method, suppose that one step of the forward sweep with shift λ has been applied to the unreduced upper Hessenberg matrix A to yield an upper

Hessenberg matrix H such that $H = \begin{pmatrix} \lambda & b^T \\ 0 & \tilde{A} \end{pmatrix}$ with $\tilde{A}$ an unreduced upper Hessenberg matrix. Then,

$$H = Q^T A Q \quad , \quad \text{and } Q \text{ is lower Hessenberg.}$$

The following algorithm is an alternate way of computing H.

1) Find an orthogonal matrix P such that

$$Pe_n = Qe_n \tag{60}$$

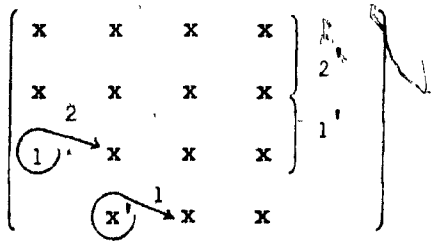
that is, P and Q have the same last column. To do this choose the rotation P^T such that $P^T Q e_n = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ x \\ x \end{pmatrix} = e_n$, more will be said on this later.

2) Form $P^T A P$, where, for example, if A was a 4×4 matrix then $P^T A P$ would have the form $\begin{pmatrix} x & x & x & x \\ x & x & x & x \\ & x & x & x \\ & x' & x & x \end{pmatrix}$

3) Find orthogonal matrices U_i, $i = 1(1)n-2$ to reduce $P^T A P$ to upper Hessenberg form H', that is

$$U_{n-2}^T \cdots U_1^T P^T A P U_1 \cdots U_{n-2} = H' \tag{61}$$

where continuing the same example H' would have the form



If we set $Q' = PU_1 \ldots U_{n-2}$, then from (61) we obtain

$$Q^T{}'AQ' = H' ,$$

and we see from the form of the U_i that $U_i e_n = e_n$, so $Q'e_n = Pe_n = Qe_n$ and from Theorem (V) we have effectively $Q' = Q$, $H' = H$. As a result we set the (unknown) (2,1) element of H' to zero, and (1,1) element to λ.

To determine P we should find the last row of Q^T. We know that $(A-\lambda I)Q = R$, where R is upper triangular matrix with elements ρ_{ij}, $i = 1(1)n$, $j = 1(1)n$. Now

$$(A-\lambda I)Q = R \quad \Rightarrow$$

$$A - \lambda I = RQ^T \quad \Rightarrow$$

$$e_n^T(A - \lambda I) = \rho_{nn} q_n^T , \qquad (62)$$

where $Q = (q_1, q_2, \ldots, q_n)$. So q_n^T is a multiple of the last row a^T of $A - \lambda I$. Since A is upper Hessenberg we know that

$$a^T = (0, \ldots, 0, \alpha_{n,n-1}, \alpha_{nn}-\lambda)$$

Then we choose P such that

$$a^T P = \pm ||a||_2 \, e_n^T \tag{62α}$$

(a P like this can always be found, see [12], page 232, theorem 3.4).
So from (62α) we get

$$a^T = \pm ||a||_2 \, e_n^T P^T \quad \Rightarrow$$

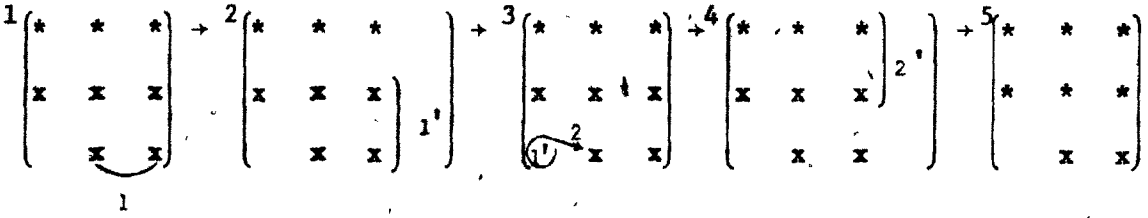
$$e_n^T P^T = \pm \frac{a^T}{||a||_2} \tag{63}$$

So using the P of (63), the last row of which is a multiple of the last row of $A - \lambda I$, we accomplish our aim, to find a matrix P such that $Pe_n = Qe_n$.

Now we will give an example with a 3×3 matrix to demonstrate how the forward sweep of the implicitly shifted method works.

Let $A = \begin{pmatrix} * & * & * \\ x & x & x \\ & x & x \end{pmatrix}$ then we find a rotation matrix P such that if

we apply it to the matrix $A - \lambda I$ from the right it will eliminate the (3,2)-element of $A - \lambda I$ into the (3,3)-element of the same matrix. Of course, P will not do the same to the matrix A unless $\lambda = 0$. So we will have the following:



Now we set the (1,1)-element of the 5th matrix equal to λ, so the (2,1)-element of the same matrix becomes zero in view of the theorem (V). Finally, we have

$$Q^T A Q = \begin{bmatrix} \lambda & b^T \\ 0 & \tilde{A} \end{bmatrix} .$$

That is what exactly we would have, using the explicitly shifted method, but without subtracting the shift from the diagonal of A and then restore it back.

Now we will describe the kth step of the backward sweep of the implicitly shifted method, which as we will see is almost the same as the backward sweep of the explicitly shifted method. At the kth step, we have

$$Q_k^T A_k Q_k = \begin{bmatrix} \lambda_k & b_k^T \\ 0 & A_{k+1} \end{bmatrix} . \qquad (63\alpha)$$

As in the explicitly shifted method we know the first row a_{k+1}^T of A_{k+1} and we want to find the first row a_k^T of A_k. Let $(\bar{\lambda}_k , \bar{a}_k^T)$ be the first row of $A_k Q_k$ before the effect of applying the orthogonal matrix Q_k^T on the left, or, as in the explicitly shifted method, before the effect of the rotation matrix P_{n-k}^T as this is the only rotation matrix of the matrix Q_k^T that will affect the first row of $A_k Q_k$, then we have

$$\begin{bmatrix} c_k & s_k \\ -s_k & c_k \end{bmatrix} \begin{bmatrix} \bar{\lambda}_k & \bar{a}_k^T \\ \gamma & a'^T_k \end{bmatrix} = \begin{bmatrix} \lambda_k & b_k^T \\ 0 & a_{k+1}^T \end{bmatrix} \qquad (63\beta)$$

from (63β) and since (γ , a'^{T}_{k}) is well known, we obtain

$$c_k\gamma - s_k\bar{\lambda}_k = 0 \quad <=>$$

$$\bar{\lambda}_k = \frac{c_k\gamma}{s_k} \tag{64}$$

and

$$c_k a'^{T}_{k} - s_k \bar{a}^{T}_{k} = a^{T}_{k+1} \quad <=>$$

$$\bar{a}^{T}_{k} = \frac{c_k a'^{T}_{k} - a^{T}_{k+1}}{s_k} \tag{65}$$

So now we know the first row $(\bar{\lambda}_k , \bar{a}^{T}_{k})$ of $A_k Q_k$, that is

$$a^{T}_{k} Q_k = (\bar{\lambda}_k , \bar{a}^{T}_{k}) \quad <=>$$

$$a^{T}_{k} = (\bar{\lambda}_k , \bar{a}^{T}_{k}) Q^{T}_{k} . \tag{66}$$

Thus from knowing the first row of A_{k+1} we can find the first row of A_k. But we know the first (only) row of $A_n = \lambda_n$ in (63α) with $k = n - 1$ and so we can go backwards to finally produce the first row of $A \equiv A_1$, our desired result.

We now make one remark and prove a theorem based on the remark. This theorem will help us develop an implicitly shifted method for the case of complex eigenvalues.

Remark (XVII): It may have been noticed that the explicitly and the implicitly shifted methods which have been described here are quite similar with the QR explicitly and implicitly shifted algorithms, respectively. The basic difference is that in our methods instead of trying to get an upper triangular matrix R by multiplying a matrix $A - \mu I$, $\mu \in \mathbb{R}$ from the left by an orthogonal matrix Q^T, we multiply $A - \mu I$ from the right, where μ is a well known shift. So the kth step of an algorithm based on our explicitly shifted method, to calculate the eigenvalues of A, would be as follows:

(1) Form $A_k - \mu_k I$

(2) Factor the above matrix in the form

$$(A_k - \mu_k I)Q_k = R_k \quad <=> \tag{67}$$

$$A_k - \mu_k I = R_k Q_k^T \tag{67a}$$

(3) Finally compute A_{k+1} as

$$A_{k+1} = Q_k^T R_k + \mu_k I \tag{68}$$

From (67) and (68) we obtain:

$$A_{k+1} = Q_k^T A_k Q_k . \tag{68a}$$

Theorem (VI): Let $\tilde{Q}_m = Q_1 Q_2 \ldots Q_m$ and $\tilde{R}_m = R_1 R_2 \ldots R_m$, where Q_i, R_i with $i = 1(1)m$ are the orthogonal and upper triangular matrices of the ith step of the above mentioned algorithm applied for m steps, then

$$\tilde{R}_m \tilde{Q}_m^T = (A - \mu_1 I)(A - \mu_2 I) \ldots (A - \mu_m I) . \tag{69}$$

Proof: Let $A \equiv A_1$. Then obviously, we obtain

$$A_{k+1} = \tilde{Q}_k^T A \tilde{Q}_k \ . \tag{70}$$

we will prove (69) by induction. For $m = 1$ (69) gives

$$\tilde{R}_1 \tilde{Q}_1^T = A_1 - \mu_1 I \quad =$$

$$R_1 Q_1^T = A_1 - \mu_1 I$$

which is true if in (67α) we set $k = 1$. Suppose (69) is true for $m = k-1$, that is, suppose that the following relation is true.

$$\tilde{R}_{k-1} \tilde{Q}_{k-1}^T = (A - \mu_1 I)\ (A - \mu_2 I) \ . \ . \ . \ (A - \mu_{k-1} I) \tag{71}$$

Now we will prove that (69) is true for $m = k$, that is, we will prove that the relation

$$\tilde{R}_k \tilde{Q}_k^T = (A - \mu_1 I)\ (A - \mu_2 I) \ . \ . \ . \ (A - \mu_{k-1} I)\ (A - \mu_k I) \tag{72}$$

is true. (68) because of (70) gives us

$$Q_k^T R_k = \tilde{Q}_k^T A \tilde{Q}_k - \mu_k I \quad <=>$$

$$Q_k^T R_k = \tilde{Q}_k^T (A - \mu_k I) \tilde{Q}_k \quad <=>$$

$$R_k = \tilde{Q}_{k-1}^T (A - \mu_k I) \tilde{Q}_k \quad =>$$

$$\tilde{R}_k = \tilde{R}_{k-1} \tilde{Q}_{k-1}^T (A - \mu_k I) \tilde{Q}_k \quad <=>$$

$$\tilde{R}_k \tilde{Q}_k^T = \tilde{R}_{k-1} \tilde{Q}_{k-1}^T (A - \mu_k I) \tag{72α}$$

(72α) becuase of (71) gives us

$$\tilde{R}_k \tilde{Q}_k^T = (A - \mu_1 I)\ (A - \mu_2 I)\ \ldots\ (A - \mu_k I)\ (A - \mu_k I)$$

which is (72).

(2) The case of complex eigenvalues:

Here we will describe how to accomplish two steps of the forward sweep of the implicitly shifted method in one pass, and we will show how we can use this double step to assign a pair of complex conjugate eigenvalues to a matrix A. Let λ_1, λ_2 be the two eigenvalues and let $A_1 \equiv A$, then from (68α) for $k = 1$ and $k = 2$ we have respectively:

$$\begin{aligned} A_2 &= Q_1^T A_1 Q_1 \quad \text{and} \\ A_3 &= Q_2^T A_2 Q_2 \quad \Rightarrow \\ A_3 &= Q_2^T Q_1^T A_1 Q_1 Q_2 \end{aligned} \tag{73}$$

where A_1, A_2 and A_3 are upper Hessenberg. In order to accomplish these two steps implicitly, we first find an orthogonal matrix P_1 such that $P_1 e_n = Q_1 Q_2 e_n$. To do this we should find the last row of $Q_2^T Q_1^T$. From theorem (VI) we have

$$R_1 R_2 Q_2^T Q_1^T = (A - \lambda_1 I)\ (A - \lambda_2 I)\ , \tag{74}$$

where the upper triangular matrices R_1, R_2 have elements $\rho_{ij}^{(1)}$, $\rho_{ij}^{(2)}$, respectively. From (74) we can see that if q_n^T is the last row of $Q_2^T Q_1^T$ then

$$(\rho_{nn}^{(2)}\ \rho_{nn}^{(1)}) q_n^T = e_n^T (A - \lambda_1 I)\ (A - \lambda_2 I)\ . \tag{75}$$

So q_n^T is a multiple of the last row a^T of the matrix $(A - \lambda_1 I)(A - \lambda_2 I)$. But A is upper Hessenberg so a^T has only three non-zero elements, $a^T = (0,\ldots,0,\alpha_{0,n-2},\alpha_{0,n-1},\alpha_{0n})$ say. Then after some trivial calculations, we find that

$$\left.\begin{aligned} \alpha_{0,n-2} &= \alpha_{n-1,n-2}\,\alpha_{n,n-1} \\ \alpha_{0,n-1} &= \alpha_{n,n-1}[\alpha_{n-1,n-1} + \alpha_{nn} - (\lambda_1 + \lambda_2)] \\ \alpha_{0,n} &= \alpha_{nn}^2 + \alpha_{n,n-1}\,\alpha_{n-1,n} - \alpha_{nn}(\lambda_1 + \lambda_2) + \lambda_1\lambda_2 \end{aligned}\right\} \qquad (76)$$

Now choose P_1 such that

$$\begin{aligned} a^T P_1 &= \pm ||a||_2\, e_n^T \quad \Leftrightarrow \\ a^T &= \pm ||a||_2\, e_n^T P_1^T \quad \Leftrightarrow \\ e_n^T P_1^T &= \pm \frac{a^T}{||a||_2} \,. \end{aligned} \qquad (77)$$

Thus

$$P_1 e_n = \pm \frac{a}{||a||_2} = \pm \frac{\rho_{nn}^{(2)}\,\rho_{nn}^{(1)}}{||a||_2}\, q_n = \pm Q_1 Q_2 e_n$$

and we have accomplished our aim.

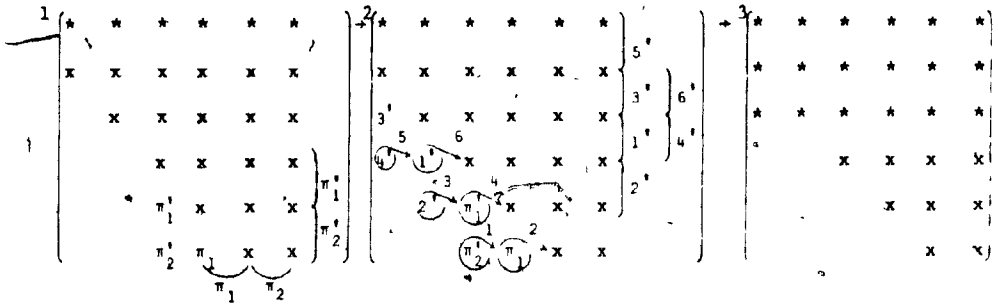
From (76) we can see that if $\lambda_1 = \lambda$ and $\lambda_2 = \bar{\lambda}$, where $\lambda \in \mathbb{C}$ and $\bar{\lambda}$ is the complex conjugate of λ, P_1 is real, and can be found by using real arithmetic, since $\lambda + \bar{\lambda}$ and $\lambda\bar{\lambda}$ are reals even when $\lambda \in \mathbb{C}$. Now, after having found P_1, the next step is to find orthogonal matrices U_i, $i = 1(1)n$ each with $U_i e_n = e_n$, such that the matrix A' is upper Hessenberg in

$$A' = Q^T A_1 Q \quad , \text{ where } \quad Q = P_1 U_1 \cdots U_n \quad .$$

Then we will have $Qe_n = P_1 e_n = \pm Q_1 Q_2 e_n$, and since A_3 has nonzero $(i+1, i)$ elements, $i = 3(1)n-1$, the last $n-2$ columns of Q and $Q_1 Q_2$ are essentially the same, as are the last $(n-2)\times(n-2)$ diagonal blocks of A' and A_3 . This follows from Theorem (V). We will give an example with a 6×6 matrix to demonstrate how the forward sweep of the double step method works. Let

$$A = \begin{pmatrix} * & * & * & * & * & * \\ x & x & x & x & x & x \\ & x & x & x & x & x \\ & & x & x & x & x \\ & & & x & x & x \\ & & & & x & x \end{pmatrix}$$

then we find an orthogonal matrix P such that if we apply it to the matrix $(A - \lambda_1 I)(A - \lambda_2 I)$ from the right it will first eliminate the $(6,4)$-element of $(A - \lambda_1 I)(A - \lambda_2 I)$ into the $(6,5)$-element of the same matrix, and then the resulting $(6,5)$-element of $(A - \lambda_1 I)(A - \lambda_2 I)$ into the $(6,6)$-element of the same matrix. So when we first form the matrix $P^T A P$, and then we reduce it to upper Hessenberg form, we will have the following:



Note: The element 3' cannot be eliminated because after the multiplication of the rotation matrix 6' , this element becomes unknown, as we can see from matrix 3 . Since from Theorem (V) the first two columns of Q span the eigenspace of A corresponding to λ_1 and λ_2 , we see that the elements (3,1) and (3,2) of matrix 3 must be equal to zero, and the elements (1,1) , (1,2) , (2,1) , (2,2) of the same matrix will be forced to be equal to γ , δ , ε , ζ , respectively, where $\gamma + \zeta = 2\alpha$ and $\det\begin{pmatrix}\gamma & \delta \\ \varepsilon & \zeta\end{pmatrix} = \alpha^2 + \beta^2$, $\lambda_1 = \alpha + j\beta$ and $\lambda_2 = \alpha - j\beta$ with $j^2 = -1$. In other words, γ , δ , ε , ζ are such that λ_1 and λ_2 are the two eigenvalues of the matrix $\begin{pmatrix}\gamma & \delta \\ \varepsilon & \zeta\end{pmatrix}$. Finally, we have after this first double step of the forward sweep

$$Q^T A Q = \begin{pmatrix} \gamma & \delta & b_1^T \\ \varepsilon & \zeta & b_2^T \\ 0 & 0 & \tilde{A} \end{pmatrix} ,$$

where b_1 , b_2 and the first row of $\tilde{A}$ are as yet unknown.

Let us now examine the backward sweep when we know the first row of A_{k+2} and we want to find the first row of A_k .

Let $$Q_k^T A_k Q_k = \begin{pmatrix} \gamma & \delta & b_k^T \\ \varepsilon & \zeta & b_{k+1}^T \\ 0 & 0 & A_{k+2} \end{pmatrix}$$

Let a_{k+2}^T and a_k^T be the first rows of A_{k+2} and A_k, respectively. From the above example, we can see that the first row of A was affected only by rotation 5', coming from the matrix Q^T. But in order to calculate the first row of A we should take into consideration rotation 6' also. So the first three rows of the last two rotation matrices from the left were:

$$\begin{pmatrix} 1 & & \\ & c_k & s_k \\ & -s_k & c_k \end{pmatrix} \begin{pmatrix} c_{k-1} & s_{k-1} & \\ -s_{k-1} & c_{k-1} & \\ & & 1 \end{pmatrix} = \begin{pmatrix} c_{k-1} & s_{k-1} & 0 \\ -c_k s_{k-1} & c_k c_{k-1} & s_k \\ s_k s_{k-1} & -s_k c_{k-1} & c_k \end{pmatrix}$$

Let $(\tilde{\alpha}, \tilde{\beta}, \tilde{b}^T)$ be the first row of $A_k Q_k$. Then

$$\begin{pmatrix} c_{k-1} & s_{k-1} & 0 \\ -c_k s_{k-1} & c_k c_{k-1} & s_k \\ s_k s_{k-1} & -s_k c_{k-1} & c_k \end{pmatrix} \begin{pmatrix} \tilde{\alpha} & \tilde{\beta} & \tilde{b}^T \\ \mu & \nu & f^T \\ \xi & \pi & g^T \end{pmatrix} = \begin{pmatrix} \gamma & \delta & b_k^T \\ \varepsilon & \zeta & b_{k+1}^T \\ 0 & 0 & a_{k+2}^T \end{pmatrix} \qquad (78)$$

Then, from (78) and since the rows (μ, ν, f^T) and (ζ, π, g^T) are computed in the forward sweep, we obtain:

$$(s_k s_{k-1}, -s_k c_{k-1}, c_k) \begin{pmatrix} \tilde{\alpha} & \tilde{\beta} & \tilde{b}^T \\ \mu & \nu & f^T \\ \xi & \pi & g^T \end{pmatrix} = (0, 0, a_{k+2}^T) \quad <=>$$

$$\begin{pmatrix} \tilde{\alpha} \\ \tilde{\beta} \\ \tilde{b}^T \end{pmatrix} = \begin{pmatrix} \dfrac{s_k c_{k-1} \mu - c_k \xi}{s_k s_{k-1}} \\ \dfrac{s_k c_{k-1} \nu - c_k \pi}{s_k s_{k-1}} \\ \dfrac{s_k c_{k-1} f^T - c_n g^T + a_{k+2}^T}{s_k s_{k-1}} \end{pmatrix} \tag{79}$$

So the first row of the matrix $A_k Q_k$ is $(\tilde{\alpha}, \tilde{\beta}, \tilde{b}^T)$ that is

$$e_1^T A_k Q_k = (\tilde{\alpha}, \tilde{\beta}, \tilde{b}^T) \quad =>$$

$$e_1^T A_k = (\tilde{\alpha}, \tilde{\beta}, \tilde{b}^T) Q_k^T$$

Thus

$$a_K^T = (\tilde{\alpha}, \tilde{\beta}, \tilde{b}^T) Q_k^T .$$

So knowing the first row of A_{k+2} we can find the first row of A_k, when we are using such a double step.

Now if we want to assign a set of eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$, where the complex conjugate pairs appear together, to an unreduced upper Hessenberg matrix A, we will use a combination of double steps or implicitly shifted

single steps, according to what type of eigenvalue we have to assign at each step; a complex conjugate pair or a single real eigenvalue. So knowing the first row of A_n or A_{n-1}, according to whether the last eigenvalue to be assigned is a real one or a complex conjugate pair respectively, we can go backwards to finally produce the first row of $A_1 \equiv A$, the desired result.

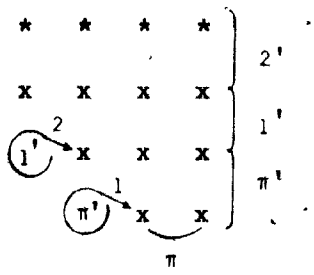
In the following and until the end of this section we shall compare the two previously described methods of assigning eigenvalues, that is, the double and the single step.

Note that four multiplications are involved when any rotation R, is applied to any vector x.

Single step

(a) Forward sweep:

Suppose we want to assign one eigenvalue to a $k \times k$ unreduced upper Hessenberg matrix with $k = 4$, then we have the following



from which we see that rotations π and π' involve 4×4 and 4×3 multiplications respectively. Similarly rotations 1, 1', 2 and 2' involve 4×4, 4×4, 4×3 and 4×4 multiplications. Thus the number of multiplications which are involved in one forward step of a 4×4 matrix can be found as follows:

$$
\begin{array}{l}
4\times4 + 4\times3 \\
4\times4 + 4\times4 \\
\underline{4\times3 + 4\times4 \quad +} \\
(4\times4 + 4\sum_{i=3}^{4} i) + (4\times4 + 4\sum_{i=3}^{4} i) = 88
\end{array}
$$

So in general we have the following:

$$
\begin{array}{l}
4k \quad + 4\times3 \\
4k \quad + 4\times4 \\
4(k-1) + 4\times5 \\
\vdots \\
4\times4 \quad + 4k \\
\underline{4\times3 \quad + 4k \quad +} \\
(4k + 4\sum_{i=3}^{k} i) + (4k + 4\sum_{i=1}^{k} i) = 8k + 8\sum_{i=1}^{k} i - 24 \\
\qquad\qquad = 4k^2 + 12k - 24 \\
\qquad\qquad \simeq 4k^2 \qquad\qquad (80)
\end{array}
$$

Hence in the single step method one forward step involves multiplications of the order $4k^2$ when it is applied to a $k\times k$ matrix. So the order of the number of multiplications involved in the forward sweep is the following:

$$
\begin{aligned}
4\sum_{k=1}^{n} k^2 &= \frac{4}{3}\, n(n+\tfrac{1}{2})(n+1) \\
&\simeq \frac{4}{3}\, n^3 , \qquad\qquad (81)
\end{aligned}
$$

where n is the order of the matrix we deal with.

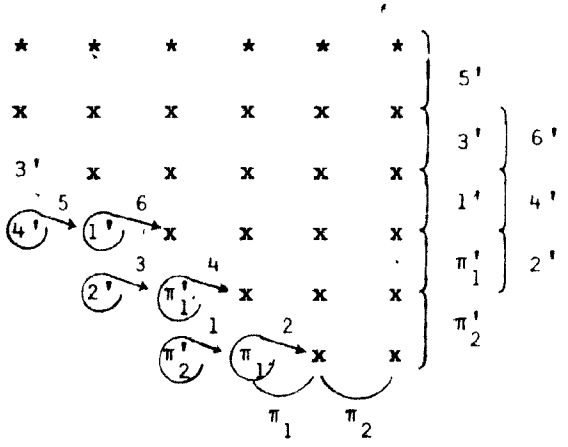
(b) Backward sweep:

Suppose we are at the kth step of the backward sweep, that is, we are to find the first row of the $k\times k$ submatrix knowing the first row of the $(k-1)\times(k-1)$ submatrix, then from (64), (65) and since at this step a'^{T}_{k} and a^{T}_{k+1} are $(k-1)$-vectors, we need k multiplications to calculate the first row of the $k\times k$ submatrix. So the order of the number of multiplications involved in the backward sweep is $\sum_{k=1}^{n} k \approx \frac{1}{2} n^2$.

Double step

(a) Forward sweep:

Suppose we want to assign two eigenvalues to a $k\times k$ unreduced upper Hessenberg matrix with $k = 6$, then we have the following:



from which we see that rotations π_1, π_2, π'_1, π'_2 involve 4×6, 4×6, 4×4 and 4×4 multiplications respectively. Similarly we can find the number of multiplications involved in rotations 1, 1', 2, 2', 3, 3', 4, 4', 5, 5', 6, 6'. Thus the number of multiplitations involved in one forward step of the double step method of a 6×6 matrix, can be found as follows:

$$
\begin{array}{l}
4\times6 + 4\times6 + 4\times4 + 4\times4 \\
4\times6 + 4\times6 + 4\times5 + 4\times5 \\
4\times5 + 4\times5 + 4\times6 + 4\times6 \\
\underline{4\times4 + 4\times4 + 4\times6 + 4\times6 \quad +} \\
4(4\times6 + 4\sum_{i=4}^{6} i) = 336
\end{array}
$$

So in general we have the following:

$$
\begin{array}{l}
4k + 4k + 4\times4 + 4\times4 \\
4k + 4k + 4\times5 \quad 4\times5 \\
4(k-1) + 4(k-1) + 4\times6 + 4\times6 \\
\quad\vdots \\
4\times5 + 4\times5 + 4k + 4k \\
\underline{4\times4 + 4\times4 + 4k + 4k \quad +} \\
4(4k + 4\sum_{i=4}^{k} i) = 16k + 16\sum_{i=1}^{k} i - 96 \\
\qquad = 8k^2 + 24k - 96 \\
\qquad \approx 8k^2 \qquad (82)
\end{array}
$$

Hence in the double step method one forward step involves multiplications of the order $8k^2$ when it is applied to a $k\times k$ matrix. So the order of the number of multiplications involved in the forward sweep, when n is even, is the following:

$$
\begin{aligned}
8\sum_{k=1}^{n/2} (2k)^2 &= 32\sum_{k=1}^{n/2} k^2 \\
&\approx \frac{32}{3}\left(\frac{n}{2}\right)^3 \\
&= \frac{4}{3} n^3 . \qquad (83)
\end{aligned}
$$

If n is odd then obviously we will have to use at least one single step. Note that the forward sweep of the double step method can be carried out using elementary reflectors [12] instead of rotations.

(b) Backward sweep:

Suppose we are at the kth step of the backward sweep, then from (79) since f, g and a_{k+2}^T are $(k-2)$-vectors we need multiplications of order $2k$ to calculate the first row of the $k \times k$ matrix. So the order of the number of multiplications involved in the backward sweep is $\frac{n^2}{2}$.

Hence from the above results we can observe that one double step involves almost as many multiplications as are involved in two single steps, thus the double step, with rotations, and the single step are equivalent in solving the eigenvalue allocation problem, in the sense that they involve the same order of number of multiplications.

CHAPTER 4

(I) Conclusion

In the second Chapter of this thesis we considered the most widely, so far, used algorithms in computing controllability and we compared them with a recently developed algorithm by C. Paige in [2]. The comparison was done by using appropriate counterexamples and it showed us that there are cases where the algorithms which were used so far fail, while the new algorithm gives the correct results. Finally in the same Chapter we presented an algorithm which using the notion of the distance of a system from the nearest uncontrollable one determines whether the system is controllable or not, the same algorithm with some changes (see note at the end of Chapter 2) can be used to calculate the distance of the given system from the nearest uncontrollable one, but it is pointed out why this algorithm is not efficient. Thus for determining the controllability of a system we have a reasonable algorithm, but for a controllable system we do not yet have an efficient algorithm to measure its distance from the nearest uncontrollable system, but we have the theoretical approach developed in section (III) of Chapter 2 of this thesis which may prove useful in developing an efficient algorithm. So the efficient computation of the distance of a system from the nearest uncontrollable one remains an open problem.

In Chapter 3 we have presented a very efficient, fast and we hope numerically stable algorithm to solve the eigenvalue allocation problem which can be used to determine the feedback such that a single input closed

loop system is stable. An algorithm for the same problem which was suggested by Luenberger in [3] appears to be unstable because it makes use of the coefficients of the characteristic polynomial of A which can be found by computing the eigenvalues of A, but we know that the problem of calculating the eigenvalues is in general ill-conditioned (if A is not symmetric), however Luenberger's algorithm can be extended to multi-input dynamic systems [10]. The method given here has also been designed with extension to multi-input systems in mind, but this has not yet been done.

APPENDIX

Computational results based on the implicitly shifted and double step algorithms. Subroutine EVA.

Based on the implicitly shifted and the double step algorithms, which were described in Chapter 3 we made a subroutine named EVA (Eigenvalue Allocation) in FORTRAN IV. This subroutine takes a matrix A without its first row and a set L of eigenvalues and evaluates the first row of A such that $\lambda(A) = L$. With this subroutine we checked some cases, where the reliability of the results was checked as follows: First the full matrix A was passed to the IMSL routine EQRH3F, in order to calculate its eigenvalues. Then the eigenvalues computed by the EQRH3F routine together with A (without its first row) were given to subroutine EVA and the first row of A was computed, and finally the full matrix A (with its computed first row) was given again to the IMSL routine EQRH3F, in order to compute its eigenvalues. The results were obtained using single precision arithmetic on the AMDAHL 470/V7 at McGill University. We also give an example in which we point out why using the above way to check EVA we may obtain results which differ significantly from those expected. This is not necessarily a fault of the algorithm (that is, it does not mean that the algorithm is necessarily numerically unstable). It may just be a result of the poor conditioning of the specific eigenproblem.

Example (I)

n = 5

The matrix A (with the original first row)

$$\begin{bmatrix} 5.279000 & 9.125000 & 4.433000 & 6.297000 & 5.687000 \\ 38.345000 & 39.492000 & 3.605000 & 5.987000 & 7.770000 \\ & -5.564000 & 6.396000 & 6.492000 & 5.889000 \\ & & 3.564000 & 9.539000 & 6.364000 \\ & & & -5.977000 & 4.796000 \end{bmatrix}$$

The eigenvalues of A using EQRH3F

46.726480

-3.995152

3.844520

9.463070 + j 3.235559

9.463070 - j 3.235559

The first row of A using EVA

5.278953 9.124928 4.433008 6.296913 5.687075

The eigenvalues of A , with the computed first row, using EQRH3F

46.726427

-3.995148

3.844508

9.463083 + j 3.235580

9.463083 - j 3.235580

Example (II)

n = 6

The matrix A with given first row

$$\begin{bmatrix} 9.452000 & -4.279000 & 5.126000 & 6.433000 & 3.297000 & 4.687000 \\ 6.474000 & -8.345000 & 79.490000 & 7.605000 & 0.987000 & 8.770000 \\ & 4.657000 & 5.564000 & 7.396000 & 7.492000 & 7.890000 \\ & & -0.998000 & 4.564000 & 9.540000 & 9.364000 \\ & & & -7.463000 & -7.977000 & 3.796000 \\ & & & & -9.897000 & 8.697000 \end{bmatrix}$$

The eigenvalues of A using EQRH3F

-21.014220

17.953000

-2.881598 + j 9.934898

-2.881598 - j 9.934898

9.028457

11.750950

The first row of A using EVA

9.452018 -4.278798 5.125249 6.429889 3.297468 4.688825

The eigenvalues of A , with the computed first row, using EQRH3F

-21.014265

17.953045

-2.881592 + j 9.934899

-2.881592 - j 9.934899

9.028496

11.750926

Example (III)

n = 9

The matrix A with given first row

$$\begin{bmatrix}
8.657000 & 5.254000 & 6.254000 & -5.565000 & 6.874000 & 0.657000 & 7.364000 & 4.655000 & -0.763000 \\
3.546000 & -6.356000 & 5.567000 & -6.534000 & 5.578000 & 5.546000 & 4.356000 & 0.768000 & 0.658000 \\
 & 6.865000 & 7.456000 & 6.457000 & 6.845000 & -0.176000 & 9.763000 & 5.765000 & -7.765000 \\
 & & 7.645000 & -6.656000 & -6.645000 & 5.657000 & 7.456000 & 5.765000 & -5.655000 \\
 & & & 7.565000 & 5.343000 & 0.678000 & 0.567000 & 6.678000 & 8.568000 \\
 & & & & -8.876000 & 5.684000 & 8.563000 & 7.456000 & 4.357000 \\
 & & & & & 7.536000 & -7.657000 & 7.456000 & -6.672000 \\
 & & & & & & -4.453000 & -6.366000 & -6.366000 \\
 & & & & & & & 5.464000 & -4.366000
\end{bmatrix}$$

The eigenvalues of A using EQRH3F

12.579090

9.194291 + j 2.895247

9.194291 - j 2.895247

-0.525684 + j 6.229814

-0.525684 - j 6.229814

-8.440416 + j 6.017367

-8.440416 - j 6.017367

-12.636600

-4.659868

The first row of A using EVA

8.656885 5.254084 6.254014 -5.565294 6.874562

0.657171 7.363661 4.657894 -0.763001

The eigenvalues of A, with the computed first row, using EQRH3F

12.579074

9.194260 + j 2.895261

9.194260 - j 2.895261

-0.525680 + j 6.229808

-0.525680 - j 6.229808

-8.440431 + j 6.017374

-8.440431 - j 6.017374

-12.636604

-4.659882

Example (IV)

n = 10.

The matrix A with given first row

$$\begin{pmatrix}
9.451000 & 8.321000 & 9.789000 & 5.789000 & -9.535000 & 4.654000 & 0.534000 & 8.654000 & 6.345000 & -0.453000 \\
7.355000 & 6.474000 & -8.345000 & 8.937000 & -3.034000 & 9.495000 & 7.605000 & 8.456000 & 0.987000 & 8.778000 \\
 & 8.657000 & 4.567000 & 4.436000 & 8.543000 & 6.345000 & -0.134000 & 9.453000 & 0.654000 & -9.342000 \\
 & & 5.323000 & 5.778000 & -4.786000 & -0.345000 & 5.675000 & 6.345000 & 6.354000 & -6.745000 \\
 & & & 4.453000 & 5.675000 & 9.563000 & 8.534000 & 5.756000 & 9.324000 & 4.354000 \\
 & & & & -5.977000 & 4.796000 & 5.534000 & 5.543000 & 8.312000 & 98.324000 \\
 & & & & & 3.134000 & 5.687000 & -0.234000 & 6.235000 & -9.522000 \\
 & & & & & & 5.314000 & -5.366000 & 5.456000 & -6.346000 \\
 & & & & & & & 4.363000 & -4.346000 & 9.346000 \\
 & & & & & & & & 0.235000 & 4.546000
\end{pmatrix}$$

The eigenvalues of A using EQRH3F

17.525430	-8.598625
0.776537 + j 10.545620	2.859068 + j 2.695472
0.776537 - j 10.545620	2.859068 - j 2.695472
8.885132 + j 5.104329	-1.585043
8.885132 - j 5.104329	4.878755

The first row of A using EVA

9.450891	8.320937	9.788930	5.789191	-9.535405
4.651105	0.547856	8.618833	6.374817	-0.460253

The eigenvalues of A , with the computed first row, using EQRH3F

17.525366	-8.598647
0.776526 + j 10.545624	2.859084 + j 2.695479
0.776526 - j 10.545624	2.859084 - j 2.695479
8.885115 + j 5.104358	-1.585052
8.885115 - j 5.104358	4.878776

Remark (XVIII): Using the previous way of checking subroutine EVA we may get some poor results due to some ill-conditioned eigenvalues.

Example (V):

Let the 10×10 matrix A without its first row be

$$\begin{pmatrix}
2 & 5 & 3 & 6 & 4 & 7 & 4 & 7 & 8 & 9 \\
 & 2 & 5 & 3 & 6 & 8 & 8 & 3 & 5 & 7 \\
 & & 8 & 3 & 4 & 5 & 7 & 8 & 4 & 6 \\
 & & & 9 & 6 & 4 & 7 & 8 & 3 & 2 \\
 & & & & 9 & 4 & 7 & 6 & 4 & -7 \\
 & & & & & 7 & 6 & 7 & 8 & 6 \\
 & & & & & & 5 & 5 & 4 & 3 \\
 & & & & & & & 8 & 6 & 7 \\
 & & & & & & & & 8 & 9
\end{pmatrix}$$

Now using EVA we want to calculate the first row of A such that A will be assigned 10 eigenvalues each equal to 7 . The first row of A computed by EVA is as follows:

21.000100	-196.998600	646.753400	-1994.128000
-873.761200	-3130.766000	-4675.128000	-8100.312000
-11546.110000	-11095.480000		

The magnitude of this first row suggests this was a poorly conditioned problem. Now giving the full matrix A to EQRH3F we get the following eigenvalues:

4.653267 + j 1.855709	8.417780 + j 2.677467
4.653267 - j 1.855709	8.417780 - j 2.677467
6.345307 + j 2.945413	9.807051 + j 1.054002
6.345307 - j 2.945413	9.807051 - j 1.054002
7.531523	4.021682

As we can see no one of the above eigenvalues is 7 . The above example does not necessarily mean that the algorithm gave an inaccurate first row for A . The difficulty probably arises because unreduced upper Hessenberg matrices are always defective when they have multiple eigenvalues and this happens because if λ is an eigenvalue of A then $\text{rank}(A-\lambda I) = n-1$; so there is one and only one linearly independent eigenvector x such that $(A-\lambda I)x = 0$, if λ is a multiple eigenvalue then it has only one linearly independent eigenvector and so A is defective. In this example the Jordan canonical form of A is a 10×10 Jordan block, the eigenvalues of which are very ill-conditioned [12, p. 301], [13], and so cannot be computed accurately.

```
      SUBROUTINE EVA  (A,N,N1,TESTL,RSTACK,SSTACK,N2,QSTACK)
C
C
C     EVA IS A SUBROUTINE WHICH BY CALLING TWO OTHER SUBROUTINES (ORW
C     AND BACKW) CALCULATES AND RETURNS THE FIRST ROW OF A SINGLE
C     PRECISION NXN MATRIX  A  IN ORDER TO GIVE CERTAIN DESIRED
C     EIGENVALUES (1,2,3,..) WHICH MAY OCCUR IN COMPLEX CONJUGATE
C     PAIRS.
C
C     ON ENTRY
C
C        A            SINGLE PRECISION NXN ARRAY WHICH CONTAINS THE
C                     MATRIX WHOSE FIRST ROW WE WILL CALCULATE. THE
C                     FIRST ROW OF  A  IS IGNORED AND OVERWRITTEN IN
C                     THE COMPUTATION.
C
C        N            INTEGER.
C                     N  IS THE ORDER OF  A
C
C        N1           INTEGER.
C                     N1  IS THE NUMBER OF ELEMENTS OF THE ARRAY  TESTL
C                     WHERE N1=2*N.
C
C        TESTL        SINGLE PRECISION ONE DIMENSIONAL ARRAY OF ORDER  N1
C                     WHICH CONTAINS THE EIGENVALUES WHICH ARE GOING
C                     TO BE THE EIGENVALUES OF  A .
C                     EVERY EIGENVALUE OCCUPIES TWO CONSECUTIVE PLACES
C                     OF THE ARRAY  TESTL, THE FIRST FOR ITS REAL PART
C                     AND THE SECOND FOR ITS IMAGINARY PART.
C                     COMPLEX CONJUGATE PAIRS MUST APPEAR TOGETHER.
C
C        RSTACK       SINGLE PRECISION NXN ARRAY.
C                     RSTACK IS A WORK ARRAY.
C
C        SSTACK       INTEGER ONE DIMENSIONAL ARRAY OF ORDER  N .
C                     SSTACK IS A WORK ARRAY.
C
C        N2           INTEGER.
C                     N2 IS THE NUMBER OF ROWS OF THE ARRAY  QSTACK
C                     WHERE N2=(N*(N-1)/2) + 1 .
C
C        QSTACK       SINGLE PRECISION N2X3 ARRAY.
C                     QSTACK IS A WORK ARRAY.
C
C
C     ON RETURN
C
C        A            SEE THE PREVIOUS DISCUSSION ABOUT  A .
C                     ON RETURN WE HAVE THE DESIRED FIRST ROW OF  A,
C                     THE REST OF  A  IS AS ON ENTRY.
C
C
C
      INTEGER N,SSTACK(N),INDEX,ISTEP,RSTEP
      REAL A(N,N),TESTL(N1),QSTACK(N2,3),RSTACK(N,N)
C
C
C     WE REARRANGE THE EIGENVALUES SUCH THAT THE LAST TWO EIGENVALUES
C     CONSIST EITHER A COMPLEX CONJUGATE PAIR OR A PAIR OF TWO REAL
```

```
C     EIGENVALUES.
C
C
          IF (ABS(TESTL(N1)).GT.0.5E-5.OR.ABS(TESTL(N1)).LE.0.5E-5.AND
     1       ABS(TESTL(N1-2)).LE.0.5E-5) GO TO 5
          C=TESTL(N1-1)
          TESTL(N1-1)=TESTL(N1-5)
          TESTL(N1-5)=C
          TESTL(N1)=TESTL(N1-4)
          TESTL(N1-4)=0.
     5    CALL FORW (N,N1,INDEX,ISTEP,RSTEP,A,TESTL,OSTACK,RSTACK,SSTACK,
     1               N2)
          CALL BACKW (N,N1,INDEX,ISTEP,RSTEP,A,TESTL,OSTACK,RSTACK,N2,
     1                SSTACK)
          RETURN
          END


C
          SUBROUTINE FORW (N,N1,INDEX,ISTEP,RSTEP,A,TESTL,OSTACK,RSTACK
     1                    ,SSTACK,N2)
C
C     FORW SUBROUTINE PERFORMES THE FORWARD SWEEP.
C
C
          COMMON /B1/ C,S,R,M
          INTEGER  RSTEP,EV,ISTEP,SW,M,DROP,INDEX,N,LINE,DROP1,SSTACK(N)
          REAL  T,C,S,A(N,N),TESTL(N1),OSTACK(N2,3),RSTACK(N,N),R,AN,AN1,
     1          AN2
C
C
C     INITIALIZATION
C
C
          INDEX=1
          ISTEP=1
          RSTEP=1
          EV=1
C
C
C     WE DETERMINE THE FORWARD SWEEP, STARTING FROM  N  UNTIL  2  .
C
C
    30    M=N-RSTEP+1
          DROP=N-M
C
C
C     THE LAST TWO EIGENVALUES FOR SIMPLICITY ARE TREATED TOGETHER.
C
C
          IF (M.EQ.2) GO TO 10
C
C
C     WE CHECK IF THE COMING EIGENVALUE IS COMPLEX OR REAL.
C
```

```
C
      IF (ABS(TESTL(EV+1)).GT.0.5E-5) GO TO 5
C
C
C     IN THIS PROGRAM SEGMENT WE DEAL WITH REAL EIGENVALUES.
C
C
      SSTACK(ISTEP)=1
      ISTEP=ISTEP+1
C
C
C     WE DETERMINE THE TRANSFORMATION MATRIX WHICH WILL HELP US MOVE ONE
C     FORWARD STEP, BY CALCULATING ITS LAST COLUMN.
C
C
      SS=A(N,N-1)
      CC=A(N,N)-TESTL(EV)
C
C
C     WE APPLY THE TRANSFORMATION.
C
C
      CALL RROT (QSTACK,N-1,N,SS,CC,A,INDEX,N,N2)
      EV=EV+2
      SW=1
      CALL LROT (A,N-1,N,SW,N)
      LINE=DROP+4
      I=LINE-1
      IF(M.EQ.3) GO TO 15
      DO 20 II=LINE,N
         I=N+LINE-II
         SS=A(I,I-2)
         CC=A(I,I-1)
         CALL RROT (QSTACK,I-2,I-1,SS,CC,A,INDEX,N,N2)
         CALL LROT (A,I-2,I-1,SW,N)
20    CONTINUE
      I=I-1
15    SS=A(I,I-2)
      CC=A(I,I-1)
      CALL RROT (QSTACK,I-2,I-1,SS,CC,A,INDEX,N,N2)
      DROP1=DROP+1
      DO 25 JJ=DROP1,N
         J=N+DROP1-JJ
         RSTACK(RSTEP,J)=A(I-1,J)
25    CONTINUE
      RSTEP=RSTEP+1
      GO TO 30
C
C
C     IN THIS PROGRAM SEGMENT WE DEAL WITH TWO EIGENVALUES WHICH ARE
C     COMPLEX CONJUGATES.
C     SO WE DETERMINE THE TRANSFORMATION MATRIX WHICH WILL HELP US MOVE
C     TWO FORWARD STEPS BY CALCULATING ITS LAST COLUMN.
C
C
5     AN2=A(N-1,N-2)*A(N,N-1)
      AN1=A(N,N-1)*(A(N-1,N-1)+A(N,N)-2.*TESTL(EV))
      AN=A(N,N)**2+A(N,N-1)*A(N-1,N)-A(N,N)*2.*TESTL(EV)+TESTL(EV)**2
```

```
     1       ITESTL(EV+1)**2
             EV=EV+4
             QSTACK(ISTEP)=2
             ISTEP=ISTEP+1
C
C
C     WE APPLY THE TRANSFORMATION
C
C
             CALL RROT (QSTACK,N-2,N-1,AN2,AN1,A,INDEX,N,N2)
             SW=1
             CALL LROT (A,N-2,N-1,SW,N)
             T=T
             CALL RROT (QSTACK,N-1,N,T,AN,A,INDEX,N,N2)
             SW=2
             CALL LROT (A,N-1,N,SW,N)
             LINE=DROP+5
             I=LINE-1
             IF (M.EQ.4) GO TO 35
             DO 40 II=LINE,N
                I=N+LINE-II
                SS=A(I,I-3)
                CC=A(I,I-2)
                CALL RROT (QSTACK,I-3,I-2,SS,CC,A,INDEX,N,N2)
                SW=1
                CALL LROT (A,I-3,I-2,SW,N)
                SS=A(I,I-2)
                CC=A(I,I-1)
                CALL RROT (QSTACK,I-2,I-1,SS,CC,A,INDEX,N,N2)
                SW=2
                CALL LROT (A,I-2,I-1,SW,N)
    40       CONTINUE
             I=I-1
    35       SS=A(I,I-3)
             CC=A(I,I-2)
             CALL RROT (QSTACK,I-3,I-2,SS,CC,A,INDEX,N,N2)
             SS=A(I,I-2)
             CC=A(I,I-1)
             CALL RROT (QSTACK,I-2,I-1,SS,CC,A,INDEX,N,N2)
             DROP1=DROP+1
             DO 45 JJ=DROP1,N
                J=N+DROP1-JJ
                RSTACK(RSTEP,J)=A(I-2,J)
                RSTACK(RSTEP+1,J)=A(I-1,J)
    45       CONTINUE
             RSTEP=RSTEP+2
             GO TO 30
C
C
C    SPECIAL TREATMENT FOR THE  2X2  MATRIX.
C    INSTEAD OF MOVING ONE FORWARD STEP AND THEN ONE BACKWARD STEP,IN
C    ORDER TO CALCULATE THE FIRST ROW OF THE 2X2 MATRIX, WE DO THIS
C    DIRECTLY USING TWO FORMULAS.
C
C
    10       A(N-1,N)=-(A(N,N)**2-(TESTL(EV)+TESTL(EV+2))*A(N,N)+TESTL(EV)*
     1                TESTL(EV+2)+TESTL(EV+1)**2)/A(N,N-1)
             A(N-1,N-1)=TESTL(EV)+TESTL(EV+2)-A(N,N)
```

```
      RETURN
      END



      SUBROUTINE RROT (QSTACK,J1,J2,S1,C1,A,INDEX,N,N2)
C
C
C     RROT IS A SUBROUTINE WHICH MULTIPLIES THE MATRIX  A  FROM THE
C     RIGHT BY A ROTATION MATRIX, COMBINING THE J1,J2 COLUMNS OF  A .
C
C
         COMMON /B1/ C,S,R,M
         INTEGER M,N,I1,J1,J2
         REAL C,S,R,C1,S1,A(N,N),QSTACK(N2,3),A1,A2
         C=C1
         S=S1
         R=SQRT(S**2+C**2)
         I1=N-M+2
         DO 5 I=I1,N
            A1=(A(I,J1)*C-A(I,J2)*S)/R
            A2=(A(I,J1)*S+A(I,J2)*C)/R
            A(I,J1)=A1
            A(I,J2)=A2
    5    CONTINUE
         QSTACK(INDEX,1)=C
         QSTACK(INDEX,2)=S
         QSTACK(INDEX,3)=R
         INDEX=INDEX+1
      RETURN
      END



      SUBROUTINE LROT (A,I1,I2,SW,N)
C
C
C     LROT IS A SUBROUTINE WHICH MULTIPLIES THE MATRIX  A  FROM THE LEFT
C     BY A ROTATION MATRIX COMBINING THE I1,I2 ROWS OF  A .
C
C
C
         COMMON /B1/ C,S,R,M
         INTEGER I1,I2,N,J1,SW
         REAL A(N,N),C,S,R,A1,A2
         J1=I1-SW
         DO 5 J=J1,N
            A1=(A(I1,J)*C-A(I2,J)*S)/R
            A2=(A(I1,J)*S+A(I2,J)*C)/R
            A(I1,J)=A1
            A(I2,J)=A2
    5    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE BACKW (N,N1,INDEX,ISTEP,RSTEP,A,TESTL,QSTACK,RSTACK,N2,
     1           SSTACK)
C
C
C     BACKW  IS A SUBROUTINE WHICH PERFORMES THE BACKWARD SWEEP.
C
C
      REAL A(N,N),TESTL(N1),QSTACK(N2,3),RSTACK(N,N),C,S,A1,A2,A3,
     1     C1,S1
      INTEGER SSTACK(N),NO,LINE,RSTEP
      INDEX=INDEX-1
      R=QSTACK(INDEX,3)
      C=QSTACK(INDEX,1)/R
      S=QSTACK(INDEX,2)/R
   20 ISTEP=ISTEP-1
      RSTEP=RSTEP-SSTACK(ISTEP)
      M=N-RSTEP+1
      LINE=RSTEP
C
C
C     WE DETERMINE IF THE CURRENT BACKWARD STEP IS ASSOCIATED WITH A
C     REAL EIGENVALUE OR WITH A COMPLEX CONJUGATE PAIR OF EIGENVALUES.
C
C
      IF (SSTACK(ISTEP).EQ.2) GO TO 5
C
C
C     HERE WE MOVE ONE BACKWARD STEP BECAUSE THE CURRENT STEP IS
C     ASSOCIATED WITH A REAL EIGENVALUE.
C
C
      NO=M-1
C
C
C     FIRST WE FIND HOW THE FIRST ROW OF A SUBMATRIX OF  A  ,ASSOCIATED
C     WITH THE CURRENT STEP, WAS, BEFORE THE EFFECT OF THE
C     TRANSFORMATION MATRIX FROM THE LEFT.
C
C
      A(LINE,LINE)=-C*RSTACK(RSTEP,RSTEP)/S
      LINE1=LINE+1
      DO 10 J=LINE1,N
         A(LINE,J)=(-C*RSTACK(RSTEP,J)+A(LINE+1,J))/S
   10 CONTINUE
C
C
C     HERE WE FIND THE CORRECT FIRST ROW OF THE SAME SUBMATRIX OF  A .
C
C
      DO 15 J=1,NO
         A1=A(LINE,LINE+J-1)*C+A(LINE,LINE+J)*S
         A2=-A(LINE,LINE+J-1)*S+A(LINE,LINE+J)*C
         A(LINE,LINE+J-1)=A1
         A(LINE,LINE+J)=A2
         INDEX=INDEX-1
         IF (INDEX.EQ.0) GO TO 35
         R=QSTACK(INDEX,3)
         C=QSTACK(INDEX,1)/R
```

```
            S=QSTACK(INDEX,2)/R
   15     CONTINUE
          GO TO 20
C
C
C     HERE WE MOVE TWO BACKWARD STEPS BECAUSE THE CURRENT STEP IS
C     ASSOCIATED WITH A COMPLEX CONJUGATE PAIR OF EIGENVALUES.
C
C
    5     R1=QSTACK(INDEX-1,3)
          C1=QSTACK(INDEX-1,1)/R1
          S1=QSTACK(INDEX-1,2)/R1
          NO=2*M-4
C
C
C     FIRST WE FIND HOW THE FIRST ROW OF A SUBMATRIX OF  A  ,ASSOCIATED
C     WITH THE CURRENT STEP, WAS, BEFORE THE EFFECT OF THE
C     TRANSFORMATION MATRIX FROM THE LEFT.
C
C
          A(LINE,LINE)=(-S*C1*RSTACK(RSTEP,RSTEP)-C*RSTACK(RSTEP+1,RSTEP)
         1             )/(S*S1)
          A(LINE,LINE+1)=(-S*C1*RSTACK(RSTEP,RSTEP+1)-C*RSTACK(RSTEP+1,
         1              RSTEP+1))/(S*S1)
          LINE2=LINE+2
          DO 25 J=LINE2,N
            A(LINE,J)=(-C*C1*RSTACK(RSTEP,J)-C*RSTACK(RSTEP+1,J)+
         1             A(LINE+2,J))/(S*S1)
   25     CONTINUE
          NO=NO/2
C
C
C     HERE WE FIND THE CORRECT FIRST ROW OF THE SAME SUBMATRIX OF  A  .
C
C
          DO 30 J=1,NO
             A1=A(LINE,LINE+J-1)*C1+A(LINE,LINE+J)*C*S1
         1      +A(LINE,LINE+J+1)*S*S1
             A2=-A(LINE,LINE+J-1)*S1+A(LINE,LINE+J)*C*C1+
         1      A(LINE,LINE+J+1)*S*C1
             A3=-A(LINE,LINE+J)*S1+A(LINE,LINE+J+1)*C
             A(LINE,LINE+J-1)=A1
             A(LINE,LINE+J)=A2
             A(LINE,LINE+J+1)=A3
             INDEX=INDEX-2
             IF (INDEX.EQ.0) GO TO 35
             R=QSTACK(INDEX,3)
             C=QSTACK(INDEX,1)/R
             S=QSTACK(INDEX,2)/R
             R1=QSTACK(INDEX-1,3)
             C1=QSTACK(INDEX-1,1)/R1
             S1=QSTACK(INDEX-1,2)/R1
   30     CONTINUE
          GO TO 20
   35   RETURN
        END
```

REFERENCES

[1] H.H. Rosenbrock, "State-space and Multivariable Theory". London: Nelson, 1970.

[2] C.C. Paige, "Properties of Numerical Algorithms Related to computing controllability", IEEE Transactions on Automatic Control, Feb. 1981, Vol. AC-26, No. 1, pp. 130-138.

[3] D.G. Luenberger, "Introduction to Dynamic Systems, Theory, Models and Applications", New York: John Wiley, 1979.

[4] E.B. Lee and L. Markus, "Foundations of optimal control theory". Wiley, 1967.

[5] R.E. Kalman, Y.C. Ho and K.S. Narendra, "Controllability of linear dynamical systems". Contrib. Diff. Equations 1, No. 2, pp. 189-213 (1963).

[6] M.L.J. Hautus, "Controllability and observability conditions of linear autonomous systems" in Proc. Kon. Akad. Wetensh. Ser. A., Vol. 72, pp. 443-448, 1969.

[7] R.N. Clark, "Introduction to Automatic control systems", Wiley, New York, 1962.

[8] E.J. Davison, W. Gesing, and S.H. Wang, "An algorithm for obtaining the minimal realization of a linear time-invariant system and determining if a system is stabilized-detectable", IEEE Trans. Automat. Contr., Vol. AC-23, pp. 1048-1054, Dec. 1978.

[9] W.M. Wonham, "On Pole Assignment in Multi-input controllable linear systems", IEEE Trans. Automatic Contror, Vol. AC-12, December (1967), pp. 660-665.

[10] D.G. Luenberger, "Canonical Forms for Linear Multivariable Systems", IEEE Transactions on Automatic Control, Vol. AC-12, No. 3, June (1967), pp. 290-293.

[11] D.G. Luenberger, "Observers for Multivariable systems" IEEE Transactions on Automatic Control, Vol. AC-11, No. 2, April (1966), pp. 190-197.

[12] G.W. Stewart, "Introduction to Matrix Computations", New York: Academic Press, 1973.

[13] J.H. Wilkinson,"The Algebraic Eigenvalue Problem,"London: Oxford University Press, 1965.

[14] C.L. Lawson and R.J. Hanson, "Solving Least Squares Problems", Englewood Cliffs, N.J.: Prentice-Hall, 1974.

[15] B.C. Moore, "Singular value analysis of linear systems", in Proc. 1978 IEEE Conference on Decision and Control, San Diego, CA, pp. 66-73, Jan. 1979.

[16] B.C. Moore, "Computational problems with modal analysis", Sixteenth Annual Allerton Conf. on Communication, Control, and Computing, Oct. 1978.

[17] P. Van Dooren, "The Generalized Eigenstructure Problem in Linear System Theory", IEEE Transactions on Automatic Control, Vol. AC-26, No. 1, pp. 111-129, Feb. 1981.

[18] P. Van Dooren, A. Emami-Naeini, and L. Silverman, "Stable extraction of the Kronecker structure of pencils," in Proc. 17th IEEE Conf. Decision Contr., Jan. 1979, pp. 521-524.

[19] R.V. Patel, "Computation of matrix fraction descriptions of linear time-invariant systems," in Proc. 18th IEEE Conf. Decision Contr., Dec. 1979, pp. 325-328.

[20] J.H. Wilkinson, "Linear differential equations and Kronecker's canonical form," in Recent advances in Numerical Analysis, C. de Boor and G.H. Golub, Eds., New York: Academic, 1978.

[21] J.H. Wilkinson, "Kronecker's canonical form and the QZ algorithm", Nat. Phys. Lab., Teddington, Middlesex, England, Rep. DNACS 10/78, Nov. 1978.

[22] P. Van Dooren, "The Computation of Kronecker's canonical form of a singular pencil," Linear Algebra and Its Applications, Vol. 27, pp. 103-140, 1979.

[23] P. Van Dooren, "The generalized eigenstructure problem applications in linear systems theory," Ph.D. dissertation, Katholieke Univ. Leuven, Leuven, Belgium, May 1979.

[24] R.W. Brockett and C.I. Byrnes, "Multivariable Nyquist Criteria, Root Loci, and Pole Placement: A Geometric Viewpoint,", IEEE Transactions on Automatic Control, Vol. AC-26, pp. 271-284, Feb. 1981.

[25] H. Kimura, "Pole assignment by gain output feedback," IEEE Trans. Automat. Contr. Vol. AC-20, pp. 509-516, 1975.

[26] H. Kimura, "A further result in the problem of pole assignment by output feedback," IEEE Trans. Autom. Contr., Vol. AC-22, pp. 458-463, 1977.

[27] E.J. Davison, "On pole assignment in linear systems with incomplete state feedback," IEEE Trans. Autom. Contr. (Short Papers), Vol. AC-15, pp. 348-351, June 1970.

[28] E.J. Davison and R. Chatterjee, "A note on pole assignment in linear system with incomplete state feedback," IEEE Trans. Autom. Contr. (Corresp.), Vol. AC-16, pp. 98-99, Feb. 1971.

[29] B. Sridhar and D.P. Lindorff, "Pole-placement with constant gain output feedback," Int. J. Contr. Vol. 18, pp. 993-1003, Nov. 1973.

[30] E.J. Davison and S.H. Wang, "On Pole Assignment in Linear Multivariable Systems Using Output Feedback," IEEE Trans. Autom. Contr., Vol. AC-20, pp. 516-518, Aug. 1975.

[31] E.J. Davison and S.G. Chow, "An algorithm for the assignment of closed-loop poles using output feedback in large linear multivariable systems," IEEE Trans. Automat. Contr., Vol. AC-18, pp. 74-75, Feb. 1973.

[32] D.Q. Mayne, P. Murdoch, "Modal Control of Linear Time Invariant Systems," Int. J. Control, Vol. 11, pp. 223-227, 1970.