Computational Methods for Multivariable System Analysis and Design

Pradeep Misra

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy at
Concordia University
Montréal, Québec, Canada

August, 1987

ISBN 0-315-37097-1

ABSTRACT

Computational Methods for Multivariable System Analysis and Design

Pradeep Misra Ph. D.,
Concordia University, 1987.

This thesis is concerned with developing computational algorithms for some of the frequently encountered problems in linear multivariable system analysis and design. The results presented here can be used to translate theoretical concepts into algorithms that can be implemented to solve practical problems.

The computational methods presented here rely heavily on numerical methods that are used routinely in matrix computations. It is shown that use of several condensed forms of 4-tuples (A,B,C,E), *e.g.*, upper Schur form, real Schur form, upper (lower) Hessenberg form, and block upper (lower) Hessenberg forms, enables us to apply sound, well-established techniques from numerical linear algebra to solve various problems of multivariable system analysis and design.

In this thesis we consider two types of systems: centralized and decentralized. For centralized systems, we treat the problems of eigenvalue assignment by means of state as well as constant and dynamic output feedback and computation of transfer function and frequency response matrices. The eigenvalue assignment problem is treated as the converse of the algebraic eigenvalue problem and a variant of the QR algorithm is developed to compute the state and output feedback matrices. The computational techniques for transfer function and frequency response matrices are based on the use of a determinant identity and are significantly more efficient than existing methods. For decentralized systems, we provide a new characterization of decentralized fixed modes and develop an efficient and reliable method for computing these modes. The problem of eigenvalue assignment in decentralized systems is created by extending the results of eigenvalue assignment using output feedback in centralized systems. Numerical performance of the algorithms proposed in this thesis is illustrated by applying them to several practical systems.

To My Parents

Prem and Mahesh

ACKNOWLEDGEMENTS

Words are not enough to express my sincere gratitude to Professor R.V. Patel, who has been my guide, friend and mentor during these last few years. Without his constant guidance, suggestions and encouragement, this work would not have been possible. I owe my knowledge of system theory to him.

I consider myself fortunate to have made acquaintance with so many wonderful people that I could fill up a few pages with their names. Some of them would always be with me in form of fond memories of joyous moments that we shared together. It is with great pleasure that I acknowledge support and motivation provided by Babu, Bala, Costas, Danny, Eddie, Kin, Lokesh and Shahrokh.

It is not always that one gets such selfless friendship and companionship that Cathy has provided me during the rather strenuous period of the final stages of this work. Without an occasional break from the work, I wonder how much longer it would have taken me to complete it.

Last but not the least, it is with pride that I acknowedge the support and inspiration of my parents Prem and Mahesh, my brothers Neeraj and Manoj and my sisters Kamana and Kalpana — back home. I believe that is where it all started.

TABLE OF CONTENTS

ABSTRACT iii
ACKNOWLEDGEMENTS vi
TABLE OF CONTENTS vii
CHAPTER I: INTRODUCTION 1
1.1 Motivation 1
1.2 Review of Reliable Analysis and Design Methods 3
1.2.1 Analysis 4
1.2.2 Design 7
1.2.3 General 9
1.3 Outline of the Thesis 11
1.4 References 14
CHAPTER II: NOTATION AND PRELIMINARIES 17
2.1 Notation 17
2.2 Matrix Theory and Numerical Analysis 19
2.2.1 Matrices and Vectors 19
2.2.2 Stability and Conditioning 25
2.3 Reduction of a System to Condensed Forms 29
2.4 References 34
CHAPTER III: EIGENVALUE ASSIGNMENT BY STATE FEEDBACK 35
3.1 Statement of the Problem 35
3.2 Theoretical Developments for Single-Input Eigenvalue Assignment Problem 37
3.2.1 Explicitly Shifted Eigenvalue Assignment 37
3.2.2 Implicitly Shifted Eigenvalue Assignment 40
3.3 An Algorithm for Eigenvalue Assignment in Single-Input Systems 44
3.4 Extension to the Multi-Input Eigenvalue Assignment Problem 52
3.5 Some Variations on the Multi-Input Eigenvalue Assignment Scheme 55
3.6 Numerical Examples 58
3.7 Concluding Remarks 61
3.8 References 63
CHAPTER IV: EIGENVALUE ASSIGNMENT BY OUTPUT FEEDBACK 65
4.1 Statement of the Problem 65
4.2 Eigenvalue Assignment by Constant Gain Output Feedback 67
4.2.1 Explicitly Shifted Eigenvalue Assignment for Single-Input Mult-Output Systems 67
4.2.2 Theoretical Considerations in Eigenvalue Assignment for Multi-Input Multi-Output Systems 75
4.3 Algorithms for Eigenvalue Assignment by Constant Gain Output Feedback 78

4.3.1 An Algorithm for Eigenvalue Assignment in Single-Input Multi-Output Systems 78

4.3.2 An Algorithm for Eigenvalue Assignment in Multi-Input Multi-Output Systems 85

4.4 Eigenvalue Assignment by Dynamic Output Feedback 87

4.5 Numerical Examples 91

4.6 Concluding Remarks 94

4.7 References 96

CHAPTER V: CONTROL OF DECENTRALIZED MULTIVARIABLE SYSTEMS 97

5.1 Introduction 97

5.2 Preliminaries 99

5.3 Characterization and Computation of Decentralized Fixed Modes 102

5.3.1 Characterization of Decentralized Fixed Modes 102

5.3.2 Computation of Decentralized Fixed Modes 105

5.4 Discussion of the Results 111

5.4.1 General Remarks 111

5.4.2 Examples 114

5.5 Numerical Examples 116

5.6 Decentralized Eigenvalue Assignment Problem 121

5.6.1 Problem Statement 122

5.6.2 Preliminary Considerations 123

5.6.3 Main Results 126

5.7 Concluding Remarks 132

5.8 References 134

CHAPTER VI: TRANSFER FUNCTION AND FREQUENCY RESPONSE MATRICES 136

6.1 Introduction 136

6.2 Preliminary Considerations 138

6.3 Computation of Transfer Function Matrices 139

6.3.1 An Algorithm for Computing Transfer Function Matrices 140

6.3.2 Discussion 143

6.3.3 Numerical Examples 145

6.4 Computation of Frequency Response Matrices 146

6.4.1 An Algorithm for Computing Frequency Response Matrices 148

6.4.2 Discussion 149

6.4.3 Operations Count 152

6.4.4 Numerical Examples 154

6.5 Concluding Remarks 155

6.6 References 156

CHAPTER VII: CONCLUSIONS AND FUTURE WORK 157

CHAPTER I

INTRODUCTION

1.1. MOTIVATION

In the last two decades numerous techniques have been proposed for the analysis and design of various aspects of multivariable systems. However, until recently [1-14], researchers have paid very little attention to the numerical aspects of the proposed algorithms. It is becoming increasingly clear that the translation of elegant theoretical ideas and design and analysis techniques into numerically reliable algorithms is far from trivial. Fortunately, significant developments and advances have been made in recent years in the areas of numerical analysis and numerical linear algebra which have not only highlighted the problems inherent in many of the existing multivariable system analysis and design methods, but have also shown ways to alleviate the difficulties (mainly computational) in a number of cases. In some cases this has led to alternate methods of solution and in many others to refinement of the existing methods to ensure better numerical performance, applied to realistic problems. The computational complexity involved in solving such problems necessitates consideration of various numerical issues. Unfortunately, specialists in control theory are frequently unaware of how advanced techniques of numerical linear algebra may be applied to their problems. As a result, a common belief among some researchers is that if an algorithm works for a few (usually low order well-conditioned) examples, there is no reason why the same algorithm should not work in general. They are often surprised by the error that accumulates when the same technique is used for a high order or badly conditioned problem. Unless the algorithm is carefully planned and implemented, it could encounter numerical problems such as "floating point overflow (underflow)" or "round-off errors", as discussed in the next chapter.

Development of superior computational schemes is especially important with an increasingly wider use of Computer-Aided Control System Design packages (CACSD) [2]. In application areas such as aerospace, power systems, *etc.*, system models tend to be of considerably high order (~100). When applied to such high order systems, conventional techniques would have limited success despite being based on extremely sound mathematical principles. This is mainly because most of these techniques were devised as analytical solutions to the problems under consideration, rather than to provide numerically accurate methods of solving the problem. If a CACSD package used for analysis and design of such systems is based on poor numerical techniques, it would be virtually impossible to obtain reliable results. The recent H^∞ theory [15-17] is perhaps a perfect example to support the above statement. The mathematical foundations of H^∞ theory are very strong However, one of the reasons why this theory has not yet become popular in actual system design is that the researchers have had limited success in formulating computational algorithms for implementation of these theories in solving realistic problems. The importance of this theory would be fully realized only when it could be translated into design packages based on computationally reliable algorithms.

In designing a reliable algorithm, two of the most important properties that must be taken into account are (1) minimization of the number of operations required by the algorithm and (2) the type of operations required by the algorithm. Each time an operation is performed, the output of the operation is stored as the "closest" machine representable number by rounding the number off. As the algorithm proceeds, it uses the previously approximated numbers for the next operation. As the number of operations increase, the results get increasingly farther from the actual ones because of these errors. Therefore, it is desirable to minimize the number of operations that an algorithm requires. The second property is equally important. If an algorithm requires a large number of operations that are known to lead to numerical errors (*e.g.*, addition of a very large number to a very small number, successive divisions by numbers that are much smaller than 1 – leading to floating point overflow or successive multiplications by numbers that are much smaller than unity – leading to floating point underflow), one cannot expect accurate results when it is implemented. In designing algorithms, such operations should be kept to a minimum if they cannot be eliminated altogether. Algorithms that do not take these properties into account, usually lead to analysis and design methods that are computationally *unreliable*.

The control systems literature has a large number of seemingly simple problems with mostly unreliable methods for solving them. When results are obtained using these unreliable methods, it is possible to make serious errors in interpreting them. As pointed out above, a lack of reliable computational techniques has been a major drawback in various approaches to control system analysis and design. This thesis is an attempt to provide reliable solutions for some of the basic problems frequently encountered in control system analysis and design.

1.2. Review of Reliable Analysis and Design Methods

In this thesis, we consider linear time-invariant (LTI) multivariable system in their *state-space* representation, given by:

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{1.2.1a}$$

$$y(t) = Cx(t) + Eu(t) \tag{1.2.1b}$$

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, $y(t) \in \mathbb{R}^p$ or their *transfer function* representation defined as

$$W(s) = C\left(sI - A\right)^{-1}B + E. \tag{1.2.2}$$

The matrices A, B, C and E are constant matrices with appropriate dimensions. The system described by (1.2.1) is usually denoted by (A, B, C, E).

Some of the basic problems encountered in studying control systems described by (1.2.2) can be classified as (1) Analysis, (2) Design and (3) General. In the rest of this section, we will discuss some of the reliable computational algorithms dealing with various analysis and design aspects of linear multivariable control that have appeared recently in the literature. Note that only the algorithms with good numerical properties are considered. For information on the more conventional methods used for analysis and design, the reader is referred to several excellent text books [18-21].

1.2.1. Analysis

In this section, we outline some important properties of multivariable systems. Conventional as well as numerically reliable techniques for studying them are discussed.

Controllability and Observability

The conventional way of checking controllability of a LTI system is by examining the rank of the *controllability matrix* Φ_c, defined as [18-21],

$$\Phi_c \doteq \left[B ,AB ,A^2B ,...,A^{n-1}B\right].$$

If rank $\Phi_c < n$, then the system is said to be uncontrollable and the rank deficiency indicates the number of uncontrollable modes of the system.

Numerically, the rank test of the matrix Φ_c is an expensive and unreliable method for checking controllability. Forming the controllability matrix Φ_c requires a large number of matrix multiplications. Moreover since the operations are being performed in finite precision arithmetic, we may lose valuable information in forming the products. An alternative test for checking the controllability of a system is based on reducing the pair (A ,B) to a "condensed form" called block upper Hessenberg form [4-7]. The details of the reduction to this form, as well as its significance, are discussed in the next chapter. It should be mentioned, however, that this reduction is considerably more reliable and efficient than the conventional method of performing the rank test of the controllability matrix.

Order-Reduction

If the system under consideration has any uncontrollable and/or unobservable modes, then the order of the system can be reduced by removing these modes without affecting the input-output behaviour of the system. The order reduction is achieved by transforming the given system to a block upper Hessenberg form to remove the uncontrollable modes and a further reduction of the controllable subsystem to a block lower Hessenberg form to remove the unobservable modes of the system [4-7]. The above reductions may be performed using *singular value decomposition* or *orthogonal triangularization* [22-25]. Softwara code for singular value decomposition as well as orthogonal triangularization can be found in FORTRAN packages such as EISPACK, LINPACK, *etc.*

Poles and Zeros of the System

Knowledge of poles and zeros of a system is important in analysis as well as design. *Poles* of the system (A, B, C, E) are the roots of the characteristic polynomial of the *state matrix* A. Computing roots of a polynomial is extremely sensitive to variations in its coefficients and is not recommended in general. A reliable way of computing the poles of the system is to solve an *algebraic eigenvalue problem* to determine the eigenvalues of A using the QR algorithm [22-25].

Multivariable systems have several different types of zeros associated with them. These zeros can be characterized as: (1) Element zeros – For a system described by a transfer-function matrix $W(s)$, an element zero is any value of s for which the numerator of an element $w_{ij}(s)$ vanishes. They are not of any special interest. (2) Decoupling zeros – For a system (A, B, C, E), the decoupling zeros consist of the uncontrollable and/or unobservable modes that do not appear in the corresponding transfer function matrix $W(s)$ [20]. These zeros can be computed efficiently, by reducing the system to block upper and lower Hessenberg forms. (3) Invariant zeros – These are the roots of monic greatest common divisors of all maximum order minors of $P(s) = \begin{bmatrix} sI-A & B \\ -C & 0 \end{bmatrix}$ [20,26], (4) Blocking zeros – These are defined as those values of s for which, given (A, B, C), $Cadj(sI-A)B = 0$ [27]. These zeros will be discussed further in Chapter V.

(5) Transmission zero – These are a very important type of zeros of the system [18-21]. A system described by a transfer function (1.2.2) can be transformed to its Smith-McMillan form [19,20] given by $M(S) = U(s)G(s)V(s) = \left[\alpha_i(s)/\beta_i(s)\right]$. Then, the roots of the numerator polynomials $\alpha_i(s)$ are the *transmission zeros* of the system. In [6,10,28,29], it is shown that computation of the transmission zeros is equivalent to solving a "generalized eigenvalue problem" [22-25], for which the numerically reliable QZ algorithm exists [30-32]. This approach is conceptually somewhat complicated, however, it is considerably superior numerically to computing transmission zeros via the Smith-McMillan form.

Evaluation of Time Response

The response of system (A,B,C,E) to step input, impulse input, *etc.*, is equivalent to evaluating expressions of the type [16-20]:

$$y(t) = Ce^{At}x(0) + \int_0^t Ce^{A(t-\tau)}Bu(\tau)d\tau .$$

For the convolution in the above equation, methods based on Padé approximation or reduction of A to a real Schur matrix are quite efficient. Of course, if A is a diagonal matrix, it would be even cheaper to evaluate the integral, but, reduction of A to a diagonal form is numerically not reliable. Several methods for computing the matrix exponential appear in [33].

Frequency Response Matrices

The *frequency response* of a system is another important representation of the characteristics of a system. For systems described in state-space by the 4-tuple (A,B,C,E), the frequency response matrix is given by $C\left(j\omega I - A\right)^{-1}B + E$. For evaluating frequency response matrices, accuracy as well as efficiency are equally important because, in general, we need to compute the matrix at a very large number of values of ω. If the matrix A is a full matrix, evaluating frequency response matrices is very expensive computationally. However, as shown in [34,35], if the state matrix A is reduced to a condensed form before evaluating the frequency response matrices, the order of computation can be reduced significantly. In [35] and Chapter VI, it is shown that further improvement in efficiency can be achieved by using the controllability and observability properties of a system.

1.2.2. Design

Several elegant theoretical approaches for design of systems exist [15-21,36]. However, there are only a few good numerical ones; most recent developments still need to be implemented in sound numerical ways. Here, we briefly review some of the design techniques that have good numerical properties. In practice, linear systems are usually obtained by linearization of a non-linear system about some nominal operating points [18-21]. To ensure that the original system has good tracking properties, the linearized system must be stable [18-21]. Stability and other properties of the systems such as transient response, *etc.*, can be improved by reassigning the poles and zeros of the system [18-21,26]. Some of the design techniques that have good numerical solutions are listed next.

Optimal Feedback Design by Solution of Matrix Riccati Equation

For the systems described by (1.2.1), the optimal linear quadratic control problem is stated as

$$minimize \int_0^\infty \left(x^T Q x + u^T R u \right) dt$$

where Q is a positive semi-definite matrix and R is a positive definite matrix [22-25]. The solution of the problem is given by

$$u_{opt} = -R^{-1} B^T P x \triangleq K_{opt}\, x$$

where $P \in \mathbb{R}^{n \times n}$ is the unique *positive semi-definite* solution of the *algebraic Riccati equation*

$$A^TP + PA - PBR^{-1}B^TP + Q = 0.$$

A reliable approach to solving the optimal control problem above is to compute the stable sub-space of the Hamiltonian matrix [37]

$$H = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix}$$

using the QR algorithm [14]. This approach has considerably better numerical properties compared to reducing the system to its diagonal form obtained by eigenvalue, eigenvector decomposition.

Eigenvalue Assignment

Design methods such as eigenvalue (pole) assignment by means of state and output feedback fall under this category. Although these are two of the most studied approaches for stabilizing systems, it is only recently that some good computational procedures for solving the eigenvalue assignment problems have appeared [8,9]. Some of the conventional techniques require reduction of the state matrix to "companion" form and then determine the feedback by comparing the coefficients of the characteristic polynomials of open-loop and closed-loop systems. But, because of the sensitivity of the roots of a polynomial to perturbations in its coefficients, this approach is numerically unreliable. Recently, several attempts have been made for providing numerically reliable methods for solving eigenvalue assignment problems These methods will be discussed in greater detail in Chapters III and IV.

Design of Observers

When all the states of a system (1.2.1) are not available for measurement, to implement a feedback of the form $u(t) = Kx(t)$, an "observer" that provides estimates of the unmeasurable states is employed [18-21]. If p is the number of independent outputs of the system, then an observer of order $(n-p)$ can be constructed with state $z(t)$ such that $z(t)$ approximates a linear combination of the system state given by $Kx(t)$, where

$K \in \mathbb{R}^{(n-l)\times m}$. The observer for the system (1.2.1) is given by

$$\dot{z}(t) = Dz(t) + Ty(t) + Ru(t)$$

Construction of the observer involves the solution of Sylvester's equation

$$KA - DK = TC$$

for K. The matrices D and T can be chosen arbitrarily, as long as D is stable. Usually the eigenvalues of the observer are made more negative than those of the system so that the states of the observer will converge rapidly to the state of the system. If D is chosen to be an upper triangular matrix and the system (A, B, C, E) is in block upper Hessenberg form, then a numerically reliable approach [38], can be employed to solve for K.

1.2.3. General

Here we consider techniques that are required for both analysis and design. They are basically concerned with the representation of systems in the appropriate "domain". The 4-tuple (A, B, C, E) is a "time" domain representation while $W(s)$ in (1.2.3) is a frequency domain or transfer function representation. Once a system has been represented in an appropriate domain, the available analysis and design tools can be employed.

Transfer Function Matrices

There are several powerful techniques for analysis and design that use the transfer function representation [16-21,39-41]. Therefore it is important to have numerical techniques for computing transfer function matrices from 4-tuples (A, B, C, E) accurately and efficiently. In [42-45], it is shown that computation of a transfer function matrix can be reduced to evaluating the poles and zeros of several single-input, single-output subsystems. Knowing the poles and zeros, the coefficients of numerator and denominator polynomials can be computed easily. However, this method is based on "iterative" solution of algebraic and generalized eigenvalue problems. If these problems are "ill-conditioned", then the computed denominator and numerator polynomials can have inaccurate coefficients. In [44,45] and Chapter VI, a direct approach for computing transfer function matrices is presented. This approach is considerably more accurate and less expensive than those in [42,43].

Matrix Fraction Descriptions

This is a natural generalization of the *scalar* transfer function representation. A transfer function matrix $W(s)$ can be represented as $W(s) = D^{-1}(s)N(s)$ (or $W(s) = N(s)D^{-1}(s)$) known respectively as left (right) matfix fraction description [18-21,36]. When $N(s)$ and $D(s)$ have no factors in common, the representation is known as a coprime matrix fraction description. This representation is of considerable interest in the *factorization approach* to analysis and design of systems [17]. A good numerical method for computation of matrix fraction descriptions from state-space representations appears in [5].

In the factorization approach, the central idea is to factor the transfer function matrix of a (not necessarily stable) system as a ratio of stable matrices. Formally, the problem can be stated as: Given the transfer function matrix $W(s)$, determine rational function matrices $N(s)$, $\overline{N}(s)$, $D(s)$, $\overline{D}(s)$, $U(s)$, $\overline{U}(s)$, $V(s)$, and $\overline{V}(s)$ such that (i) all of above matrices are stable rational matrices, (ii) $D(s)$ and $\overline{D}(s)$ are non-sigular, (iii) $W(s) = N(s)D^{-1}(s) = \overline{D}^{-1}(s)\overline{N}(s)$ and (iv)

$$\begin{bmatrix} V(s) & U(s) \\ -\overline{N}(s) & \overline{D}(s) \end{bmatrix} \begin{bmatrix} D(s) & -\overline{U}(s) \\ N(s) & \overline{V}(s) \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}$$

The above matrices can be easily evaluated using eigenvalue assignment by state feedback and a technique for computing transfer function matrices for LTI multivariable systems [17,46]. The above factorization is also called doubly coprime factorization of $W(s)$.

These are some of the problems in the control systems literature that have numerically reliable solutions. Several other problems *e.g.*, computation of multivariable system inverses using the "structure algorithm", [47], transformation of a transfer function matrix to its Smith-McMillan form [48], *etc.*, have varying degrees of numerical reliability. However, this is only a small subset of the problems that need to be examined from a numerical analysis point of view. A large number of problems still remains unsolved. For example, recent development of H^∞ theory has an excellent mathematical foundation. However, very little attention has been paid to developing reliable algorithms that would enable us to apply it to practical systems of moderately high orders.

1.3. OUTLINE OF THE THESIS

The thesis presents some new computational methods for analysis and design of linear multivariable systems. Chapter III and IV are mainly concerned with computation of feedback matrices, while Chapter V contains some analysis of "decentralized systems" followed by computation of decentralized feedback. Chapter VI falls under the 'General' category described above. New algorithms for evaluation of transfer function and frequency response matrices are proposed The main contents of each chapter are given next.

Chapter II: Preliminaries

This chapter introduces some of the concepts from matrix theory and numerical analysis that form the foundation of most of the computational methods existing in the literature and presented in this thesis Several useful condensed forms for linear multivariable systems are discussed. Notation to be used throughout the thesis is formalized.

Chapter III: Eigenvalue Assignment by State Feedback

This chapter introduces the problem of eigenvalue assignment by state feedback. A brief survey of existing techniques for solving the problem is followed by an outline of an approach that can be considered as the "converse" of the algebraic eigenvalue problem. Based on this approach, a computational algorithm for single-input systems is presented. The results are then extended to multi-input systems and the performance of the proposed method illustrated by means of numerical examples.

Chapter IV: Eigenvalue Assignment by Output Feedback

This is a generalization of the eigenvalue assignment problem considered in Chapter III. The underlying principles of the technique developed are the same as in Chapter III. The theory is developed for eigenvalue assignment by constant gain output feedback for single-input, multi-output systems. It is then shown how the case of multi-input, multi-output systems can be treated via a two-stage eigenvalue assignment problem. This viewpoint leads us to determine a rank 2 feedback. The results are then extended to the computation of dynamic output feedback compensators. Numerical examples are considered to illustrate the performance of various algorithms.

Chapter V: Control of Decentralized Systems

Decentralization of system is a natural generalization of the systems considered in (1.2.2). When the system has several non-interacting control stations and observation points, and only local feedback is permitted, the system is said to be decentralized. The first half of the chapter develops a computational method for determining the "fixed-modes" of decentralized systems These are the modes that cannot be moved using local output feedback. It is then shown that when the system does not have any decentralized fixed modes, we can generalize the results of Chapter IV to eigenvalue assignment in decentralized systems. A few theoretical issues remain to be solved. However, a first attempt at developing a good numerical technique for decentralized eigenvalue assignment is described.

Chapter VI: Transfer Function and Frequency Response Matrices

This chapter is concerned with computation of transfer function and frequency response matrices of systems given in their state-space representation. Using a determinant identity, we develop algorithms that are more accurate and efficient than the existing ones. These aspects of the algorithms are discussed in detail and illustrated by applying them to an ill-conditioned system.

Chapter VII: Conclusions and Future Work

This thesis is not a start of a new field nor culmination of an existing one: It is a continuation of the significant effort that is being made to bridge the gap between elegant theories and reliable computational techniques for their implementation. Several issues that arise from various considerations in the thesis are presented in this chapter. Possible extensions of the results to more specific problems or perhaps new problems are discussed. A brief section on the things that were *not* treated is included as a pointer for those who might wish to contribute to this exciting area of research.

It is perhaps worth mentioning at this point that the various steps in most of the algorithms presented in this thesis have been kept rather general and extensive explanatory remarks have been introduced where necessary. This serves a twofold purpose: First, it permits us to understand the actual concept behind that step and second, it gives freedom to a software specialist to code these algorithms as portable software.

1.4. REFERENCES

[1] *Control Systems Magazine*, (Special issue on CACSD), Vol. 2, Dec 1982.

[2] *Proc. IEEE*, (Special Section on CACSD), Vol. 72, Dec. 1984.

[3] *AMS-IMS-SIAM 1984 Joint Summer Research Conference on Linear Algebra and its Role in Systems Theory*, July 29 - Aug. 4, 1984.

[4] C.C. Paige, "Properties of Numerical Algorithms Related to Computing Controllability.", *IEEE Trans. Automat. Contr.*, Vol. AC-26, pp. 130- , 1981.

[5] R.V. Patel, "Computation of Matrix Fraction Descriptions of Linear Time-Invariant Systems", *IEEE Trans. Automat. Contr.*, Vol. AC-30, pp. 148-161, 1981.

[6] P. Van Dooren, " The Generalized Eigenstructure Problem in Linear System Theory", *IEEE Trans. Automat. Contr.*, Vol. AC-26, pp. 111-129, 1981.

[7] R.V. Patel, "Computation of Minimal Order State-Space Realization and Observability Indices using Orthogonal Transformations," *Int. J. Contr.*, Vol. 33, pp. 227-246, 1981.

[8] G.S. Miminis and C.C. Paige, "An algorithm for pole assignment of time invariant multi-input systems", *IEEE Conf. Dec. Contr.*, Orlando, FL, pp. 62-67, 1982.

[9] R.V. Patel and P. Misra, "Numerical Algorithms for Eigenvalue Assignment by State Feedback", *Proc. IEEE*, Vol. 72, pp. 1755-1764, 1984.

[10] A. Emami-Naeini and P. Van Dooren, "Computation of Zeros of Linear Multivariable Systems", *Automatica*, Vol. 18, pp 415-430, 1982.

[11] A.J. Laub, "Efficient Multivariable Frequency Response Calculations", *IEEE Trans. Automat. Contr.*, Vol. AC-26, pp. 407-409, 1981.

[12] P. Misra and R.V. Patel, "A Computational Method for Frequency Response of Multivariable Systems", *IEEE Conf. Dec. Contr.*, Ft. Lauderdale, Florida, pp. 1248-1249, 1985.

[13] R.V. Patel and P. Misra, "A Numerical Algorithm for Eigenvalue Assignment by Output Feedback", in *Computational and Combinatorial Methods in Linear System Theory*, C. Byrnes and A. Lindquist, *Eds*.

[14] A.J. Laub, "A Schur Method for Solving Algebraic Riccati Equation", *IEEE Trans. Automat. Contr.*, Vol. AC-24, pp. 913-921, 1979.

[15] G. Zames, "Feedback and Optimal Sensitivity: Model Reference Transformations Multiplicative Seminorms and Approximate Inverses", *IEEE Trans. Automat. Contr.*, Vol. AC-26, pp. 301-320, 1981.

[16] J.W. Helton, "Worst Case Analysis in the Frequency Domain: The H^∞ Approach to Control", *IEEE Trans. Automat. Contr.*, Vol. AC-30, pp. 1154-1266, 1985.

[17] M. Vidyasagar, *Control Systems Synthesis: A Factorization Approach*, MIT Press, Cambridge, 1985.

[18] H.H. Rosenbrock, *Computer-Aided Control Systems Design*, Academic Press, London, 1974.

[19] T. Kailath, *Linear Systems*, Prentice Hall, Englewood Cliffs, N.J., 1980.

[20] R.V. Patel and N. Munro, *Multivariable Systems Theory and Design*, Pergamon Press, Oxford, 1982.

[21] W.M. Wonham, *Linear Multivariable Control: A Geometric Approach, Springer Verlag, N.Y., 1979.*

[22] G.W. Stewart, *Introduction to Matrix Computations*, Academic Press, N.Y., 1973.

[23] J.H. Wilkinson, *An Algebraic Eigenvalue Problem*, Oxford University Press, London, 1965.

[24] G.E. Forsythe, M.A. Malcolm and C.B. Moler, *Computer Methods for Mathematical Computations*, Prentice-Hall, Englewood Cliffs, N.J., 1977.

[25] G.H. Golub and C. VanLoan, *Matrix Computations*, The John Hopkins Press, Baltimore, MD, 1983.

[26] F. Fallside, *Ed.*, *Control System Design by Pole-Zero Assignment*, Academic Press, London, 1977.

[27] R.V. Patel, "On Blocking Zeros in Linear Multivariable Systems", *IEEE Trans. Automat. Contr.*, Vol. AC-31, pp. 239-241, 1971.

[28] R.V. Patel, "On Computing the Invariant Zeros of Multivariable Systems", *Int. J. Contr.*, Vol. 24, pp. 145-146, 1976.

[29] A.J. Laub and B.C Moore, "Calculation of Transmission Zeros using QZ Techniques", *Automatica*, Vol. 14, pp. 557-566, 1978.

[30] B.T. Smith, *et al.*, *Matrix Eigensystems Routines - EISPACK Guide*, Lecture notes in Comp. Sci., Vol. 6, Springer Verlag, N.Y., 1976.

[31] B.S. Garbow, *et al.*, *Matrix Eigensystems Routines - EISPACK Guide Extension*, Lecture notes in Comp. Sci., Vol. 51, Springer Verlag, N.Y., 1977.

[32] J.J. Dongarra, *et al.*, *LINPACK Users' Guide*, Philadelphia, SIAM, P.A., 1979.

[33] C.B. Moler and C.F. Van Loan, "Nineteen Dubious Ways to Compute the Exponential of a Matrix", *SIAM Review*, Vol. 20, pp. 801-837, 1978.

[34] A.J. Laub, "Efficient Multivariable Frequency Response Computations", *IEEE Trans. Automat. Contr.*, Vol. AC-26, pp. 407-408, 1981.

[35] P. Misra and R.V. Patel, "A Determinant Identity with Application in Computing Frequency Response Matrices", to appear in *'Linear Algebra in Signals, Systems and Control'*, B.N. Datta, *Ed.*, SIAM, 1987.

[36] W.A. Wolovich, *Linear Multivariable Systems*, Springer-Verlag, N.Y., 1974.

[37] C.C. Paige and C.F.Van Loan, "A Schur Decomposition for Hamiltonian Matrices", *Linear Alg. and Its Appl.*, Vol. 41, pp. 11-32, 1981.

[38] G.H. Golub, S. Nash and C.F. Van Loan, "A Hessenberg-Schur Method for the Problem $AX+XB+C$", *IEEE Trans. Automat. Contr.*, Vol. AC-24, pp. 909-913, 1979.

[39] I. Horowitz, *Synthesis of Feedback Systems*, Academic Press, N.Y., 1963.

[40] D.H Owens, "Multivariable Root-Loci: An Emerging Design Tool?", *IEE Int. Conf. on Contr. and Appl.*, Warwick, 1981.

[41] I. Postlewaite and A.G.J. MacFarlane, *Complex Variable Methods for Linear Multivariable Feedback Systems*, Taylor and Francis Ltd., London, 1980.

[42] A. Varga and V. Sima, "Numerically Stable Algorithm for Transfer Function Matrix Evaluation", *Int. J. Contr.*, Vol. 33, pp. 1123-1133, 1981

[43] A. Emami-Naeini and P. Van Dooren, "On Computation of Transmission Zeros and Transfer Functions", *IEEE Conf. Dec. Contr.*, Las Vegas, pp. 51-56, 1984.

[44] P. Misra and R.V. Patel, "A Novel Approach for Computing Transfer Function Matrices of Multivariable Systems", *IEEE Int. Symp. Circuits and Syst.*, Philadelphia, pp.296-299, June 4-7, 1987.

[45] P. Misra and R.V. Patel, "Computation of Transfer Function Matrices of Multivariable Systems", to appear in *Automatica*.

[46] C.N. Nett, C.A. Jacobson abd M.J. Balas, "A Connection Between State-Space and Doubly Coprime Fractional Representations", *IEEE Trans. Automat. Contr.*, Vol. AC-29, pp. 831-832, 1984.

[47] L.M. Silverman, "Inversion of Multivariable Linear Systems", *IEEE Trans. Automat. Contr.*, Vol. AC-14, pp. 270-276, 1969.

[48] P. Van Dooren, P. Dewilde and J. Vandewalle, "On the Determination of the Smith-Macmillan Form of a Rational Matrix from its Laurent Expansion", *IEEE Trans. Cir. Syst.*, Vol. CAS-26, pp. 180-189, 1979.

CHAPTER II

NOTATION AND PRELIMINARIES

This chapter presents the notation, several definitions and basic concepts from *linear algebra* and *linear system theory*. These results are later used to develop computational methods for analysis and design of multivariable systems. The topics from linear algebra surface quite naturally when studying linear systems because the information required to describe the systems, usually in terms of high order differential or difference equations, can be expressed compactly in the form of *matrices, linear vector spaces, linear transformations, etc*. It is, therefore, useful to review the relevant topics from linear algebra and to express them in the framework of linear system theory.

The chapter is divided into two main sections. Section 2.2. contains results from *matrix theory* and *numerical analysis* where topics such as stability of algorithms, formation of orthogonal transformations and reduction of matrices into condensed forms are discussed. Section 2.3. is concerned with describing some basic principles and various useful representations of linear multivariable systems.

2.1. NOTATION

Unless stated otherwise, we will use the following notation in this and subsequent chapters:

$\mathbb{R}^n$	real n-dimensional vector space
$\mathbf{0}$	a zero vector of appropriate dimension
e_i	i-th unit vector (a zero vector with only i-th element equal to 1)
$\mathbb{R}^{n \times m}$	the set of real $n \times m$ matrices
0	a zero matrix of appropriate dimension
I_n	an $n \times n$ identity matrix

A^T	transpose of A
$\mathbb{C}^n$	complex n-dimensional vector space
$\mathbb{C}^{n \times m}$	the set of complex $n \times m$ matrices
$\sigma(A)$	the set of eigenvalues of A
A^{-1}	inverse of A
$\mathrm{rank}(A)$	rank of A
$\mathrm{adj}(A)$	adjoint of A
$\det(A)$	determinant of A
$\\|(\cdot)\\|_p$	p-th norm of $(\cdot)$
$fl(\cdot)$	the expression $(\cdot)$ evaluated in floating-point arithmetic

The following abbreviations are used frequently in rest of the thesis:

UHM	Upper Hessenberg matrix
UHF	A single-input system (A, b, C) in upper Hessenberg form
BUHM	Block upper Hessenberg martix
BLHM	Block lower Hessenberg martix
BUHF	A multivariable system (A, B, C) in block upper Hessenberg form
BLHF	A multivariable system (A, B, C) in block lower Hessenberg form
USM	Upper Schur matrix
USF	A multivariable system (A, B, C) in upper Schur form
RSM	Real Schur matrix
RSF	A multivariable system (A, B, C) in real Schur form
c-c	complex-conjugate
EVA	Eigenvalue-assignment
DFM	Decentralized fixed mode.

Throughout the text, upper case letters will denote real as well as complex matrices, vectors that correspond to a column or row of a matrix will be denoted by lower case bold italic letters with a single subscript. The subscript will correspond to the column or the row of the matrix to which the vector corresponds *e.g.*, a_i denotes the i-th column of A. Lower case Greek and roman letters will denote scalars. Elements of a vector will be identified by a single subscript and elements of a matrix by a double subscript *e.g.*, the element in the i-th row and j-th column of matrix A is denoted by $a_{i,j}$ and the i-th element of a vector b is denoted by b_i. All vectors are assumed to be column vectors, the row vectors are denoted with a tranposition sign.

2.2. MATRIX THEORY AND NUMERICAL ANALYSIS

The main purpose of this section is to review the relevant results and not to provide an exhaustive treatment of the vast literature that exists on numerical analys. and matrix theory. The interested reader is referred to excellent texts on matrix theory [1-6] and numerical analysis [7-12]. The topics discussed here will be used extensively in subsequent chapters.

2.2.1. Matrix Theory

This section presents the notation, terminology and definitions from matrix theory that will be used in the subsequent chapters. The following definitions will be used frequently [10]:

Definition 2.1. Norms of Vectors and Matrices A vector norm on $\mathbb{R}^n$ is a function ν $: \in \mathbb{R}^n \rightarrow \mathbb{R}$ that satisfies the following conditions:

1. $x \neq 0 => \nu(x) > 0,$

2. $\nu(ax) = |a| \; \nu(x),$

3. $\nu(x+y) = \nu(x) + \nu(y).$

The most commonly used norms are the 1-, 2-, and ∞- norms defined as

$$\| x \|_1 = \sum_{i=1}^{n} | x_i |$$

$$\| x \|_2 = \left(\sum_{i=1}^{n} | x_i^2 | \right) \equiv (x^T x),$$

and

$$\| x \|_\infty = \max | x_i | : i = 1, \cdots, n$$

The 2-norm is also known as the Euclidean norm of a vector. For matrices, the 1-, 2- and ∞- norms are defined as (for $A \in \mathbb{R}^{n \times m}$):

$$\| A \|_1 = \max \left(\sum_{i=1}^{n} | a_{i,j} | : j = 1, \cdots, m \right)$$

$$\| x \|_2 = \max_{\| x \|_2 = \| y \|_2 = 1} | y^T A x |$$

and

$$\| A \|_\infty = \max \left(\sum_{j=1}^{m} | a_{i,j} | : i = 1, \cdots, n \right).$$

Another commonly used norm, *Frobenius* norm, for matrices is defined as:

$$\| A \|_F = \left(\sum_{i=1}^{n} \sum_{j=1}^{m} a_{i,j}^2 \right)^{\frac{1}{2}}.$$

It can be easily shown that the Frobenius norm satisfies all the properties of a norm [8,10,12].

Definition 2.2. Orthogonal and Unitary Matrices: A square matrix $A \in \mathbb{R}^{n \times n}$ is said to be an *orthogonal* matrix and the columns of A are *orthonormal* if

$$A^{-1} = A^T$$

i.e.,

$$A^T A = I_n$$

The complex counterpart of an orthogonal matrix satisfying $A^{-1} = A^H$ is said to be *unitary.*

Definition 2.3. Householder Transformations and Plane Rotations: These transformations are used for annihilating the elements of a vector or a matrix in certain order *e.g.* in QR decomposition of a matrix defined later in this section. The matrices representing these transformations are orthogonal. Typically, a *Householder transformation* is a matrix of the form—

$$I - uu^T \tag{2.2.1a}$$

where $u^T u = 1$.

Consider $x \in \mathbb{R}^n$ and let $\sigma = \pm \| x \|_2$ such that $x \neq -\sigma e_1$ where $e_1 = [1, 0, \cdots, 0]$. Let

$$u = x + \sigma e_1 \tag{2.2.1b}$$

and

$$\pi = \| u \|_2^2. \tag{2.2.1c}$$

Then, $U = I_n - \pi^{-1} u u^T$ is a Householder transformation matrix and $Ux = -\sigma e_1$.

A *plane rotation* in the (i, j) plane is a matrix of the form

$$P_{ij} = \begin{array}{c} \\ \\ \\ i \\ \\ j \\ \\ \\ \\ \end{array} \left[\begin{array}{cccccccc} 1 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & & & \vdots & & \vdots & & \vdots \\ 0 & 0 & \cdots & \alpha & \cdots & \beta & \cdots & 0 \\ \vdots & & & \vdots & & \vdots & & \vdots \\ 0 & 0 & \cdots & -\beta & \cdots & \alpha & \cdots & 0 \\ \vdots & & & \vdots & & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \\ & & & i & & j & & \end{array} \right], \tag{2.2.2}$$

where, $\alpha^2 + \beta^2 = 1$. A plane rotation may be used to annihilate the elements of a vector. However, it is different from the Householder transformations, because while the Householder transformation operates on the entire vector at one time, a plane rotation only uses the i-th and j-th elements of a vector at a time.

The transformations described above are used frequently in numerical linear algebra because they are orthogonal and have very desirable numerical properties. Further accounts of their properties and applications can be found in [7-12].

Definition 2.4. Elementary Transformations: An *elementary transformation* is represented by an elementary lower triangular matrix of order n, and index k of the form

$$M = I_n - m e_k^T \tag{2.2.3a}$$

where,

$$m e_i^T = 0 \quad i = 1,2 \cdots, k \tag{2.2.3b}$$

More specifically,

$$M = \begin{bmatrix} 1 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & \cdots & 0 \\ \cdot & \cdot & & \cdot & & \cdot \\ \cdot & \cdot & & \cdot & & \cdot \\ 0 & 0 & \cdots & 1 & \cdots & 0 \\ 0 & 0 & \cdots & m_{k+1,k} & \cdots & 0 \\ \cdot & \cdot & & \cdot & & \cdot \\ \cdot & \cdot & & \cdot & & \cdot \\ 0 & 0 & \cdots & m_{n,k} & \cdots & 1 \end{bmatrix} \tag{2.2.3c}$$

Elementary transformations can also be used for annihilating the elements of a vector or a matrix *e.g.*, in Gaussian elimination or LU decomposition of a square matrix discussed later in this section.

One frequently needs to perform transformations on a matrix to represent it in a more "convenient" form. Definitions of some of the more useful transformations and decompositions are listed next.

Definition 2.5. LU Decomposition: Any non-singular matrix $A \in \mathbb{R}^{n \times n}$ can be represented as a product of a *unit* lower triangular matrix L (1's along the diagonal) and an upper triangular matrix U such that

$$A = LU. \tag{2.2.4a}$$

Further,

$$det(A) = det(U) = \prod_{i=1}^{n} u_{ii} \tag{2.2.4b}$$

The LU decomposition together with solutions of two sets of linear equations form a reliable way of computing the *inverse* of a matrix [8-12].

Definition 2.6. QR Decomposition: Let $A \in \mathbb{R}^{n \times m}$ with $n > m$, have linearly independent columns. The matrix A can be written uniquely in the form $A = QR$, where Q is an orthogonal matrix and R is an upper triangular matrix with positive elements along the diagonal. Note that if $n < m$, R is an upper *trapezoidal* matrix.

The QR decomposition is useful in row and/or column compression of a matrix. It also forms the basis of efficient algorithms for computation of eigenvalues of matrices [7-12]. The QR algorithm produces a sequence of matrices A_i, $i = 0,1,2, \cdots$ as follows. At the i-th step, given A_i, a λ_i scalar called *shift of origin*, is determined from the elements of A_i. The matrix $A_i - \lambda_i I$ is then factored as

$$A_i - \lambda_i I = Q_i R_i \tag{2.2.5a}$$

where Q_i is unitary and R_i is upper triangular. This factorization always exists provided $A_i - \lambda_i I$ is non-singular [10]. Finally, A_{i+1} is computed as

$$A_{i+1} = Q_i R_i + \lambda_i I \tag{2.2.5a}$$

With a proper choice of λ_i, the subdiagonal elements of A_i quadratically converge to zero and the diagonal terms towards the eigenvalues of matrix A.

Definition 2.7. Singular Value Decomposition: For a matrix $A \in \mathbb{R}^{n \times m}$, the singular value decomposition is obtained by finding two orthogonal matrices U and V such the

$$UAV^T = \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix}.$$

where, $\Sigma = \text{diag}(\sigma_1, \sigma_2, \cdots, \sigma_r)$, the terms σ_i are in descending order and $\sigma_r > 0$ [1-6]. The scalars σ_i, $i = 1,2, \cdots, r$ are called the singular values of A and r denotes the rank of A [10].

The LU and QR decompositions described above are obtained by transformations on either rows or columns only. Such transformations, while preserving the rank, determinant, etc., do not preserve the eigenvalues. All of the above properties, including the eigenvalues are preserved under *similarity transformations* of a matrix *B to A*, *characterized by*

$$A = T^{-1}BT,$$

where T is a square invertible transformation matrix.

If the transformation matrices used are orthogonal or unitary, it can be shown [8,10] that the resulting transformed matrix A is exact for a "slightly" perturbed original matrix B. However, if a general similarity transformation involving the inverse of a matrix is employed, nothing specific can be said about the "exactness" of the corresponding solutions. The terms "slightly perturbed" and "exactness" will be discussed in detail in the next section. In the next few paragraphs, only those reductions of a square matrix have been discussed which can be obtained using orthogonal or unitary similarity transformations.

1. Hessenberg Matrix: Any square matrix $A \in \mathbb{R}^{n \times n}$ can be reduced by means of an orthogonal similarity transformation $T \in \mathbb{R}^{n \times n}$ to an *Upper Hessenberg Matrix* (UHM) F *i.e.*,

$$F = T^T A T = \begin{bmatrix} f_{11} & f_{12} & \cdot & \cdot & f_{1,n-1} & f_{1,n} \\ f_{21} & f_{22} & \cdot & \cdot & f_{2,n-1} & f_{2,n} \\ 0 & f_{32} & \cdot & \cdot & f_{3,n-1} & f_{3,n} \\ \cdot & \cdot & & & \cdot & \cdot \\ \cdot & \cdot & & & \cdot & \cdot \\ 0 & 0 & \cdot & \cdot & f_{n,n-1} & f_{n,n} \end{bmatrix}. \tag{2.2.6}$$

The matrix F is said to be an *unreduced* UHM if $f_{i+1,i} \neq 0$, $i = 1,2, \cdots, n-1$. Similar results can be stated about the reduction to a *lower Hessenberg matrix* (LHM) [8-12]. Note that in above reduction, due to finite precision arithmetic, we cannot expect the elements of the sub-diagonal to be exactly zero. It can, however, be shown that computed F is orthogonally similar to $A+E$, where $\| E \|_F \leq \gamma n^2 \epsilon \| A \|_F$ where n is the order of the matrix and γ is a constant of order unity. In practice ϵ is taken to be of the order of 10^{-t}, where t is the number of significant digits used in calculations.

2. Block Hessenberg Matrix: If the elements of the matrix F in (2.2.6) are themselves matrices of appropriate dimension, then the matrix is said to be a *Block Upper Hessenberg Matrix* (BUHM). Further, if each of the sub-diagonal block sub-matrices has full rank; then the matrix is said to be an *unreduced* BUHM. The reduction to BUHM can be achieved using orthogonal similarity transformations. Similar results can be stated for reduction of a matrix to a block lower Hessenberg matrix (BLHM).

3. Real Schur Matrix: Any square matrix $A \in \mathbb{R}^{n \times n}$ can be reduced by means of orthogonal similarity transformations to a *Real Schur Matrix* (RSM) F such that $F = T^T AT$ is a *quasi-upper triangular* matrix with only scalars and 2×2 blocks along the (block) diagonal [1-6]. Each scalar on the diagonal corresponds to a real eigenvalue of A and each 2×2 block corresponds to a complex-conjugate pair of eigenvalues of A.

4. Upper Schur Matrix: *Upper Schur Matrix* (USM) is the complex counterpart of an RSM. The transformations required for reduction to USM are unitary. An USM may have complex elements as it has its eigenvalues (real as well as complex-conjugate pairs) along the diagonal.

For both real and upper Schur matrices, it is always possible to rearrange the order in which the eigenvalues appear along the diagonal. Numerically efficient and stable algorithms as well as software exist for the above reductions [13-14].

For various condensed forms described above, efficient FORTRAN codes may be found in [15-17]. It should be pointed out that there are several other useful condensed forms to which a matrix can be reduced *e.g., diagonal form, Jordan canonical form, companion form, etc.*, but reduction to these forms is numerically unreliable and therefore, their use in developing algorithms for analysis and design of multivariable systems is not recommended. The interested reader can find a detailed exposition on the subject in [1-6].

2.2.2. Stability and Conditioning

The algebraic operations represented by the algorithms in subsequent chapters are executed on digital computers. Because of the limited precision arithmetic used on the computers, the basic operations of addition and multiplication invariably incur rounding errors. When an algorithm involving such operations is implemented on a computer, the result *can* be far from the expected one. The errors incurred usually arise from the following two sources:

1. Use of a "bad" algorithm to solve the problem

2. Inherent "ill-posedness" of the problem itself.

The terms "bad" and "ill-posedness" will be defined and explained in numerical analysis terminology in the next few paragraphs. While one has some control over the type of algorithms used to solve the problem, the second source of error, when it exists, cannot be removed. This is illustrated by the examples given below, where for the sake of illustration, it is assumed that all operations are carried out on a computer accurate to 4 significant places.

Example 2.1 : Find the determinant of the matrix

$$A = \begin{bmatrix} 4.000 & 8.780 \\ 2.000 & 4.391 \end{bmatrix}. \qquad (2.2.7)$$

There are several ways in which one could evaluate the determinant. One expression for the determinant of a matrix $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ is given by

$$\det(A) = a_{11} \times a_{22} - a_{12} \times a_{21} \qquad (2.2.8)$$

therefore,

$$\det(A) = 4.000 \times 4.391 - 2.000 \times 8.780.$$

The above computation on implementation on a 4-digit computer is executed as

$$\begin{aligned} \det(A) &= fl(4.000 \times 4.391) - fl(2.000 \times 8.780) \\ &= fl(17.564) - fl(17.56) \\ &= 17.56 - 17.56 \\ &= 0.000 \end{aligned}$$

which is wrong because the actual answer can be easily seen to be 0.004.

A second method for evaluating the determinant is by carrying out an LU decomposition of A as given in (2.2.4), giving us

$$\begin{bmatrix} 4.000 & 8.780 \\ 2.000 & 4.391 \end{bmatrix} = \begin{bmatrix} 1.000 & 0.000 \\ 0.500 & 1.000 \end{bmatrix} \begin{bmatrix} 4.000 & 8.780 \\ 0.000 & 0.001 \end{bmatrix} \qquad (2.2.9)$$

and

$$det(A) = det(U) = fl(4.000 \times 0.001) \qquad (2.2.10)$$
$$= 0.004$$

thereby, giving the correct answer.

Example 2.2 : Find the determinant of the matrix

$$A = \begin{bmatrix} 4.000 & 5.000 \\ 3.511 & 4.391 \end{bmatrix}. \qquad (2.2.11)$$

The actual determinant is 0.009. On applying the algorithm in (2.2.8),

$$\begin{aligned} det(A) &= fl(4.000 \times 4.391) - fl(3.511 \times 5.000) \\ &= fl(17.564) - fl(17.555) = fl(17.56 - 17.56) \\ &= 0.0 \end{aligned}$$

Again, applying algorithm in (2.2.4), we have

$$\begin{bmatrix} 4.000 & 5.000 \\ 3.511 & 4.391 \end{bmatrix} = \begin{bmatrix} 1.000 & 0.000 \\ fl\left(\frac{3.511}{4.000}\right). & 1.000 \end{bmatrix} \begin{bmatrix} 4.000 & 5.000 \\ 0.000 & fl\left\{ fl\left(\frac{3.511}{4.000}\right) \times 5.000 \right\} - 4.391 \end{bmatrix}$$
$$= \begin{bmatrix} 1.000 & 0.000 \\ 0.8778 & 1.000 \end{bmatrix} \begin{bmatrix} 4.000 & 5.000 \\ 0.000 & -0.002 \end{bmatrix}$$

Therefore,

$$det(A) = det(U) = fl(-0.002 \times 4.000)$$
$$= -0.008.$$

The answer obtained by using either of the algorithms is far from correct.

On first sight, the inaccuracy incurred can be attributed to the rounding errors in the implementation. But a closer examination reveals that despite the underlying cause being rounding errors, the actual reason for getting wrong answers in the two cases are quite different. The failure to get an accurate answer in Example 2.1 was caused by using a *bad* algorithm because the use of alternate algorithm in (2.2.4) gave the correct answer. Whereas, in Example 2.2, the error was not because of the type of the algorithm used, rather the data itself does not permit the algorithms to obtain the accurate answer. A problem whose data exhibits such behaviour is said to be *ill-conditioned* and one cannot hope to obtain correct answer irrespective of the algorithm used. Note that in both of the above examples, the matrices are ill-conditioned with respect to inversion, the same cannot be said about computation of their determinants.

In Example 2.1, it was noticed that the accuracy of the answer is dependent on the type of algorithm employed. If an algorithm yields a solution that is the *exact* solution of a slightly perturbed problem, the algorithm is said to be *numerically stable.*

The concepts of stability and conditioning can be defined more precisely as follows [10]: Assume that we are given a mathematically defined function f that, on operation on some data $d \in \mathbb{D}$ gives an evaluated solution $f(d)$ where $\mathbb{D}$ is some set of data. The numerical problem is then to compute an approximation to $f(d)$ given d. As was shown in the examples above, the nature of the function f limits the accuracy of the evaluated solutions. Suppose that only a "close" approximation $\overline{d}$ of the data d is known to us. If the evaluated solutions $f(d)$ and $f(\overline{d})$ differ greatly, then the problem is said to be *ill-conditioned.*

Formulating an algorithm for solving the given mathematical problem associated with f is equivalent to defining a new mathematical function $\overline{f}$ that, given data d, approximates a solution $\overline{f}(d)$. It is not reasonable to expect that $\overline{f}$ can give a more accurate solution than the data would permit, however, the following is expected of $\overline{f}$: For any $d \in \mathbb{D}$, there is a nearby $\overline{d} \in \mathbb{D}$ such that $f(\overline{d})$ is near $\overline{f}(d)$. Equivalently, it is expected that the selected algorithm yields a solution that is near the exact solution of a slightly perturbed problem. An algorithm with this property is called *backward stable*. It is easy to see that while condition is a property of a problem, stability is a property of an algorithm.

It becomes increasingly important to use numerically stable algorithms as the "size" of the problem increases. The effect of using a stable algorithm is to minimize the errors incurred in implementation. Therefore, for *well-conditioned* problems, the conclusions drawn from using numerically stable analysis and design techniques would be considerably more reliable.

2.3. Reduction of a System to Condensed Forms

A large number of methods used for analysis and design of systems are based on availability of a linear model of the system under consideration. Such models are usually obtained by linearizing the given system about certain operating points. It may happen (as is usually the case in practice) that the operating point of the system is not fixed, and therefore, several models at different operating points may have to be considered for a satisfactory performance of the system over the whole *operating region*. Unless one can represent these models in compact forms that are especially suited for analysis and design, the actual analysis and/or design of these linear models using numerical techniques at all operating points can become a formidable task. Most of the techniques developed for analysis and design in the subsequent sections are based on *state-space* representation of linear multivariable systems.

Representation of a system (A,B,C,E) in condensed forms enables us to apply several well-known numerical linear algebra techniques for analysis and design of systems. In this section, several frequently used condensed forms are described. The section also serves the purpose of familiarizing the reader with the format in which the algorithms will be presented in the subsequent chapters. The algorithms in this and later chapters, are presented keeping *readability* in mind and not to describe the details of coding at each step. Thus a matrix product is written as $A = BC$ instead of *e.g.*,

$$a_{ij} = \sum_{k=1}^{p} b_{ik}\, c_{kj}\,, \quad i=1,n\,, \quad j=1,m.$$

Comments are used to clarify the steps in algorithm to enable the reader to understand the sequence and significance of each step. The reason for keeping the statements and steps of the algorithms rather general is to permit the software specialists to code the algorithm for best performance (both in terms of efficiency and stability).

In transforming the system (A,B,C,E), using similarity transformations, the matrix E is left unchanged and hence instead of considering the system (A,B,C,E), we simply consider the triple (A,B,C) [18,19]. The condensed forms of the system triple (A,B,C) used in subsequent chapters are the following:

1. **Upper Schur Form:** Any system (A,B,C) can be transformed by means of unitary similarity transformations T to (F,G,H) in upper Schur form (USF) such that,

$$(F,G,H) = (T^H A T, T^H B, CT)$$

where the matrix F is an USM, and the matrices G and H have no specific structure. The eigenvalues of A can be made to appear along the diagonal of F in any desired order [13-14].

2. **Real Schur Form:** The above definition can be readily modified to describe a system in its *Real Schur Form* (RSF). The transformation matrix T in this case will be orthogonal and the matrix F, is consequently reduced to a real Schur matrix. The (1×1) and (2×2) blocks along the diagonal can be made to occur in any desired order.

3. **Block Upper Hessenberg Form:** A system triple (A,B,C) can be reduced to a *Block Upper Hessenberg Form* (BUHF) $(F,G,H.)$, by means of an orthogonal state coordinate transformation matrix T [20-22], such that

$$F = T^T A T = \begin{bmatrix} F_{11} & F_{11} & \cdot & \cdot & F_{1,k} & F_{1,k+1} \\ F_{21} & F_{21} & \cdot & \cdot & F_{2,k} & F_{2,k+1} \\ 0 & F_{32} & \cdot & \cdot & F_{3,k} & F_{3,k+1} \\ \cdot & \cdot & & & \cdot & \cdot \\ \cdot & \cdot & & & \cdot & \cdot \\ 0 & 0 & . & . & F_{k,k} & F_{k,k+1} \\ 0 & 0 & . & . & 0 & F_{k+1,k+1} \end{bmatrix} \tag{2.3.1a}$$

$$G = T^T B = \begin{bmatrix} G_1^T & 0 & \cdots & 0 & 0 \end{bmatrix}^T \tag{2.3.1b}$$

$$H = CT = \begin{bmatrix} H_1 & H_2 & \cdots & H_k & H_{k+1} \end{bmatrix} \tag{2.3.1c}$$

where $F_{ij} \in \mathbb{R}^{l_{i-1}\times l_{j-1}}$, $H_j \in \mathbb{R}^{p\times l_{j-1}}$ and $G_1 \in \mathbb{R}^{l_0\times m}$. The integers l_i, $i=0,\cdots,k$ are defined as $l_0=$ rank (B), $l_i=$ rank $F_{i+1,i}$, $i=1,\cdots,k-1$ and $\sum_{i=0}^{k-1} l_i=\mu$ where μ is the dimension of the *controllable subspace*. Further, $\mu=n$ if and only if the system is controllable [21-23].

A similar result can be stated for reducing the triple to a *Block Lower Hessenberg Form* (BLHF). An algorithm for reduction to BLHF appears in [20]. Reduction to BUHF can be formalized along the same lines.

4. Upper Hessenberg Form: This condensed form is the single-input analog of BUHF. Given a single-input, multi-output system (A, b, C), there exists an orthogonal state coordinate transformation T such that $(F, g, H) = (T^T AT, T^T b, CT)$ where F is an upper Hessenberg matrix, $g = [g_1 \; 0 \; \cdots \; 0]^T$ and H has no specific structure [24].

The matrix F is an unreduced upper Hessenberg matrix if and only if (A, b) is a *controllable pair*. Further, if F is not unreduced, then the system can be partitioned as

$$F = \begin{bmatrix} F_{11} & F_{12} \\ 0 & F_{22} \end{bmatrix}, \quad g = \begin{bmatrix} g_1 \\ 0 \end{bmatrix} \qquad (2.3.2a)$$

$$\text{and } H = \begin{bmatrix} H_1 & H_2 \end{bmatrix}. \qquad (2.3.2b)$$

From where, it is easy to see that the pair (F_{11}, g_1) is controllable. Similar results can be stated for reducing a multi-input, single-output system to a *lower Hessenberg form*.

A slight variation of the UHF is given by $(F, G, h) = (T^T AT, T^T B, c^T)$ where F is an upper Hessenberg matrix, G has no specific structure and $h = [0 \; \cdots \; 0 \; h_n]$. The matrix F is unreduced if and only if (A, c) is an *observable pair*, failing which, the system can be partitioned as

$$F = \begin{bmatrix} F_{11} & F_{12} \\ 0 & F_{22} \end{bmatrix}, \quad g = \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} \qquad (2.3.3b)$$

$$\text{and } H = \begin{bmatrix} 0 & h_2 \end{bmatrix}. \qquad (2.3.3b)$$

Here, (F_{22}, h_2) forms an observable pair. This form finds applications in evaluating *frequency response matrices* of multivariable systems. Note that the matrices F_{11} and F_{22} are both square matrices of appropriate diemnsions. Further details appear in Chapter VI. An algorithm for reducing a single-input pair (A, b) to UHF is given next.

Algorithm 2.1: (Reduction of single input pair (A, b) to UHF)

Step I: (Transform b)

1) Set $i=0$ and perform Householder transformations or plane rotations (denoted by an orthogonal matrix T_0^T) on b such that

$$g = T_o^T b_0 = \left[g_1^{(0)} \ \ 0 \cdots 0 \right]^T$$

2) Set $F_0 = T_0^T A T_0$ and $T = T_0$.

Step II: (Reduce A to an UHM)

1) If $i=n-2$, go to *Step III* (1), else partition F_i as

$$F_i = \begin{bmatrix} f_{11}^{(i)} & f_{12}^{(i)} & \cdots & f_{1,i+1}^{(i)} & f_{1,i+2}^{(i)T} \\ f_{21}^{(i)} & f_{22}^{(i)} & \cdots & f_{2,i+1}^{(i)} & f_{2,i+2}^{(i)T} \\ 0 & f_{32}^{(i)} & \cdots & f_{3,i+1}^{(i)} & f_{3,i+2}^{(i)T} \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & f_{i+1,i+1}^{(i)} & f_{i+1,i+2}^{(i)T} \\ 0 & 0 & \cdots & f_{i+2,i+1}^{(i)} & F_{i+2,i+2}^{(i)} \end{bmatrix}$$

where $F_{i+2,i+2}^{(i)} \in \mathbb{R}^{n-i-1 \times n-i-1}$.

Set $i=i+1$ and $b_i = f_{i+1,i}^{(i-1)}$.

2) If $b_i = 0$ go to *Step III* (2), else perform Householder transformations or plane rotations T_i^T on b_i such that

$$T_i^T b_i = \left[a_{i+1,i}^{(i)} \ \ 0 \cdots 0 \right]^T$$

3) Set

$$F_i = \begin{bmatrix} I_i & 0 \\ 0 & T_i^T \end{bmatrix} F_{i-1} \begin{bmatrix} I_i & 0 \\ 0 & T_i \end{bmatrix}$$

and

$$T = T \begin{bmatrix} I_i & 0 \\ 0 & T_i \end{bmatrix}$$

where I_i is the $i \times i$ identity matrix.

Go to *Step II* (1).

Step III: (Exit)

1) If $f_{i+2,i+1}=0$, set $\mu=n-1$, else, set $\mu=n$;

   go to *Step III*(3)

2) Set $\mu=i$.

3) Set $F=F_i$, $g=g_0$ and STOP.

At the end of Algorithm 2.1, the pair (F,g) is in UHF and μ is the dimension of the controllable subsystem.

For most of the development in subsequent sections, the mathematical preliminaries presented in this section should suffice. However, where necessary, we will introduce further mathematical or system theory principles.

2.4. REFERENCES

[1] C.C. MacDuffee, *The Theory of Matrices*, Chelsea Publishing Co., N.Y., 1950.

[2] F.R. Gantmacher, *Theory of Matrices*, Vol. I and II, Chelsea Publishing Co., N.Y., 1959.

[3] R.E. Bellman, *Matrix Analysis*, 2nd ed., McGraw-Hill, N.Y., 1968.

[4] S. Barnett, *Matrices in Control Theory*, Van Nostrand Reinhold, N.Y., 1972.

[5] G. Strang, *Linear Algebra and Its Applications*, Academic Press, N.Y., 1978.

[6] B. Noble, *Applied Linear Algebra*, 2nd ed., Prentice-Hall, Englewood Cliffs, N.J., 1977.

[7] A.S. Householder, *The Theory of Matrices in Numerical Analysis*, Ginn-Blaisedell, Waltham, Mass., 1964.

[8] J.H. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford University Press, London, 1965.

[9] J.H. Wilkinson and C. Reinsch, Eds., *Handbook for Automatic Computations*, Springer-Verlag, N.Y., 1971.

[10] G.W. Stewart, *Introduction to Matrix Computations*, Academic Press, N.Y., 1973.

[11] G.E. Forsythe, M.A. Malcolm and C.B. Moler, *Computer Methods for Mathematical Computations*, Prentice-Hall, Englewood Cliffs, N.J., 1977.

[12] G.H. Golub and C. VanLoan, *Matrix Computations*, The John Hopkins Press, Baltimore, MD, 1983.

[13] G.W. Stewart, "HQR3 and EXCHNG: Fortran Subroutines for Calculating and Ordering the Eigenvalues of a Real Upper Hessenberg Matrix", *ACM Trans. Math. Software*, Vol. 2, pp. 275-280, 1976.

[14] P. Misra and R.V. Patel, "A Simplified Algorithm for Rearranging the Eigenvalues of an Upper Hessenberg Matrix", to be presented at *Int. Conf. on Linear Algebra and Applications*, Sept. 28-30, Spain, 1987.

[15] B.T. Smith, *et al.*, *Matrix Eigensystems Routines - EISPACK Guide*, Lecture notes in Comp. Sci., Vol. 51, Springer Verlag, N.Y., 1976.

[16] B.S. Garbow, *et al.*, *Matrix Eigensystems Routines - EISPACK Guide Extension*, Lecture notes in Comp. Sci., Vol. 51, Springer Verlag, N.Y., 1977.

[17] J.J. Dongarra *et al.*, *LINPACK Users' Guide*, SIAM, Philadelphia, 1979.

[18] T. Kailath, *Linear Systems*, Prentice-Hall, Englewood Cliffs, N.J., 1980.

[19] R.V. Patel and N. Munro, *Multivariable System Theory and Design*, Pergamon Press, Oxford, 1982.

[20] R.V. Patel, "Computation of Minimal-Order State-Space Realization and Observability Indices using Orthogonal Transformations", *Int. J. Contr.*, Vol. 33, pp. 227-246, 1981.

[21] C.C. Paige, "Properties of Numerical Algorithms Related to Computing Controllability", *IEEE Trans. Automat. Contr.*, Vol. AC-26, pp. 130-138, 1981.

[22] P. Van Dooren, "The Generalized Eigenstructure Problem in Linear System Theory", *IEEE Trans. Automat. Contr.*, Vol. AC-26, pp. 111-129, 1981.

[23] R.V. Patel, "Computation of Matrix Fraction Descriptions of Linear Time-Invariant Systems", *IEEE Trans. Automat. Contr.*, Vol. AC-26, pp. 148-161, 1981.

[24] R.V. Patel and P. Misra, "Numerical Algorithms for Eigenvalue Assignment by State Feedback", *Proc. IEEE*, Vol. 72, pp. 1755-1764, 1984.

CHAPTER III

EIGENVALUE ASSIGNMENT BY STATE FEEDBACK

In this chapter, numerical algorithms for solving the eigenvalue assignment (EVA) problem for multi-input systems by means of state feedback are considered. The outline of the chapter is as follows: In Section 3.1, the problem of EVA by state feedback is stated and a survey of the existing computational methods for solving the problem is given. Section 3.2 presents the theoretical basis for the proposed algorithms. An algorithm for EVA in single-input systems is presented in Section 3.3 and the results are extended to multi-input systems in Section 3.4. Section 3.5 discusses some variations on the multi-input EVA problem together with treatment of several special cases that may arise in practice, and then proposes various modifications to the algorithms to deal with these special cases. The numerical performance of the algorithms are illustrated by means of numerical examples in Section 3.6. Finally in Section 3.7, the results presented in this chapter and their possible extension to more specific applications are discussed briefly.

3.1. STATEMENT OF THE PROBLEM

Consider a linear time-invariant multi-input system described by its state equation

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{3.1.1}$$

where $x(t) \in \mathbb{R}^n$ and $u(t) \in \mathbb{R}^m$. Assume that the pair (A, B) is controllable. The problem that we consider is to find an $m \times n$ constant gain matrix K such that, under the feedback law,

$$v(t) = u(t) - Kx(t) \tag{3.1.2}$$

the resulting closed-loop state matrix

$$A_{cl} = A - BK \tag{3.1.3}$$

has all its eigenvalues at desired locations in the complex plane (symmetric about the real axis). This problem has been investigated by numerous researchers and several algorithms exist for solving the problem, *e.g.* , see [1-14]. However, with the exception of a few recent ones [7-14], not much attention has been paid to the numerical properties and performance of these algorithms. The conceptual simplicity of the eigenvalue assignment (EVA) problem tends to hide the potential numerical difficulties that can arise when using many of the well-known algorithms. Some techniques (e.g. [1-4]) require the reduction of a given state space system to a canonical form. Such reductions are a potential source of numerical instability. Some other algorithms such as the one in [5] require the system to be in transfer function form and uses polynomial arithmetic which *can* cause numerical difficulties.

The algorithms presented in [7-14] have attempted to address the numerical issues involved in the EVA problem. In [7], an algorithm has been presented for carrying out EVA in single-input systems. The algorithm is based on the well-known QR algorithm [15-18] and uses only numerically stable orthogonal transformations. The multi-input case is treated in [8-14]. The algorithms in [8,11] reduce the system to a block upper Hessenberg form by means of orthogonal state coordinate transformations. However, the methods in [8,11] are not straightforward extensions of the method in [7] and in fact it can be shown that they can lead to floating point overflows or underflows. The algorithms in [13-14] are based on the reduction of the system to a RSF by means of orthogonal state coordinate transformations. If the eigenvalue problem of the state matrix of open-loop system is ill-conditioned [15-16], the RSF (and hence the computed eigenvalues) obtained can be extremely inaccurate: If the feedback gains are computed using the inaccurate values of the open-loop eigenvalues, then on applying the feedback, the closed-loop poles *can* be far from the desired ones. This is, therefore a weak point of this and other algorithms that require knowledge of the open-loop eigenvalues.

Numerically, the EVA problem can be treated as the converse of the *algebraic eigenvalue problem*; in the latter, the problem is to find the eigenvalues of a matrix while in the former, the problem is to modify the state matrix of a given system by means of state feedback such that the resulting closed-loop system has a desired set of eigenvalues. Approaching the EVA problem from this point of view makes it significantly easier for us to take into account the numerical issues involved since the algebraic eigenvalue problem has been treated extensively in the numerical analysis literature. The proposed approach is based on the numerically stable QR algorithm [17,18] for finding the eigenvalues of a matrix.

3.2. THEORETICAL DEVELOPMENT FOR SINGLE-INPUT EVA PROBLEM

For simplicity of presentation, we give below some theoretical considerations for single-input systems. These considerations will also apply to multi-input systems as will be seen in later sections. We shall assume without loss of generality that the system pair (A, b) is in UHF. Further, it is assumed that (A, b) is a controllable pair.

It is well known that the eigenvalues of a system can be assigned at any desired locations in the complex plane, subject to complex-conjugate $(c-c)$ pairing, by means of state feedback *if and only if* the system is controllable [2]. Note that if the system is not controllable, the eigenvalues corresponding to the uncontrollable modes of the system cannot be altered. Hence the assumption that (A, b) is a controllable pair is not a restriction on the proposed method.

3.2.1. Explicitly shifted EVA

Consider the controllable single-input system

$$\dot{x}(t) = Ax(t) + bu(t) \tag{3.2.1}$$

where $x(t) \in \mathbb{R}^n$ and $u(t) \in \mathbb{R}$. The problem that we consider in this section is to find a feedback vector $k \in \mathbb{R}^n$ such that the closed-loop state matrix $A_{cl} = A - bk^T$ has n eigenvalues at desired locations. Let $\Lambda = \left\{\lambda_1, \cdots, \lambda_n\right\}$ denote the set of desired locations where the closed-loop eigenvalues are to be assigned and assume that at the start of the l-th step ($l < n$), we have the following structure:

$$U_{l-1}^T \cdots U_0^T \left(A - b \sum_{i=1}^{l-1} k_i^T \right) U_0 \cdots U_{l-1} = \begin{bmatrix} A_{l-1} & * \\ 0 & A_l \end{bmatrix}. \tag{3.2.2}$$

Since, by assumption, the system is controllable, $A_l \in \mathbb{R}^{(n-l+1 \times n-l+1)}$ is an unreduced upper Hessenberg matrix, A_{l-1} is an upper Schur matrix with $\lambda_1 \cdots \lambda_{l-1}$, the assigned eigenvalues being the elements along its diagonal, $U_0 = I_n$ (the $n \times n$ identity matrix), $\sum_{i=1}^{l-1} k_i^T$ is the state feedback vector and

$$U_{l-1}^T \cdots U_0^T b = \left[b_{l-1}^T \quad b_l^T \right]^T \tag{3.2.3}$$

with b_l $(\in \mathbb{R}^{n-l+1}) = \left[b_{l,1}\, 0 \cdots 0 \right]^T$.

In the l-th step, we calculate k_l^T in order to assign an eigenvalue at λ_l. To accomplish this, we first transform the matrix $A_l - \lambda_l I_{n-l+1}$ to an upper triangular form by means of orthogonal plane rotations

$$\left(A_l - \lambda_l I_{n-l+1} \right) P_{n-1,n} \cdots P_{n-l+1,n-l} \triangleq R_l . \tag{3.2.4}$$

In the trivial case, when λ_l is also an eigenvalue of the open-loop system, using the algorithm in [19-20], we can rearrange the eigenvalues of the state matrix such that λ_l appears along the diagonal. In this case, the matrix R_l will be a singular matrix. However, in the case when λ_l is not an eigenvalue of the open-loop system, since A_l is an unreduced upper Hessenberg matrix, the plane rotations above are non-trivial and the diagonal elements of R_l have *non-zero* values.

Denote the product $P_{n-1,n} \cdots P_{n-l+1,n-l}$ by $\bar{Q}_l$, and consider the matrix $A_l - \lambda_l I_{n-l+1} - b_l k_l^T$ where k_l has only its l-th element non-zero. Apply the similarity transformation $\bar{Q}_l$ to the closed-loop system,

$$\bar{Q}_l^T \left(A_l - \lambda_l I_{n-l+1} - b_l k_l^T \right) \bar{Q}_l = \bar{Q}_l^T \left(R_l - b_l k_l^T \bar{Q} \right). \tag{3.2.5}$$

where $k_l^T \bar{Q} = \left[k_1, 0, \cdots, 0 \right]$. Due to the structures of R_l and b_l, the matrix R_l -

$b_l k_l^T \tilde{Q}$ retains the upper triangular structure and its (1,1) element is given by

$$r^l_{1,1} - b_{l,1} k_1 \qquad (3.2.6).$$

where $r^l_{1,1}$ is the (1,1)-th element of R_l and $b_{l,1}$ is the first element of b_l. We choose k_1 to eliminate the (1,1) element of $R_l - b_l k_l \tilde{Q}$. Since by assumption of controllability, $b_{l,1}$ is non-zero, this can be done by selecting

$$k_1 = \frac{r^l_{1,1}}{b_{l,1}}. \qquad (3.2.7)$$

Then, the first column of $\tilde{Q}_l^T(R_l - b_l k_l^T \tilde{Q})$ and therefore, the first column of the expression on the left hand side of (3.2.5) are zero vectors. Also,

$$\tilde{Q}_l^T\left(A_l - \lambda_l I_{n-l+1} - b_l k_l^T\right)\tilde{Q}_l + \lambda_l I_{n-l+1} = \tilde{Q}_l^T\left(A_l - b_l k_l^T\right)\tilde{Q}_l \qquad (3.2.8)$$

where the right hand side has the following structure:

$$\tilde{Q}_l^T\left(A_l - b_l k_l^T\right)\tilde{Q}_l = \begin{bmatrix}\lambda_l & *^T \\ 0 & A_{l+1}\end{bmatrix} \qquad (3.2.9)$$

with an eigenvalue at λ_l. To verify that λ_l is also an eigenvalue of the closed-loop state matrix, define

$$U_l \triangleq \begin{bmatrix} I_{n-1} & 0 \\ 0 & \tilde{Q}_l \end{bmatrix} \qquad (3.2.10)$$

where $\tilde{Q}_l$ is the product of the plane rotations defined in (3.2.4). Then, using (3.2.2)-(3.2.4),

$$\begin{aligned}
&U_l^T \cdots U_0^T\left(A - b\sum_{i=1}^{l} k_i^T\right)U_0 \cdots U_l \\
&= U_l^T\left[U_{l-1}^T \cdots U_0^T\left(A - b\sum_{i=1}^{l-1} k_i^T\right)U_0 \cdots U_{l-1}\right. \\
&\qquad \left. - \left(U_{l-1}^T \cdots U_0^T b\right)\left(k_l^T U_0 \cdots U_{l-1}\right)\right]U_l \\
&= U_l^T\left\{\begin{bmatrix} A_{l-1} & * \\ 0 & A_l \end{bmatrix} - \begin{bmatrix} * \\ b_l \end{bmatrix}\begin{bmatrix} 0 & k_l^T \end{bmatrix}\right\}U_l \\
&= \begin{bmatrix} A_{l-1} & * \\ 0 & \tilde{Q}_l^T\left(A_l - b_l k_l^T\right)\tilde{Q}_l \end{bmatrix}
\end{aligned} \qquad (3.2.11)$$

$$= \begin{bmatrix} A_{l-1} & * & * \\ 0^T & \lambda_l & *^T \\ 0 & 0 & A_{l+1} \end{bmatrix} \tag{3.2.12}$$

where we have substituted for $\tilde{Q}_l^T\left(A_l - b_l k_l^T\right)\tilde{Q}_l$ from (3.2.9). From (3.2.12), it is clear that we have assigned an eigenvalue at λ_l while preserving the previously assigned eigenvalues $\lambda_1, \cdots, \lambda_{l-1}$. Moreover, completing the similarity transformation on the pair (A, b), we get

$$\tilde{Q}_l^T b_l = \left[b_{l,1} \;\; b_{l+1}^T\right]^T. \tag{3.2.13}$$

On noting that controllability is invariant under state feedback, the pair $\left(A_{l+1}, b_{l+1}\right)$ remains a controllable one. Further, because of the sequence in which the non-trivial plane rotations are applied, only the first element of b_{l+1} is non-zero and A_{l+1} is an unreduced upper Hessenberg matrix. Therefore, the EVA process may be continued to assign λ_{l+1} by applying the feedback to the pair $\left(A_{l+1}, b_{l+1}\right)$.

3.2.2. Implicitly shifted EVA

Using the explicitly shifted EVA algorithm we would require complex arithmetic to assign $c-c$ pairs of eigenvalues. Therefore it is important to develop a method that would enable us to assign $c-c$ pairs of eigenvalues using real arithmetic only. This can be done by performing the shifts "implicitly" [15-18]. We shall illustrate the use of implicit shifts for assigning real eigenvalues first and then extend it for assigning $c-c$ pairs of eigenvalues. The following result, in a slightly modified form, is taken from [15]:

Theorem 3.1 : Let $X, Y, Q \in \mathbb{R}^{n \times n}$ with Q orthogonal and Y an unreduced upper Hessenberg matrix with positive sub-diagonal elements. If $Y \triangleq Q^T X Q$, then Y and Q are both uniquely determined by the last column of Q.

Proof : Consider the relation

$$YQ^T = Q^T X.$$

Assume that we have already computed the columns $q_n, q_{n-1}, \cdots, q_{k+1}$ (q_i is the i-th column of the orthogonal matrix Q) and the last $n-k-1$ rows $y_n^T, y_{n-1}^T, \cdots, y_{k+2}^T$ of Y. Since Y is unreduced, we may compute the $k+1$-th row of Y and q_k as

$$y_{k+1,k}\, q_k^T + y_{k+1,k+1} q_{k+1}^T + \cdots + y_{k+1,n}\, q_n^T = q_{k+1}^T X. \tag{3.2.14}$$

By virtue of the fact that Q is orthogonal, $q_i^T q_j = 0$, $i \neq j$, then, postmultiplying the (3.2.14) by q_i, we have

$$y_{k+1,i} = q_{k+1}^T X q_i, \quad i = n, n-1, \cdots, k+1 \tag{3.2.15}$$

thereby determining the $(k+1)$-th row of Y except for $y_{k+1,k}$. However, since Y is unreduced by assumption, $y_{k+1,k} \neq 0$, giving

$$q_k^T = \frac{1}{y_{k+1,k}} \left(q_{k+1}^T X - \sum_{i=n}^{k+1} y_{k+1,i}\, q_i^T \right).$$

Further, since Q is orthogonal, $q_k^T q_k = 1$, and hence q_k^T and the element $y_{k+1,k}$ are uniquely determined. It should be pointed out that since we know q_n, the process for finding the remaining columns of Q and the matrix Y is well started.

Note that if Y is not unreduced *i.e.*, at least one of the sub-diagonal elements of Y is zero, then the eigenspace of X is split into two subspaces such that if $Q = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}$, and $Y = \begin{bmatrix} Y_{11} & Y_{12} \\ 0 & Y_{22} \end{bmatrix}$ where $Q_1 \in \mathbb{R}^{n \times r}$ and $Y_{11} \in \mathbb{R}^{r \times r}$, then $XQ_1 = Q_1 Y_{11}$ [21]. Moreover, if $r = 1$,

$$X q_1 = q_1 y_{1,1}.$$

In other words, if the sub-diagonal element $y_{2,1}$ is zero, then q_1 is an eigenvector of X corresponding to an eigenvalue $y_{1,1}$.

To apply the results of Theorem 3.1 to formalize the basis for implicitly shifted EVA algorithm, assume that one step of QR algorithm in (2.2.5) with shift λ_l has been applied to the unreduced upper Hessenberg matrix A_l. Then we can state the following result:

Theorem 3.2 : Let $\bar{A}_l = Q_l^T A_l Q_l$, where Q_l is the orthogonal part of QR factorization of A_l. We can compute $\bar{A}_l$ employing the following steps:

1) Find an orthogonal matrix P that has the same last column as Q_l

2) Reduce $P^T A_l P$ to upper Hessenberg form $\bar{\bar{A}}_l$, using the transformations in (3.2.4) then $\bar{\bar{A}}_l = \bar{A}_l$.

Proof: To show that $\bar{\bar{A}}_l = \bar{A}_l$, set

$$\bar{Q}_l = PP_{n-1,n} \cdots P_{n-l+1,n-l}$$

where $P_{n-i,n-i+1}$, $i=1, \cdots, l-1$ are the transformations in (3.2.4). Note that unlike the conventional QR decomposition, the decomposition here starts from the lower right hand corner. Now, $\bar{\bar{A}}_l = \bar{Q}_l^T \bar{A}_l \bar{Q}_l$. However, because of the structures of the matrices $P_{n-i,n-i+1}$, post-multiplication by these matrices does not change the *last column* of the product $PP_{n-1,n} \cdots P_{n-l+1,n-l}$. Thus $\bar{Q}_l$ has the same last column as P, which by construction has the same last column as Q_l. Therefore, by Theorem 3.1, $Q_l = \bar{Q}_l$ and hence $\bar{A}_l = \bar{\bar{A}}_l$. Further, since $Q_l = \bar{Q}_l$ the transformed matrix $\bar{Q}^T \left(b_l k_l^T \right) \bar{Q}$ will be the same as $\tilde{Q}^T \left(b_l k_l^T \right) \tilde{Q}$ in (3.2.11). Therefore, the closed-loop matrix $Q_l^T \left(A_l - b_l k_l^T \right) Q_l^T$ will have an eigenvalue at the desired location λ_l.

Clearly, we have concentrated the effect of the shifts of the origin into the transformation matrix P. To determine P, we need to find the last column of Q_l. Consider

$$\left(A_l - \lambda_l I_{n-l+1} \right) Q_l = R_l . \tag{3.2.16}$$

Defining $Q_l \triangleq \left(q_1, \cdots q_n \right)$ and premultiplying the above equation by e_n^T, we have

$$e_n^T \left(A_l - \lambda_l I_{n-l+1} \right) = e_n^T R_l Q_l^T . \tag{3.2.17}$$

Equivalently,

$$\left(0 \; 0 \cdots 0 \; a_{n,n-1}^l \quad a_{n,n}^l - \lambda_l \right) = r_{n,n} q_n^T$$
$$\hat{a}_n^T = r_{n,n} q_n^T .$$

Selecting P such that

$$\hat{a}_n^T P = \| \hat{a}_n \|_2 e_n^T , \tag{3.2.18}$$

we have,

$$e_n^T P^T = \frac{\hat{a}_n^T}{\| \hat{a}_n^T \|_2} = q_n^T \tag{3.2.19}$$

i.e., the last column of P is the same as the last column of Q_l. Hence with this choice of P and therefore of Q_l, we can transform the pair (A, b) and find the feedback k_l^T in (3.2.5) that assigns an eigenvalue at λ_l.

To assign a $c-c$ pair of eigenvalues, we need to form a *double-step* implicit shift. To apply the double-step shift, assume that we have already applied one QR step with a shift λ_l to A_l to obtain

$$\overline{A}_{l+1} = Q_l^T A_l Q_l .$$

We then apply another shift λ_{l+1} to $\overline{A}_{l+1}$ to get

$$\overline{A}_{l+2} = Q_{l+1}^T Q_l^T A_l Q_l Q_{l+1}. \tag{3.2.20}$$

Using similar reasoning as for the single step shift, we can compute $\overline{A}_{l+2}$ by the following the steps:

1) Find an orthogonal matrix P with the same last column as $Q_{l+1} Q_l$.

2) Reduce the matrix $P^T A_l P$ to an upper Hessenberg matrix $\overline{\overline{A}}_{l+2}$.

Once again, because of the structure of the plane rotation matrices required to transform $P^T A_l P$ to an upper Hessenberg matrix $\overline{\overline{A}}_{l+2}$, the product of these matrices have the same last column as $Q_l Q_{l+1}$. Therefore, from Theorem 3.1, $\overline{\overline{A}}_{l+2} = \overline{A}_{l+2}$ and we have accomplished the transformation without explicitly using the shifts λ_1 and λ_2. It can be shown in a manner similar to that for single-step shift that the vectors b_{l+1} and k_{l+1}^T would be the same whether we employ two explicit shifts or one double-step implicit shift. Next, to determine the transformation P, let R_l and R_{l+1} be the upper triangular parts of the QR decomposition of A_l and A_{l+1}. Then,

$$\left(A_l - \lambda_l I_{n-l+1}\right)\left(A_{l+1} - \lambda_{l+1} I_{n-l+1}\right) = R_l R_{l+1} Q_{l+1}^T Q_l^T. \tag{3.2.21}$$

Premultiplying by e_n^T, we get

$$\left(0 \cdots 0 \hat{a}_{n,n-2}\ \hat{a}_{n,n-1}\ \hat{a}_{n,n}\right) = r_{n,n}^{l}\, r_{n,n}^{l+1} q_n^T$$

$$d_n^T = r_{n,n}^l \, r_{n,n}^{l+1} q_n^T \tag{3.2.22}$$

where d_n^T is the last row of $\left(A_l - \lambda_l I_n\right) \left(A_{l+1} - \lambda_{l+1} I_n\right)$, q_n^T is the last row of $Q_{l+1}^T Q_l^T$ and the elements of d_n^T are given by

$$d_{n,n-2} = a_{n-1,n-2} a_{n,n-1} \tag{3.2.23a}$$

$$d_{n,n-1} = a_{n,n-1} \left[a_{n-1,n-1} + a_{n,n} - \left(\lambda_1 + \lambda_2\right) \right] \tag{3.2.23b}$$

$$d_{n,n} = a_{n,n}^2 + a_{n,n-1} a_{n-1,n} - a_{n,n} \left(\lambda_1 + \lambda_2\right) + \lambda_1 \lambda_2 \tag{3.2.23c}$$

Now, if we select P such that

$$d_n^T P = \| d_n \|_2 e_n^T$$

or

$$e_n^T P^T = \frac{d_n^T}{\| d_n \|_2} = \frac{r_{n,n}^l \, r_{n,n}^{l+1}}{\| d_n \|_2} e_n^T Q_l^T Q_{l+1}^T$$

i.e., P and $Q_l Q_{l+1}$ have the same last column.

The main application of the above discussion is in assigning $c-c$ pairs of eigenvalues. In the case when λ_{l+1} is the conjugate of λ_l, (3.2.23) becomes

$$d_{n,n-2} = a_{n-1,n-2} a_{n,n-1} \tag{3.2.24a}$$

$$d_{n,n-1} = a_{n,n-1} \left[a_{n-1,n-1} + a_{n,n} - \left(\lambda_l + \lambda_l^*\right) \right] \tag{3.2.24b}$$

$$d_{n,n} = a_{n,n}^2 + a_{n,n-1} a_{n-1,n} - a_{n,n} \left(\lambda_l + \lambda_l^*\right) + \lambda_l \lambda_l^* \tag{3.2.24c}$$

Noting that $\lambda_l + \lambda_l^*$ and $\lambda_l \lambda_l^*$ are both real, a double shift can be performed using real arithmetic only.

Based on the developments in this section, we now outline an algorithm for eigenvalue assignment for real as well as $c-c$ pairs of eigenvalues.

3.3. An Algorithm for EVA in Single-Input Systems

We consider single-input systems described by (3.2.1). Using Algorithm 2.1, the pair (A, b) can be reduced to an UHF (F, g). It is assumed without loss of generality that the pair (A, b) and therefore, the pair (F, g) is controllable i.e., F is an unreduced upper Hessenberg matrix and $g^T = \left[g_1 0 \cdots 0 0\right]$.

The problem of EVA by means of state feedback is to determine a vector $k^T \in \mathbb{R}^n$ such that the unreduced upper Hessenberg matrix

$$F_{cl} = F - gk^T \tag{3.3.1}$$

has all its eigenvalues at desired locations in the complex plane. These desired locations in the complex plane, corresponding to the eigenvalues of the closed-loop system, are denoted by $\left\{\lambda_i, i=1,2, \cdots, n\right\}$. It is assumed that the desired eigenvalues have been arranged such that the complex eigenvalues occur in conjugate pairs and appear consecutively.

Algorithm 3.1: (EVA in single-input systems using implicit shifts)

Step I: (Initialization)

Set, $k^T = k_0^T = 0^T$, $g_1 = g$, $F_1 = F$, $T = I_n$ and $i = 1$.

Step II: (Real Eigenvalues)

If λ_i is complex, go to *Step III*; else:

(1) If $i = n$, go to (6); else determine an orthogonal matrix P_i such that

$$\hat{f}_i^T P_i = \pm \| \hat{f}_i \|_2 e_n^T \tag{3.3.2}$$

where

$$\hat{f}_i^T = \left[0\ 0 \cdots f_{n,n-1}\ f_{n,n} - \lambda_i\right] \tag{3.3.3}$$

is the last row of $F_1 - \lambda_i I_n$, $\| \hat{f}_i \|_2 = \left(\hat{f}_i^T \hat{f}_i\right)^{\frac{1}{2}}$ and e_n^T is a vector of length n defined as $[0\ 0 \cdots 1]$.

Comment : A matrix P_i can always be found to accomplish this step. The aim is to eliminate $f_{n,n-1}$ using $f_{nn} - \lambda_i$.

(2) Set $\hat{F}_i = P_i^T F_i P_i$ and $\bar{g}_i = P_i^T g_i$.

Comment : Note that the transformations represented by P_i are applied to F_i and not to $F_i - \lambda_i I_n$. This is because the shift has been accounted for in forming the matrix P_i. The matrix $\bar{F}_i$ (*i.e.*, at the i-th iteration) has the following structure:

$$\bar{F}_i = \begin{bmatrix} \lambda_1 & * & \cdot\cdot & * & * & * & \cdot\cdot & * & * & * \\ 0 & \lambda_2 & \cdot\cdot & * & * & * & \cdot\cdot & * & * & * \\ \cdot & \cdot & & \cdot & \cdot & \cdot & & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot\cdot & \lambda_{i-1} & * & * & \cdot\cdot & * & * & * \\ 0 & 0 & \cdot\cdot & 0 & * & * & \cdot\cdot & * & * & * \\ 0 & 0 & \cdot\cdot & 0 & * & * & \cdot\cdot & * & * & * \\ 0 & 0 & \cdot\cdot & 0 & 0 & * & \cdot\cdot & * & * & * \\ \cdot & \cdot & & \cdot & \cdot & \cdot & & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot\cdot & 0 & 0 & 0 & \cdot\cdot & * & * & * \\ 0 & 0 & \cdot\cdot & 0 & 0 & 0 & \cdot\cdot & * & * & * \end{bmatrix}$$

$$= \begin{bmatrix} \bar{F}_i^{11} & \bar{F}_i^{12} \\ 0 & \bar{F}_i^{22} \end{bmatrix} \qquad (3.3.4)$$

Note that the above matrix is not in UHF since its $(n,n-2)$-th element is nonzero. This is a consequence of performing the row operation represented by P_i^T. For ease of representation, it was assumed that the eigenvalues λ_1, λ_1, $\cdots$, λ_{i-1} are real and therefore the submatrix $\bar{F}_i^{11}$ is an upper triangular matrix.

(3) Reduce $\bar{F}_i$ to an UHF $\hat{F}_i$ by means of plane rotations $P_{i,j}$; $j=1,2,\cdots,n-i-1$, i.e.,

$$\hat{F} = P^T_{i,n-i-1} \cdots P^T_{i,1} \bar{F}_i P_{i,1} \cdots P_{1,n-i-1} \qquad (3.3.5a)$$

Also, let

$$g_{i+1} = P^T_{i,n-i-1} \cdots P^T_{i,1} \bar{g}_i . \qquad (3.3.5b)$$

Comment : Note that since the transformation matrices $P_{i,j}$ have the structure $\begin{bmatrix} I_{i-1} & 0 \\ 0 & * \end{bmatrix}$, the $(i-1)\times(i-1)$ upper triangular matrix $\bar{F}_i^{11}$ is not altered by this transformation and only the submatrix $\bar{F}_i^{22}$ is reduced to an unreduced UHM. The vector g_{i+1} has the following structure:

$$g_{i+1} = \left[* \; * \; \cdots \; * \mid * \; * \; 0 \cdots 0 \; 0 \right]^T \tag{3.3.6}$$

where the partition is conformal with the partition of $\hat{F}_i$.

(4) Determine a feedback vector k_i^T such that the $(i+1,i)$-th element of $\hat{F}_i - g_{i+1}k_i^T$ is zero. The vector k_i^T has the following structure:

$$k_i^T = \left[0 \; 0 \; \cdots \; 0 \mid * \; 0 \; 0 \cdots 0 \; 0 \right]. \tag{3.3.7}$$

Comment : If the non-zero element of the feedback vector is chosen so as to reduce the $(i+1,i)$-th element of $\hat{F}_i$ to zero, then the (i,i)-th element of $\hat{F}_i - g_{i+1}k_i^T$ will be equal to λ_i.

(5) Set

$$T_i = P_i P_{i,1} \cdots P_{i,n-i-1} \tag{3.3.8}$$

$$T = TT_i \tag{3.3.9}$$

$$k^T = k^T + k_i^T T^T \tag{3.3.10}$$

$$F_{i+1} = \hat{F}_i - g_{i+1} k^T . \tag{3.3.11}$$

Set $i = i+1$ and go to *Step II.*

Comment : The vector g_{i+1} required for the next iteration is given by (3.3.5b). Equations (3.3.8-9) accumulate all the transformations performed up to that stage and (3.3.10) gives the state feedback vector in the coordinates of (3.3.1).

(6) Determine a feedback vector k_n such that the (n,n)-th element of $F_n - g_n k_n^T$ is equal to λ_n. The vector k_n^T is a vector of length n with only the last element being non-zero.

Comment : The non-zero element in k_n^T is chosen such that

$$\left(F_n\right)_{n,n} - \left(g_n\right)_n \left(k_n^T\right)_n = \lambda_n \tag{3.3.12}$$

where $(F_n)_{n,n}$ denotes the (n,n)-th element of F_n, and $(k_n)_n$ and $(g_n^T)_n$ denote the nth elements of g_n and k_n^T respectively. Note that $(g_n)_n \neq 0$ since the pair (A, b) is controllable.

(7) Set

$$F_{n+1} = F_n - g_n k_n^T \qquad (3.3.13)$$

$$g_{n+1} = g_n \qquad (3.3.14)$$

$$k^T = k^T + k_n^T \qquad (3.3.15)$$

and STOP.

Step III: (Complex-Conjugate pairs of Eigenvalues)

(1) If $i = n - 1$, go to (6); else determine an orthogonal matrix P_i such that

$$f_i^T P_i = \pm \| f_i \|_2 b_n^T \qquad (3.3.16)$$

where

$$f_i^T = [0, 0, \cdots, f_{n,n-2}, f_{n,n-1}, f_{n,n}]$$

is the last row of $(F_i - \lambda_i I_n)(F_i - \lambda_i^* I_n)$, λ_i^* being the *c-c* of λ_i. The elements $f_{n,n-2}$, $f_{n,n-1}$ and f_{nn} are given by

$$f_{n,n-2} = f_{n-1,n-2} f_{n,n-1} \qquad (3.3.17a)$$

$$f_{n,n-1} = f_{n,n-1}\left[f_{n-1,n-1} + f_{nn} - \left(\lambda_i + \lambda_i^*\right) \right] \qquad (3.3.17b)$$

$$f_{nn} = f_{nn}^2 + f_{n,n-1} f_{n-1,n} - f_{nn}\left(\lambda_i + \lambda_i^*\right) + \lambda_i \lambda_i^* \qquad (3.3.17c)$$

where f_{ij} denotes the (i, j)-th element of F_i.

Comment: The elements $f_{n,n-2}$, $f_{n,n-1}$ and f_{nn} are all real since $\lambda_i + \lambda_i^*$ and $\lambda_i \lambda_i^*$ are real. Therefore P_i is real (orthogonal) and can be determined using real arithmetic only.

(2) Set $\bar{F}_i = P_i^T F_i P_i$ and $\bar{g}_i = P_i^T g_i$.

Comment: The matrix $\bar{F}_i$ has the structure shown below:

$$\bar{F}_i = \begin{bmatrix} F_i^{11} & F_i^{12} \\ 0 & F_i^{22} \end{bmatrix} \qquad (3.3.18)$$

where,

$$F_i^{22} = \begin{bmatrix} * & * & \cdot & \cdot & * & * & * & * \\ * & * & \cdot & \cdot & * & * & * & * \\ 0 & * & \cdot & \cdot & * & * & * & * \\ \cdot & \cdot & & & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & & & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & \cdot & \textcircled{*}\,3 & * & * & * \\ 0 & 0 & \cdot & \cdot & \textcircled{*}\,2 & \textcircled{*}\,1 & * & * \end{bmatrix}$$

The matrix $\bar{F}_i$ is not in UHF since its $(n-1, n-3)$-th, $(n, n-3)$-th and

$(n, n-2)$-th elements are nonzero.

(3) Apply plane rotations $P_{i,j}$, $j=1,2, \cdots ,3(n-i-1)$ in order to make F_i^{22} as close to upper Hessenberg as possible, i.e.,

$$\hat{F} = P_{i,3(n-i-1)}^T \cdots P_{i,1}^T \overline{F}_i P_{i,1} \cdots P_{1,3(n-i-1)} \tag{3.3.19a}$$

Also, let

$$g_{i+1} = P_{i,3(n-i-1)}^T \cdots P_{i,1}^T \overline{g}_i . \tag{3.3.19b}$$

Comment : It should be pointed out that F_i^{22} cannot be reduced to the UHF by the transformations described in the above step. The closest we can get to is a matrix with following structure:

$$\hat{F}_i^{22} = \begin{bmatrix} * & * & * & \cdot & \cdot & * & * & * \\ * & * & * & \cdot & \cdot & * & * & * \\ \circledast & * & * & \cdot & \cdot & * & * & * \\ 0 & 0 & * & \cdot & \cdot & * & * & * \\ \cdot & \cdot & \cdot & & & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & & & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \cdot & \cdot & * & * & * \\ 0 & 0 & 0 & \cdot & \cdot & 0 & * & * \end{bmatrix}$$

where the encircled element cannot be eliminated by the orthogonal transformations. The upper triangular matrix F_i^{11} is unaltered by the transformations. The vector g_{i+2} is a vector of length n with the following structure

$$g_{i+2} = \left[* \ * \ \cdots \ * \mid * \ * \ * \ 0 \cdots 0 \right]^T$$

(4) Determine a feedback vector k_i^T such that the $(i+1,i)$-th and $(i+2,i+1)$-th elements of $\hat{F}_i - g_{i+1} k_i^T$ are zero. The required vector k_i^T will have the following structure

$$k_i^T = \left[0\,0 \ \cdots \ 0 \mid * \ * \ 0 \cdots 0 \right]. \tag{3.3.20}$$

where the two nonzero elements are chosen so as to reduce the $(i+2,i)$-th and $(i+2,i+1)$-th elements of $\hat{F}_i$ to zero by applying feedback. This results in a 2×2 matrix in the ith and $(i+1)$-th rows and columns of the closed-loop matrix $\hat{F}_i - g_{i+2} k_i^T$ with eigenvalues λ_i and λ_i^*.

Comment : The matrix $\hat{F}_i - g_{i+2} k_i^T$ has the following structure:

$$\begin{bmatrix} F_i^{11} & * & * \\ 0 & F_i^{22} & * \\ 0 & 0 & F_i^{33} \end{bmatrix}$$

where $F_i^{22} = \begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix}$. This implies that we have assigned two desired eigenvalues at the eigenvalues of the matrix F_i^{22}. Because of the block upper triangular structure of the above matrix, $\hat{F}_i - g_{i+2}k_i^T$ has the desired eigenvalues at λ_i and λ_i^*.

(5) Set

$$T_i = P_i P_{i,1} \cdots P_{i,3(n-i-1)} \tag{3.3.21}$$

$$T = TT_i \tag{3.3.22}$$

$$\boldsymbol{k}^T = \boldsymbol{k}^T + \boldsymbol{k}_i^T T^T \tag{3.3.23}$$

$$F_{i+2} = \hat{F}_i - g_{i+2}\boldsymbol{k}_i^T. \tag{3.3.24}$$

Set $i = i+2$ and go to *Step II.*

Comment : In this step, we increment i by 2 because we have assigned two eigenvalues.

(6) Determine a feedback vector $\boldsymbol{k}_{n-1}^T$ such that the 2×2 matrix in the last two rows and columns of $F_{n-1} - g_{n-1}\boldsymbol{k}_{n-1}^T$ has the eigenvalues λ_i and λ_i^*. The vector $\boldsymbol{k}_{n-1}^T$ will be a vector of length n with only the last two elements being nonzero.

Comment : The structure of the last two rows and columns of $F_{n-1} - g_{n-1}\boldsymbol{k}_{n-1}^T$ is given by

$$\begin{bmatrix} * & * \\ * & * \end{bmatrix} - \begin{bmatrix} * \\ 0 \end{bmatrix} [* \ \ *]. \tag{3.3.25}$$

The effect of applying feedback $\boldsymbol{k}_{n-1}^T$ is to change the first row of the 2×2 matrix above so that by the appropriate choice of the two nonzero elements of the feedback vector, we can ensure that the 2×2 matrix in (3.3.25) has eigenvalues at λ_{n-1} and λ_{n-1}^*.

(7) Set

$$F_{n+1} = F_{n-1} - g_{n-1}\boldsymbol{k}_{n-1}^T \tag{3.3.26}$$

$$g_{n+1} = g_{n-1} \tag{3.3.27}$$

$$\boldsymbol{k}^T = \boldsymbol{k}^T + \boldsymbol{k}_{n-1}^T \tag{3.3.28}$$

and STOP.

At the end of the Algorithm, the vector k contains the gains required to assign all the eigenvalues of the given system to the desired locations in the complex plane.

A few remarks are required to clarify certain points regarding the implementation and properties of the algorithm.

Remarks

(1) Algorithm 3.1 assigns real as well as $c-c$ pairs of eigenvalues using real arithmetic. One element of k^T is determined for each real eigenvalue and two elements for assigning a $c-c$ pair. The results of the algorithm consist of matrices F_{n+1} and T and vectors g_{n+1} and k^T where F_{n+1} is an RSM with each 2×2 block on the diagonal of the matrix corresponding to a desired $c-c$ eigenvalue pair and each scalar on the diagonal being a desired real eigenvalue. The matrix T is an orthogonal matrix that transforms the closed-loop upper Hessenberg matrix $F - gk^T$ to its RSF. The vector k^T is the desired state feedback vector in the coordinates of UHF (F, g).

(2) Steps $II(6)$ and $III(6)$ require a direct computation of the feedback gains. This is necessary because we cannot apply a real implicit shift to a 1×1 block or a double step implicit shift to a 2×2 block. The case of a real eigenvalue in given in Step $II(6)$, for a $c-c$ pair of eigenvalues λ, λ^*, assume that the system in (3.3.25) is given by

$$\begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{bmatrix} - \begin{bmatrix} g_{11} \\ 0 \end{bmatrix} \begin{bmatrix} k_1 & k_2 \end{bmatrix} \qquad (3.3.29)$$

where k_1 and k_2 are to be determined such that the resulting closed-loop matrix has eigenvalues at λ, λ^*. The elements $f_{11}, f_{12}, f_{21}, f_{22}$ and g_{11} are known. By some simple algebraic manipulations, the values of the feedback gains can be determined as

$$k_1 = \frac{1}{g_{11}}\left[f_{11} + f_{22} - (\lambda + \lambda^*)\right] \text{ and} \qquad (3.3.30a)$$

$$k_1 = \frac{1}{f_{21} g_{11}}\left[\lambda\lambda^* + f_{21} f_{12} + f_{22}^2 - f_{22}\left(\lambda + \lambda^*\right)\right]. \qquad (3.3.30b)$$

Since the system has been assumed controllable, the elements f_{21} and g_{11} are both non-zero. Further, since the sum and products of λ and λ^* are real numbers, the feedback gains can be computed using real arithmetic only.

(3) Algorithm 3.1 is based on the implicitly shifted QR algorithm [15,16] for computing the eigenvalues of a matrix. However, unlike QR algorithm, the algorithm for eigenvalue assignment is *not* iterative in nature, since the shifts (desired closed-loop eigenvalues) are known *a-priori*. While the former converges iteratively to the eigenvalues of the matrix, the latter recursively modifies the state matrix using the state feedback and reduces it to a RSM with the desired eigenvalues appearing along the quasi-diagonal.

(4) It is worth noting that the proposed algorithm, unlike some of the algorithms that require the state matrix to be a RSM, would not suffer from round-off errors accumulated in reducing the state matrix to a RSM. Any ill-conditioning in the eigenvalue problem of the open-loop system will, therefore, not directly affect the numerical performance of the algorithm.

(5) EVA problem depends on controllability of the system. For the system (F, g) in UHF, a "weak" controllability may result in small subdiagonal elements of F or a small g_1, thereby leading to higher gains in the feedback matrix. This, however, is due to the ill-conditioning of the EVA problem rather than restriction of the algorithm. A useful feature that may enhance the performance of the algorithm would be an *a-priori* estimate of the condition of the EVAA problem.

3.4. EXTENSION TO THE MULTI-INPUT EVA PROBLEM

For a given multi-input controllable pair (A, B), suppose that the system is in UHF with respect to the first input, *i.e.*, the pair $\left(A, b_1\right)$ is in the UHF, where b_1 is the first input of the system. Let the integer μ denote the dimension of the controllable subsystem from the first input. If $\mu < n$, then the state matrix $F = T^T A T$ is a block upper triangular matrix as shown in (2.3.2), with $F_{11} \in \mathbb{R}^{\mu \times \mu}$ an unreduced upper Hessenberg matrix. The following result then gives us the required extension for solving the EVA problem for multi-input systems.

Theorem 3.3 : Given the controllable pair (A, B) corresponding to the multi-input system (3.1.1), there exists an orthogonal matrix V such that

$$F = V^T A V = \begin{bmatrix} F_{11} & F_{12} & \cdot & \cdot & F_{1,\rho} \\ 0 & F_{22} & \cdot & \cdot & F_{2,\rho} \\ \cdot & \cdot & & & \cdot \\ \cdot & \cdot & & & \cdot \\ 0 & 0 & \cdot & \cdot & F_{\rho,\rho} \end{bmatrix}$$

$$G = V^T B = \begin{bmatrix} g_{11} & g_{11} & \cdot & \cdot & g_{1,\rho} & \cdot & \cdot & g_{1,m} \\ 0 & g_{22} & \cdot & \cdot & g_{2,\rho} & \cdot & \cdot & g_{2,m} \\ \cdot & \cdot & & & \cdot & & & \cdot \\ \cdot & \cdot & & & \cdot & & & \cdot \\ 0 & 0 & \cdot & \cdot & g_{\rho,\rho} & \cdot & \cdot & g_{\rho,m} \end{bmatrix} \qquad (3.4.1)$$

where the matrices F_{ii}, $i = 1, \cdots, \rho$, are unreduced upper Hessenberg matrices and $g_{ii} = [\gamma_i \; 0 \cdots 0]^T$, $i = 1, \cdots, \rho$ with $\gamma_i \neq 0$.

Proof : The proof is by construction and an algorithm for obtaining the matrices F, G, V is outlined next.

Algorithm 3.2: (Reduction of an m-input EVA problem to m single-input EVA problems)

Step I: (Initialization)

(1) Set $i = 1$, $A_1 = A$ and b_1 = first column of B.

(2) Apply Algorithm 2.1 to the pair (A_1, b_1) to get an orthogonal transformation matrix T_1 and dimension of the controllable subspace μ_1.

(3) Set $F_1 = T_1^T A T_1$, $G_1 = T_1^T B$, $V = T_1$ and $n_c = \mu_1$.

Step II: (Determination of Controllable Subsystems)

(1) If $n_c = n$, go to *Step III*, else partition F_i and G_i as follows:

$$F_i = \begin{bmatrix} F_{11} & F_{12} & \cdot & \cdot & F_{1,i} & F_{1,i+1} \\ 0 & F_{22} & \cdot & \cdot & F_{2,i} & F_{2,i+1} \\ \cdot & \cdot & & & \cdot & \cdot \\ \cdot & \cdot & & & \cdot & \cdot \\ 0 & 0 & \cdot & \cdot & F_{ii} & F_{i,i+1} \\ 0 & 0 & \cdot & \cdot & 0 & F_{i+1,i+1} \end{bmatrix}$$

$$G_i = \begin{bmatrix} g_{11} & g_{12} & \cdot & \cdot & g_{1,i} & g_{1,i+1} & \cdot & \cdot & g_{1,m} \\ 0 & g_{22} & \cdot & \cdot & g_{2,i} & g_{2,i+1} & \cdot & \cdot & g_{2,m} \\ \cdot & \cdot & & & \cdot & \cdot & & & \cdot \\ \cdot & \cdot & & & \cdot & \cdot & & & \cdot \\ 0 & 0 & \cdot & \cdot & g_{i,i} & g_{i,i+1} & \cdot & \cdot & g_{i,m} \\ 0 & 0 & \cdot & \cdot & 0 & g_{i+1,i+1} & \cdot & \cdot & g_{i+1,m} \end{bmatrix}$$

where $F_{r,r} \in \mathbb{R}^{\mu_r \times \mu_r}$ and $g_{r,r} \in \mathbb{R}^{\mu_r}$, $r = 1, \cdots, i$. The matrix $F_{i+1,i+1} \in \mathbb{R}^{(n-n_c)\times(n-n_c)}$ and the vector $g_{i+1} \in \mathbb{R}^{n-n_c}$.

(2) Set $i = i+1$, $A_i = F_{ii}$ and $b_i = g_{ii}$.

(3) Apply Algorithm 2.1 to the pair (A_i, b_i) to get an orthogonal transformation matrix T_i and dimension of the controllable subspace μ_i.

(4) Set

$$F_i = \begin{bmatrix} I_{n_c} & 0 \\ 0 & T_i^T \end{bmatrix} F_{i-1} \begin{bmatrix} I_{n_c} & 0 \\ 0 & T_i \end{bmatrix}$$

$$G_i = \begin{bmatrix} I_{n_c} & 0 \\ 0 & T_i^T \end{bmatrix} G_{i-1}$$

$$V = V \begin{bmatrix} I_{n_c} & 0 \\ 0 & T_i \end{bmatrix}$$

and $n_c = n_c + \mu_i$; and go to *Step II* (1)

Step III: (Exit)

Set $F = F_i$, $G = G_i$ and $\rho = i$.

Note that the vectors $g_{\rho,\rho}, \cdots, g_{\rho,m}$ cannot all be zero, because then the system would be uncontrollable, which is contrary to the assumption. This completes the proof of Theorem 3.3.

Algorithm 3.2 uses only numerically stable, orthogonal transformations by repeated application of Algorithm 2.1. Note that integer ρ denotes the smallest number of inputs, starting with the first, from which the system (3.1.1) is controllable. It is also clear from the structures of (F_{ii}, g_{ii}) $i = i, \cdots, \rho$, that they are controllable pairs if (A, B) is a controllable pair. Also, the eigenvalues of F (and therefore of A) are the eigenvalues of F_{ii} $i = 1, \cdots, \rho$. We can therefore assign the eigenvalues of A by carrying out EVA for the single-input systems $\left(F_{ii}, g_{ii}\right)$ $i = i, \cdots, p$ using Algorithm 3.1.

3.5. Some Variations on the Multi-Input EVA Scheme

In this section, the results of the preceding section are modified to allow for greater flexibility in solving the multi-input EVA problem. Some special cases are also considered. From the design point of view, it may be desirable to spread the task of EVA to several inputs instead of the first ρ inputs as suggested in Section 3.4. This may allow the designer to use lower feedback gains to achieve EVA, which in turn would yield better numerical accuracy. There are also some special cases where it may not be possible to assign any desired set of eigenvalues using a subset of the inputs. To illustrate this case, we consider a simple example. Assume that the given system is of order 3 with three inputs and that both the state matrix as well as the input matrix are diagonal with the input matrix having non-zero diagonal entries. The given system is therefore controllable. Also, it is clear that each input can control at most one of the states. Using the single-input EVA scheme, we can assign three *real* eigenvalue at any arbitrary location. However, we cannot assign one real and one *c-c* pair of eigenvalues using Algorithm 3.1.

The case mentioned above can be handled by combining the basic operations of Algorithms 3.1 and 3.2. More specifically, consider the i-th iteration of the EVA scheme where we wish to assign upto r_i eigenvalues by feedback to the i-th input. Assume that n_1 eigenvalues have already been assigned in the preceding steps and that at the start of the i-th iteration, the pair $\left(A_i, B_i\right)$ is in the form

$$A_i = \begin{bmatrix} A_i^{11} & A_i^{12} \\ 0 & A_i^{22} \end{bmatrix}, \quad B_i = \begin{bmatrix} B_i^{1} \\ B_i^{2} \end{bmatrix} \qquad (3.5.1)$$

where $A_i^{11} \in \mathbb{R}^{n_1 \times n_1}$ is an RSM.

$$B_i = Q_{i-1}^T B \qquad (3.5.2)$$

and

$$A_i = Q_{i-1}^T \left(A - BK_{i-1} \right) Q_{i-1} \tag{3.5.3}$$

where Q_{i-1} is an orthogonal matrix and K_{i-1} is the state feedback matrix (in the coordinate system of (A, B)) required to assign some of the eigenvalues of A at the same locations as those of A_i^{11}.

Let $\hat{b}_i$ be the i-th column of B_i^2 and let $W_i \in \mathbb{R}^{n_2 \times n_2}$ be an orthogonal matrix which reduces $\left(A_i^{22}, \hat{b}_i \right)$ to UHF, *i.e.*,

$$W_i^T A_i^{22} W_i = \begin{bmatrix} * & * & \cdot & \cdot & * & * & * \\ * & * & \cdot & \cdot & * & * & * \\ 0 & * & \cdot & \cdot & * & * & * \\ . & . & & . & . & . & . \\ 0 & 0 & \cdot & \cdot & * & * & * \\ 0 & 0 & \cdot & \cdot & 0 & 0 & A_i^{33} \end{bmatrix}, \quad W_i^T \hat{b}_i = \begin{bmatrix} * \\ 0 \\ 0 \\ . \\ 0 \\ 0 \end{bmatrix} \tag{3.5.4}$$

where A_i^{33} denotes the uncontrollable part of $\left(A_i^{22}, \hat{b}_i \right)$.

Complete the coordinate transformation on $\left(A_i ; B_i \right)$ to get

$$\bar{A}_i = \begin{bmatrix} I_{n_1} & 0 \\ 0 & W_i^T \end{bmatrix} A_i \begin{bmatrix} I_{n_1} & 0 \\ 0 & W_i \end{bmatrix}, \quad \bar{B}_i = \begin{bmatrix} I_{n_1} & 0 \\ 0 & W_i^T \end{bmatrix} B_i . \tag{3.5.5}$$

Let the controllable subspace of $\left(A_i^{22}, \hat{b}_i \right)$ be of dimension $\hat{r}_i$ so that by feedback to the i-th input, we can assign at most $\hat{r}_i$ eigenvalues. Note that we wish to assign upto r_i eigenvalues during this iteration. Two cases are possible:

(1) $r_i \leq \hat{r}_i$: Here we assign all r_i eigenvalues by feedback to the i-th input using Algorithm 3.1. Setting $n_1 = n_1 + r_i$ and $n_2 = n - n_1$, we get a pair $\left(A_{i+1}, B_{i+1} \right)$ with

$$A_{i+1} = \begin{bmatrix} A_{i+1}^{11} & A_{i+1}^{12} \\ 0 & A_{i+1}^{22} \end{bmatrix} \quad B_{i+1} = \begin{bmatrix} B_{i+1}^{1} \\ B_{i+1}^{2} \end{bmatrix} \tag{3.5.6}$$

where $A_{i+1}^{11} \in \mathbb{R}^{n_1 \times n_1}$ is an RSM.

(2) $r_i > \hat{r}_i$: In this case, we can only assign at most $\hat{r}_i$ eigenvalues by feedback to the i-th input, therefore, we set $r_i = \hat{r}_i$. Here, we have the following two possibilities:

(i) If the set Λ_i can be partitioned into two disjoint sets Λ_{i_1} and Λ_{i_2}, such that Λ_{i_1} has exactly f_i elements and that the members of a $c-c$ pair belong to the same set, then we use Algorithm 3.1 to assign the f_i eigenvalues in Λ_{i_1} by feedback to the i-th input. Setting $n_1 = n_1 + f_i$ and $n_2 = n - n_1$, we get a pair $\left(A_{i+1}, B_{i+1}\right)$ with the structure in (3.5.6).

(ii) If the set Λ_i cannot be partitioned as in (i), then it can be partitioned such that Λ_{i_1} has f_i-1 elements and all elements of Λ_{i_2} are $c-c$ pairs. The f_i-1 elements in Λ_{i_1} are assigned by the feedback to the i-th input using Algorithm 3.1. Setting, $n_1=n_1+f_i-1$ and $n_2=n-n_1$, we get a pair $\left(A_{i+1}, B_{i+1}\right)$ with the structure in (3.5.6).

The above procedure can be repeated until all n eigenvalues have been assigned at the desired locations or all m inputs have been considered but not all n eigenvalues have been assigned. The latter will occur if case (1) with $r_i \geq f_i$ or (ii) of case (2) has occurred for one or more of the inputs. The former situation can be treated by repeating the above procedure starting with the first input to assign ~~the remaining~~ eigenvalues. The latter situation means that there may be one or more $c-c$ pairs of eigenvalues that cannot be assigned by feedback to one input alone using Algorithm 3.1. These eigenvalues must be assigned by applying feedback to a pair of inputs. A typical situation would involve solving linear equations of the form

$$\begin{bmatrix} a_{i,i} & a_{i,i+1} \\ 0 & a_{i+1,i+1} \end{bmatrix} - \begin{bmatrix} b_{i,p} & b_{i,q} \\ 0 & b_{i+1,q} \end{bmatrix} \begin{bmatrix} k_{p,i} & k_{p,i+1} \\ k_{q,i} & k_{q,i+1} \end{bmatrix} = \begin{bmatrix} \hat{a}_{i,i} & \hat{a}_{i,i+1} \\ \hat{a}_{i+1,i} & \hat{a}_{i+1,i+1} \end{bmatrix}$$

for the four feedback elements $k_{p,i}$, $k_{q,i}$, $k_{p,i+1}$ and $k_{q,i+1}$ where the elements on the right hand side of the equations are chosen such that the matrix $\begin{bmatrix} \hat{a}_{i,i} & \hat{a}_{i,i+1} \\ \hat{a}_{i+1,i} & \hat{a}_{i+1,i+1} \end{bmatrix}$ has the desired $c-c$ pair of eigenvalues. The above equation can be solved for the elements $k_{p,i}$, $k_{p,i+1}$, $k_{q,i}$ and $k_{q,i+1}$ if and only if $b_{i,p} \neq 0$ and $b_{i+1,q} \neq 0$.

3.6. Numerical Examples

In this section, we illustrate the numerical performance of the proposed EVA algorithms by means of some numerical examples. The computations were performed on a VAX 11/780 in double precision (accurate up to 1.0 E–16). The desired closed loop eigenvalues were chosen to illustrate the numerical performance of the algorithms and not to meet any specific design criteria. A comparison of the performances of various existing EVA algorithms with the proposed one is not possible without a detailed analysis of each one of them and therefore has not been included.

Example 3.1: This example illustrates the use of the algorithm for EVA in single-input systems. The example is the 16-th order model of the F100 Turbofan engine [22]. The parameters of the system are given in Table 3.1. The model is controllable from each of its five inputs. For the purpose of illustration, we use the first input to carry out the EVA. To get an idea of the numerical performance of the algorithms, it is assumed that the data given is exact. This is, of course, not true in practice since there will be a degree of uncertainty in the model. Table 3.2 gives the desired closed-loop eigenvalues (column 1) together with the computed closed-loop eigenvalues (column 2).

Example 3.2: This example illustrates the special features described in Section 3.5. For the example considered above, Algorithm 3.1 was applied such that the EVA load was distributed arbitrarily as 3, 3, 3, 4 and 3 from the five inputs respectively. The first input was responsible for assigning eigenvalues at -575.0, -175.0 and -59.0, second at -38.5 and $-17.8\pm4.78i$, third at -50.5 and $-21.3\pm0.8i$, fourth at -21.3, -47.0 and $-6.7\pm1.3i$ and the fifth input at -0.65, -1.9 and -2.6. The results for such a distribution are given in Table 3.2 (column 3). The feedback gains for this case were reduced considerably. When all eigenvalues were assigned from the first input only, the F-norm of the feedback matrix was 48.299 while distributing the EVA between the five inputs resulted in a feedback matrix of norm 10.841.

-4.3280E+00	1.7140E-01	5.3760E+00	4.0160E+02	-7.2460E+02	-1.9330E+00	1.0200E+00	-9.8200E-01	9.9900E-01	1.5210E+00	-4.0620E+00	9.5670E+00	1.0080E+01	-6.0170E-01	-1.3120E-01	9.6020E-02
-4.4020E-01	-5.6430E+00	1.2750E+02	-2.3350E+02	-4.3430E+02	2.6590E+01	2.0400E+00	-2.5920E+00	1.1320E+01	1.0900E+01	-4.0710E+00	-5.7390E-02	-6.0630E-01	-7.4880E-02	-5.9360E-01	-9.6020E-02
1.0380E+00	6.0730E+00	-1.6500E+02	-4.4830E+00	1.0490E+03	-8.2450E+01	-5.3140E+00	5.0970E+00	-9.3890E-03	1.3520E-01	5.6380E+00	2.2460E-02	1.7970E-01	2.4070E-02	1.1000E+00	2.7430E-02
5.3040E-01	-1.0860E-01	1.3130E+02	-5.7830E+02	1.0200E+02	-9.2400E+00	-1.1460E+00	-2.4080E+00	-3.0810E+00	-4.5290E+00	5.7070E+00	-2.3460E-01	-2.1110E+00	-2.4600E-01	-4.6860E-01	-3.2230E-01
8.4760E-03	-1.5630E-02	5.6020E-02	1.5730E+00	-1.0050E+01	1.9520E-01	-8.8040E-03	-2.1100E-02	2.0900E-03	-5.2560E-02	-4.0770E-02	-9.1820E-03	-8.1780E-02	3.4280E-02	4.9950E-03	-1.2560E-02
8.3500E-01	-1.2490E-02	-3.5670E-02	-6.0740E-01	3.7650E+01	-1.9790E+01	-1.8130E-01	-2.9620E-02	-1.9530E-02	-1.6220E-01	-6.4390E-03	-2.3460E-02	-2.2010E-01	-2.5140E-02	-3.7490E-03	-3.3610E-02
6.7680E-01	-1.2640E-02	-9.6830E-02	-3.5670E-01	8.0240E+01	-8.2390E-02	-2.0470E+01	-3.9280E-02	1.8780E-02	-2.1290E-01	-9.3370E-03	-3.1440E-02	-2.9190E-01	-3.3700E-02	8.8730E-02	-4.4580E-02
-9.6960E-02	8.6660E-01	1.6870E+01	1.0510E+00	-1.0230E+02	2.9660E+01	5.9430E-01	-1.9970E+01	2.2530E-02	1.7910E-01	8.3700E-03	2.6450E-02	2.5600E-01	2.8350E-02	-3.7490E-02	3.6360E-02
-8.7850E-03	-1.6360E-02	1.8470E-01	2.1690E-01	-8.4200E+00	7.0030E-01	5.6660E-02	6.6230E+00	-4.9990E+01	6.7600E-02	3.9460E+01	4.9910E-03	8.9830E-02	5.3490E-03	0.0000E+00	1.3720E-02
-1.2980E-04	-2.4300E-04	2.7180E-03	3.2140E-03	-1.2460E-01	1.0390E-02	8.3950E-04	9.8120E-02	-6.6660E-01	-6.6570E-01	5.8470E-01	6.6540E-05	1.3470E-03	7.1310E-05	0.0000E+00	2.0570E-04
-1.2070E+00	-6.7170E+00	2.6260E+01	1.2490E+01	-1.2690E+03	1.0300E+02	7.4800E+00	3.6840E+01	2.8540E-01	2.3320E+00	-4.7650E+01	3.4060E-01	3.0650E+00	3.6240E-01	-4.3430E-01	4.6810E-01
-2.7300E-02	-4.5390E-01	-5.2720E+01	1.9880E+02	-2.8090E+01	2.2430E+00	1.7940E-01	9.7500E+00	-9.6270E+00	-9.5570E+00	3.8480E+01	-5.0010E+01	1.0110E-01	1.2030E-02	-4.6860E-02	1.7150E-02
-1.2060E-03	-2.0170E-02	-2.3430E+00	8.8350E+00	-1.2480E+00	9.9750E-02	8.0590E-03	4.3330E-01	-4.2780E-01	-4.2450E-01	1.7100E+00	-2.0000E+00	-1.9960E+00	5.3490E-04	-1.9990E-03	7.5440E-04
-1.6130E-01	-2.4690E-01	-2.4050E+01	2.3380E+01	1.4630E+02	1.6380E+00	1.3850E-01	4.4860E+00	-4.4140E+00	-4.3540E+00	1.7660E+01	-3.1130E+00	-3.0180E+00	-1.9770E+01	-4.9990E-02	1.5090E-02
-1.2440E-02	3.0200E-02	-1.1980E-01	-4.8210E-02	5.6750E+00	-4.5250E-01	1.9810E+01	1.2490E-01	-1.1270E-03	-6.7600E-03	1.8350E-02	-9.9810E-04	-1.3470E-02	-1.0700E-03	-2.0000E+01	-2.0570E-03
-1.6530E+00	1.8310E+00	-3.8220E+00	1.1340E+02	3.4140E+02	-2.7340E+01	-2.0400E+00	-6.1660E-01	5.0040E-01	-1.4370E-01	-2.4160E+00	-1.0730E-01	-1.0780E+00	3.0630E+01	1.9890E+01	-5.0160E+01

Table 3.1a : F100 Turbofan Engine Model, Matrix A

$$\begin{bmatrix}
-4.5700E-02 & -4.5160E+02 & -1.0580E+02 & -1.5060E+00 & 8.5150E+02 \\
1.1140E-01 & -5.4610E+02 & -6.5750E+00 & -1.0780E+02 & 3.5260E+03 \\
2.1530E-01 & 1.3620E+03 & 1.3460E+01 & 2.0140E+01 & -6.7770E+04 \\
3.2620E-01 & 2.0800E+02 & -2.8880E+00 & -1.6530E+00 & -2.6910E+02 \\
9.9480E-03 & -9.8390E+01 & 5.0690E-01 & -1.9400E-01 & -9.4700E+01 \\
2.7280E-02 & 7.1620E+01 & 9.6080E+00 & -3.1600E-01 & -1.8410E+02 \\
1.7160E-02 & 7.1710E+01 & 8.5710E+00 & 7.9890E-01 & -5.1520E+02 \\
-7.7410E-02 & -1.4120E+02 & -8.2150E-01 & 3.9740E+01 & 1.3760E+03 \\
3.8550E-02 & -7.7100E+00 & -4.3710E-02 & -1.0240E-01 & -6.6840E+03 \\
5.7070E-04 & -1.1440E-01 & -6.3590E-04 & -1.4320E-03 & -9.9020E+01 \\
5.7270E+00 & -1.7450E+03 & -8.9400E+00 & -1.7960E+01 & 8.8980E+04 \\
1.3920E-01 & -2.4300E+01 & -2.7360E-01 & -3.4030E-01 & -6.9310E+03 \\
6.1720E-03 & -1.0820E+00 & -1.1830E-02 & -1.4520E-02 & -3.0770E+02 \\
6.7770E-02 & 1.6600E+01 & 3.9800E-01 & 2.3110E-02 & -2.5880E+03 \\
1.8800E-03 & 9.1470E+00 & -8.2410E-01 & 8.9840E-02 & -3.2310E+01 \\
1.6770E-01 & 4.3580E+02 & -8.9940E+01 & 4.9000E+00 & -2.9550E+02
\end{bmatrix}$$

Table 3.1b : F100 Turbofan Engine Model, Matrix B

$$\left[\begin{array}{cccccc}
4.8660E-01 & -6.7410E-01 & 5.3920E+00 & 9.5420E+01 & 2.4030E+01 & 1.0520E+01 \\
1.3830E-02 & 2.7890E-06 & 0.0000E+00 & 0.0000E+00 & -1.0810E-02 & -5.5450E-05 \\
0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
7.4180E-05 & 5.4960E-06 & 4.7900E-06 & 1.4700E-04 & -1.5040E-02 & -6.5030E-05 \\
-1.5380E-05 & -1.2010E-04 & 2.5790E-03 & 1.6090E-04 & -1.6180E-02 & 1.0710E-03
\end{array}\right.$$

$$\begin{array}{cccccc}
8.1900E-01 & -4.4920E-01 & 5.1950E-01 & 8.4370E-01 & -1.8630E+00 & 5.7090E-02 \\
4.7220E-05 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 1.0000E+00 & 0.0000E+00 \\
8.8200E-05 & 4.9990E-06 & 3.4340E-06 & 2.7270E-05 & 1.1280E-06 & 4.0020E-06 \\
9.5610E-05 & 5.5030E-06 & 3.7320E-06 & 2.9960E-05 & 1.2340E-06 & 4.3800E-06
\end{array}$$

$$\left.\begin{array}{cccc}
4.8150E-01 & 3.4280E+00 & 2.1610E+00 & 7.6810E-02 \\
0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
3.6730E-05 & 4.2900E-06 & -4.9580E-06 & 5.6090E-06 \\
4.0240E-05 & 4.7210E-06 & -5.3240E-06 & 6.1030E-06
\end{array}\right]$$

Table 3.1c : F100 Turbofan Engine Model, Matrix C

Desired	Computed closed-loop eigenvalues for	
e.v.'s	Example 1	Example 2
- 5.75E+02	- 5.750000000000001E+02	- 5.750000000000001E+02
- 1.75E+02	- 1.750000000000000E+02	- 1.750000000000000E+02
- 5.90E+01	- 5.899999999999997E+01	- 5.900000000000082E+01
- 5.05E+01	- 5.050000000000009E+01	- 5.049999999999894E+01
- 4.70E+01	- 4.700000000000013E+01	- 4.700000000000078E+01
- 3.85E+01	- 3.850000000000002E+01	- 3.849999999999928E+01
- 1.78E+01	- 1.779999999999998E+01	- 1.779999999999990E+01
±4.78E+00i	±4.780000000000059E+00i	±4.779999999999864E+00i
- 2.13E+01	- 2.130000000000001E+01	- 2.130000000000030E+01
±8.00E-01i	±8.000000000000613E-01i	±8.000000000005697E-01i
-1.86E+01	- 1.860000000000000E+01	- 1.859999999999976E+01
- 6.70E+00	- 6.700000000000017E+00	- 6.700000000000349E+00
±1.30E+00i	±1.299999999999992E+00i	±1.299999999999160E+00i
- 6.50E- 01	- 6.499999999999976E- 01	- 6.500000000000173E- 01
- 1.90E+00	- 1.899999999999996E+00	- 1.900000000000093E+00
- 2.60E+00	- 2.599999999999958E+00	- 2.599999999999334E+00

Table 3.2 : Desired and computed closed-loop eigenvalues for Examples 1 and 2

3.7. Concluding Remarks

In this chapter, we described numerically reliable algorithms for eigenvalue assignment in linear multivariable systems by means of state feedback. For a multi-input systems (A, B), the EVA problem was reduced to one or more single-input EVA problems. The algorithms for EVA were based on the converse of the principle on which implicitly shifted QR algorithm for eigenvalue determination is based. In the proposed algorithms, a double step enabled us to assign a $c-c$ pair of eigenvalues using real arithmetic only. Starting with a controllable pair (F, g) in the UHF, Algorithm 3.1 determines a state feedback vector k^T and an orthogonal transformation matrix T such that the closed loop system $(F - gk^T T)$ has the desired set of eigenvalues. Apart from the fact that the shifts (the desired closed-loop eigenvalues) are known *a priori*, an iteration of Algorithm 3.1 differs from the QR algorithm essentially in the step that computes the elements of the feedback vector k^T. In fact, a few minor changes to a computer code for QR algorithm [23,24] would enable us to obtain a code for Algorithm 3.1.

For multi-input systems, a variation in the application of Algorithms 3.1 and 3.2

enables us to distribute the task of EVA between several inputs. Apart from enabling us to assign any desired set of eigenvalues, this modification is also desirable from the design point of view since it provides greater integrity (against failure in some feedback paths). Furthermore, the feedback gains will in general be smaller than when a single or only a few inputs were used.

The numerical performance of the algorithms was illustrated by means of a model of the F100 Turbofan engine. The elements of various matrices describing the system are widely separated. However, as can be seen from the computed closed-loop eigenvalues, they do not seem to affect the accuracy of the algorithm.

It is perhaps worth mentioning that while for the single-input systems, the feedback vector is unique, for the multi-input systems, the feedback matrix is highly nonunique and provides us with extra freedom [25]. There are several ways in which the researchers have tried to use this freedom. In [11], the extra freedom has been used to make the system "robust" in presence of errors in the system parameters, while the proposed method can utilize it to lower the feedback gains. The best way to utilize this freedom is still an open question and would make an interesting research problem. It appears that the order in which the eigenvalues are assigned (in the multi-input systems) and the choice of inputs from which those eigenvalues are assigned also has some effect on the feedback gains. A systematic method of determining the subset of eigenvalues and the order in which they should be assigned from a particular input so as to obtain a minimum norm state feedback matrix is also open problem.

Since the EVA by state feedback is dependent on the controllability of the open-loop system, it is reasonable to assume that the conditioning of the EVA problem is related to the controllability of the open-loop system. It is not clear how the two are related explicitly. A more detailed analysis of the problem may reveal this relationship, and lead to design of an algorithm with an even better numerical properties than the proposed one.

3.8. REFERENCES

[1] F. Fallside Ed., *Control System Design by Pole-Zero Assignment*, Academic Press, London, 1977.

[2] W.H. Wonham, *Linear Multivariable Control: A Geometric Approach*, Springer-Verlag, N.Y., 1979.

[3] T. Kailath, *Linear Systems*, Prentice Hall, Englewood-Cliffs, N.J., 1980.

[4] R.V. Patel and N. Munro, *Multivariable System Theory and Design*, Pergamon Press, Oxford, 1982.

[5] B.D.O. Anderson and D.G. Luenberger, "Design of Multivariable Feedback Systems", *Proc. Inst. Elect. Eng.*, Vol. 114, pp. 395-399, 1967.

[6] F. Fallside and H. Seraji, "Direct Design Procedure for Multivariable Feedback Systems", *Proc. Inst. Elect. Eng.*, Vol. 118, pp. 797-801, 1971.

[7] G.S. Miminis and C.C. Paige, "An Algorithm for Pole Assignment of Time-Invariant Linear Systems", *Int. J. Contr.*, Vol. 35, pp. 341-354, 1982.

[8] G.S. Miminis and C.C. Paige, "An Algorithm for Pole Assignment of Time Invariant Multi-Input Systems", *21st IEEE Conf. Decision and Contr.*, Orlando, FL, pp.62-67, 1982.

[9] R.V. Patel and P. Misra, "Numerical Algorithms for Eigenvalue Assignment by State Feedback", *Proc. IEEE*, Vol. 72, pp. 1755-1764, 1984.

[10] R.V. Patel and P. Misra, "Numerical Considerations in Pole Assignment by State Feedback", *IEEE International Conference on Circuits, Systems and Signal Processing*, Bangalore, India, pp. 487-490, Dec. 1984.

[11] J. Kautsky, N.K. Nichols and P. VanDooren, "Robust Pole Assignment in Linear Feedback", *Int. J. Contr.*, Vol. 41, pp. 1129-1155, 1985.

[12] B.N. Datta, "An Algorithm to Assign Eigenvalues in a Hessenberg Matrix: Single Input Case", *IEEE Trans. Automat. Contr.*,, Vol. AC-32, pp. 414-417, 1987.

[13] M.M. Konstantinov, P.H. Petkov and N. Christov, "A Schur Approach to the Pole Assignment Problem", in *Proc. IFAC 8th World Congress*, Kyoto, Japan, 1981.

[14] A. Varga, "A Schur Method for Pole Assignment", *IEEE Trans. Automat. Contr.*, Vol. AC-26, pp. 517-519, 1981.

[15] G.W. Stewart, *Introduction to Matrix Computations*, Academic Press, N.Y., 1973.

[16] J.H. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford University Press, London, 1965.

[17] J.G.F. Francis, "The QR Transformation I", *Computing Jour.*, Vol. 4, pp. 265-271, 1961.

[18] J.G.F. Francis, "The QR Transformation II", *Computing Jour.*, Vol. 4, pp. 332-345, 1961.

[19] G.W. Stewart, "HQR3 and EXCHNG: Fortran Subroutines for Calculating and Ordering the Eigenvalues of a Real Upper Hessenberg Matrix", *ACM Trans. Math. Software*, Vol. 2, pp. 275-280, 1976.

[20] P. Misra and R.V. Patel, "A Simplified Algorithm for Rearranging the Eigenvalues of an Upper Hessenberg Matrix", to be presented at *Int. Conf. on Linear Algebra and Applications*, Sept. 28-30, Spain, 1987.

[21] G.S. Miminis, "Numerical Algorithm for Controllability and Eigenvalue Allocation", M. Comp. Sc. dissertation, School of Comp. Sci., McGill Univ., Montreal, 1981.

[22] M.K. Sain, J.L. Peczkowski and J.L. Melsa, *Eds., Alternatives for Linear Multivariable Control*, Nat. Eng. Consortium, Chicago, 1978.

[23] B.T. Smith, *et al.*, *Matrix Eigensystems Routines - EISPACK Guide*, Lecture notes in Comp. Sci., Vol. 6, Springer Verlag, N.Y., 1976.

[24] J.J. Dongarra, *et al.*, *LINPACK Users' Guide*, Philadelphia, SIAM, P.A., 1979.

[25] B.C. Moore, "On the Flexibilty Offered by State Feedback in Multivariable Systems Beyond Closed Loop Eigenvalue Assignment", *IEEE Trans. Automat. Contr.*, Vol. AC-21, pp. 689-692, 1976.

CHAPTER IV

EIGENVALUE ASSIGNMENT BY OUTPUT FEEDBACK

In this chapter, we extend the results of the previous chapter to EVA by means of output feedback. The chapter is organized as follows: The problem of EVA by means of output feedback is stated formally in Section 4.1 where a survey of existing methods for EVA is also included. In Section 4.2, we develop the theoretical foundation on which the algorithms for EVA in single-input, multi-output (SIMO) systems are based. The results are then extended to multi-input, multi-output (MIMO) systems, where it is shown that the EVA problem for MIMO systems can be reduced to solving EVA problems for two SIMO systems. A sufficient condition under which the eigenvalues of the closed-loop system can be assigned arbitrarily close to desired locations in the complex plane is that for the given controllable and observable system triple (A, B, C), the sum of number of inputs (m) and number of outputs (p) is greater than the number of states (n) [1,2]. Based on this condition, numerical algorithms for EVA by means of output feedback in SIMO and MIMO systems are then presented in Section 4.3. In Section 4.4 it is shown that the same algorithm can be applied (with slight modifications) to compute dynamic output feedback for the case when the sufficient condition $(m + p > n)$ is not satisfied. Numerical properties of the algorithm are illustrated by means of examples in Section 4.5 and finally in Section 4.6, we discuss the results presented in this chapter.

4.1. STATEMENT OF THE PROBLEM

Consider a linear time-invariant multivariable system described by its state-space equations

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{4.1.1a}$$
$$y(t) = Cx(t) \tag{4.1.1b}$$

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$ and $y(t) \in \mathbb{R}^p$. We assume that (A, B, C) is a controllable and observable triple. It is desired to compute a constant gain output feedback matrix $K \in \mathbb{R}^{m \times p}$ defined by the feedback law

$$u(t) = v(t) - Ky(t) \tag{4.1.2}$$

such that the resulting closed-loop state matrix

$$A_{cl} = A - BKC \tag{4.1.3}$$

has some desired set of eigenvalues. In (4.1.3), if the matrix C is an identity matrix of order n, then the closed-loop state matrix is given by $A_{cl} = A - BK$, and the problem is reduced to that of EVA by means of state-feedback [3-5] which was considered in the previous chapter. EVA by output feedback can therefore be considered as a generalization of EVA by state feedback. For EVA by means of output feedback, considerable theoretical work has been done [1,2,6-9]. However, until recently, hardly any attention was paid to developing numerically reliable algorithms to solve the problem. The algorithms described in this chapter were reported in [10-12]. The results presented in this chapter and in [10-12] appear to be the first of their kind in the literature. Consequently, the derivations required in developing the algorithms have been described in detail.

There are several other numerical algorithms existing in the literature that achieve EVA by output feedback. However, they are not based on sound numerieal analysis principles and therefore their numerical reliability is questionable. Several algorithms [6-9] require the system to be in *companion* or other *canonical* forms *e.g.*, controller or controllability canonical forms [8]. Such algorithms will invariably incur numerical difficulties because reduction of a general triple (A, B, C) to a canonical form is a numerically unstable step. Algorithms that use the transfer function matrix [6-9] of the given system will be extremely sensitive to perturbations in the coefficients of the numerator and denominator polynomials of transfer function matrix. Another approach is based on knowledge of the eigenvalues of the open-loop system. However, for systems with state matrices having ill-conditioned open-loop eigenvalue problems, this approach may lead to unsatisfactory performance.

The algorithms described in subsequent sections avoid the use of potentially unstable transformations. EVA by output feedback (as that for by-state feedback) is treated as a "converse" of the algebraic eigenvalue problem. In developing the algorithms, we use sufficient conditions [1,2] under which the feedback law of the type (4.1.2) exists.

4.2. Eigenvalue Assignment by Constant Gain Output Feedback

In this section, we will develop the theory on which the EVA algorithm is based. For the sake of simplicity, the results for explicitly shifted EVA are discussed for real eigenvalues. The case for *c-c* pairs of eigenvalues is similar to that for the EVA problem by means of state feedback, treated in the previous chapter and has not been repeated here. Using the sufficient condition [1,2], we will develop an algorithm for "almost" arbitrary EVA using constant gain output feedback (of rank 2) for the systems in which $(m + p > n)$

In order to achieve this, we reduce the EVA problem to two single-input EVA problems, the first to assign $p-1$ eigenvalues and the other to assign the remaining $n-p+1$ eigenvalues while preserving the $p-1$ previously assigned eigenvalues. In Section 4.2.1., we discuss the design philosophy for EVA in single-input, p-output systems using explicit shifts and in Section 4.2.2., the results are extended to EVA in multi-input, multi-output systems.

4.2.1. Explicitly Shifted EVA for Single-Input, Multi-Output Systems

Consider a controllable and observable single-input multi-output system

$$\dot{x}(t) = Ax(t) + bu(t) \tag{4.2.1a}$$
$$y(t) = Cx(t) \tag{4.2.1b}$$

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}$, $y(t) \in \mathbb{R}^p$ and rank $(C) = p$ $(\leq n)$. We shall assume without any loss of generality that the given system triple (A, b, C) is in UHF. Moreover, since the system (4.2.1) is controllable, the state matrix A is an unreduced upper

Hessenberg matrix.

It is well-known that in a controllable and observable single-input system with p outputs, we can assign p closed-loop eigenvalues at *almost* any desired locations in the complex plane by means of *finite, constant gain* output feedback. The locations where the eigenvalues cannot be positioned exactly by finite output feedback correspond to the transmission zeros of the system [1,2,6-9]. Note that we have no control over the remaining $n-p$ eigenvalues. We next discuss how "explicit" shifts of origin may be employed to assign the eigenvalues by means of output feedback.

The problem that we consider in this section is to find a constant gain output feedback vector $\boldsymbol{k} \in \mathbb{R}^p$ such that the closed-loop state matrix $A_{cl} = A + \boldsymbol{bk}^T C$ has r $(\leq p)$ eigenvalues at desired locations. Let $\Lambda = \left\{\lambda_1, \cdots, \lambda_r\right\}$ denote the set of desired locations to which the closed-loop eigenvalues are to be assigned and assume that at the start of the l-th step ($l \leq r$), we have

$$U_{l-1}^T \cdots U_0^T \left(A - b\sum_{i=1}^{l-1} \boldsymbol{k}_i^T C\right) U_0 \cdots U_{l-1} = \begin{bmatrix} A_{l-1} & * \\ 0 & A_l \end{bmatrix} \qquad (4.2.2)$$

where by the assumption of controllability, $A_l \in \mathbb{R}^{(n-l+1)\times(n-l+1)}$ is an unreduced upper Hessenberg matrix, A_{l-1} is a upper Schur matrix with the assigned eigenvalues, $\lambda_1 \cdots \lambda_{l-1}$, along its diagonal, $U_0 = I_n$ (the $n \times n$ identity matrix) and $\sum_{i=1}^{l-1} \boldsymbol{k}_i^T$ is the output feedback vector;

$$U_{l-1}^T \cdots U_0^T b = \begin{bmatrix} b_{l-1}^T & b_l^T \end{bmatrix}^T \qquad (4.2.3)$$

with $\boldsymbol{b}_l \ (\in \mathbb{R}^{n-l+1}) = \begin{bmatrix} b_{l,1}\, 0 & \cdots & 0 \end{bmatrix}^T$ and

$$CU_0 \cdots U_{l-1} = \begin{bmatrix} c_1\, c_2 & \cdots & c_{l-1}\, C_l \end{bmatrix} \qquad (4.2.4)$$

where $\boldsymbol{c}_i, i = 1, \cdots, l-1$ are $p \times 1$ vectors.

In the l-th step, we calculate $\boldsymbol{k}_l^T$ in order to assign an eigenvalue at λ_l. To accomplish this, we first transform the matrix $A_l - \lambda_l I_{n-l+1}$ to an upper triangular form by means of orthogonal plane rotations

$$\left(A_l - \lambda_l I_{n-l+1}\right) P_{n-1,n} \cdots P_{n-l+1,n-l} \triangleq R_l . \tag{4.2.5}$$

where, $P_{n-1,n}$ is a plane rotation in the $(n-1,n)$-th plane [10,11] such that the element $a_{n,n-1}$ is annihilated using the element $a_{n,n} - \lambda_l$. Since A_l is an unreduced upper Hessenberg matrix, the plane rotations above are non-trivial and the diagonal elements of R_l have non-zero values.

Denoting the product $P_{n-1,n} \cdots P_{n-l+1,n-l}$ by $\tilde{Q}_l$, we have

$$C_l \tilde{Q}_l = \left[\tilde{c}_l \; \tilde{c}_{l+1} \; \cdots \; \tilde{c}_n\right]. \tag{4.2.6}$$

In order to preserve the $(l-1)$ eigenvalues already assigned at the locations $\lambda_1, \cdots \lambda_{l-1}$, we let

$$k_l^T = \theta_l \tilde{k}_l^T T \tag{4.2.7}$$

where $\tilde{k}_l^T \in \mathbb{R}^p$ is selected such that

$$\tilde{k}_l^T T c_i = 0 \;,\; i=1, \cdots, l-1 \tag{4.2.8a}$$

$$\tilde{k}_l^T T \tilde{c}_j \neq 0 \;,\; j=l, \cdots, n \tag{4.2.8b}$$

and T is an orthogonal transformation matrix to be determined. In (4.2.8) above, c_i and $\tilde{c}_l$ are the vectors defined in (4.2.4) and (4.2.6) respectively, and θ_l is a scalar that is to be determined so as to assign an eigenvalue at λ_l. We define a vector ϕ_l as

$$\phi_l = \left[\phi_{l,1} \; \phi_{l,2} \; \cdots \; \phi_{l,n-l+1}\right]^T \triangleq \tilde{k}_l^T T C_l \tilde{Q}_l . \tag{4.2.9}$$

Next, consider the matrix $A_l - \lambda_l I_{n-l+1} - b_l k_l^T C_l$. With the similarity transformation $\tilde{Q}_l$ applied *i.e.*, using (4.2.5)-(4.2.9), we have

$$\tilde{Q}_l^T \left(A_l - \lambda_l I_{n-l+1} - b_l k_l^T C_l\right) \tilde{Q}_l = \tilde{Q}_l^T \left(R_l - b_l \theta_l \phi_l^T\right). \tag{4.2.10}$$

Because of the structures of R_l and b_l, the matrix $R_l - b_l \theta_l \phi_l^T$ is upper triangular and its first row is given by

$$\left[r^l_{1,1} - b_{l,1}\theta_l \phi_{l,1} \;\; r^l_{1,2} - b_{l,1}\theta_l \phi_{l,2} \;\cdots\; r^l_{1,n-l+1} - b_{l,1}\theta_l \phi_{l,n-l+1}\right]. \tag{4.2.11}$$

where $r^l_{i,j}$, $i,j=1,\cdots,n-l+1$ is the (i,j)-th element of R_l. We choose θ_l to eliminate the (1,1) element of $R_l - b_l\theta_l\phi_l^T$. By assumption of controllability $b_{l,1} \neq 0$ and from (4.2.8a), $\phi_{l,1} \neq 0$ therefore we can eliminate the (1,1) element of $R_l - b_l\theta_l\phi_l^T$ by selecting

$$\theta_l = \frac{r^l_{1,1}}{b_{l,1}\phi_{l,1}}. \qquad (4.2.12)$$

Then, the first column of $\tilde{Q}_l^T\left(R_l - b_l\theta_l\phi_l^T\right)$ and therefore, the first column of the expression on the left hand side of (4.2.10) are zero vectors. Also, the matrix

$$\tilde{Q}_l^T\left(A_l - \lambda_l I_{n-l+1} - b_l k_l^T C_l\right)\tilde{Q}_l + \lambda_l I_{n-l+1} = \tilde{Q}_l^T\left(A_l - b_l k_l^T C_l\right)\tilde{Q} \qquad (4.2.13)$$

from which it follows that

$$\tilde{Q}_l^T\left(A_l - b_l k_l^T C_l\right)\tilde{Q}_l = \begin{bmatrix} \lambda_l & *^T \\ 0 & A_{l+1} \end{bmatrix} \qquad (4.2.14)$$

i.e., the matrix $A_l - b^T k_l^T C_l$ has an eigenvalue at λ_l. To verify that λ_l is also an eigenvalue of the closed-loop state matrix, define

$$U_l \triangleq \begin{bmatrix} I_{n-l} & 0 \\ 0 & \tilde{Q}_l \end{bmatrix} \qquad (4.2.15)$$

where $\tilde{Q}_l$ is the product of plane rotations defined in (4.2.5). Then, using (4.2.2)-(4.2.5),

$$\begin{aligned}
& U_l^T \cdots U_0^T\left(A - b\sum_{i=1}^{l} k_i^T C\right)U_0 \cdots U_l \\
&= U_l^T\left[U_{l-1}^T \cdots U_0^T\left(A - b\sum_{i=1}^{l-1} k_i^T C\right)U_0 \cdots U_{l-1}\right. \\
&\qquad \left. - \left(U_{l-1}^T \cdots U_0^T b\right)k_l^T\left(CU_0 \cdots U_{l-1}\right)\right]U_l \\
&= U_l^T\left\{\begin{bmatrix} A_{l-1} & * \\ 0 & A_l \end{bmatrix} - \begin{bmatrix} * \\ b_l \end{bmatrix}\begin{bmatrix} 0 & k_l^T C_l \end{bmatrix}\right\}U_l \\
&= \begin{bmatrix} A_{l-1} & * \\ 0 & \tilde{Q}_l^T\left(A_l - b_l k_l^T C_l\right)\tilde{Q}_l \end{bmatrix} \qquad (4.2.16) \\
&= \begin{bmatrix} A_{l-1} & * & * \\ 0^T & \lambda_l & *^T \\ 0 & 0 & A_{l+1} \end{bmatrix} \qquad (4.2.17)
\end{aligned}$$

where we have substituted for $\tilde{Q}_l^T(A_l - b_l k_l^T C_l)\tilde{Q}_l$ from (4.2.14). From (4.2.17), it is clear that we have assigned an eigenvalue at λ_l while preserving the previously assigned eigenvalues $\lambda_1, \cdots, \lambda_{l-1}$. Moreover, completing the similarity transformation on the triple (A, b, C), we get

$$\tilde{Q}_l^T b_l = \begin{bmatrix} b_{l,1} & b_{l+1}^T \end{bmatrix}^T \tag{4.2.18a}$$

and

$$C\tilde{Q}_l = \begin{bmatrix} \tilde{c}_l & \tilde{C}_{l+1} \end{bmatrix}. \tag{4.2.18b}$$

On noting that controllability is invariant under output feedback, the first element of b_{l+1} must be non-zero and A_{l+1} must be an unreduced upper Hessenberg matrix. Therefore, the EVA process may be continued to assign λ_{l+1} by output feedback to the triple $\left(A_{l+1}, b_{l+1}, \tilde{C}_{l+1}\right)$.

There are certain points that must be considered to formalize the EVA algorithm. The first one is to show how the vector $\boldsymbol{k}^T$ can be selected to satisfy (4.2.7) and (4.2.8). To do this we compute an orthogonal matrix T such that

$$\overline{C} = T\begin{bmatrix} c_1 & c_2 & \cdots & c_{l-1} & \tilde{c}_l & \cdots & \tilde{c}_n \end{bmatrix} \tag{4.2.19}$$

is a lower *row echelon* matrix. For example, a 3×5 row echelon matrix may have the following structure:

$$\overline{C} = \begin{bmatrix} 0 & 0 & 0 & 0 & * \\ 0 & 0 & * & * & * \\ * & * & * & * & * \end{bmatrix}$$

The number of linearly independent columns of $\overline{C}$ can be determined by inspection of its lower row echelon form. If the rank of the first $l-1$ columns is denoted by α_{l-1}, then the vector $\boldsymbol{k}_l^T$ is chosen as

$$\boldsymbol{k}_l^T = \theta_l \bar{\boldsymbol{k}}_l^T T \tag{4.2.20a}$$

where

$$\theta_l \bar{\boldsymbol{k}}_l^T \triangleq [\underbrace{0 \cdots 0}_{p-\alpha_{l-1}} * \underbrace{0 \cdots 0}_{\alpha_{l-1}}]. \tag{4.2.20b}$$

From (4.2.20) it is easy to see that the condition (4.2.8a) will always be satisfied. To satisfy (4.2.8b), it is required that $c_{p-\alpha_{l-1},l} \neq 0$. However, if $c_{p-\alpha_{l-1},l} = 0$, the element $\phi_{l,1}$ in (4.2.11) is equal to zero. This situation occurs when $T\bar{c}_l$ in (4.2.19) is linearly dependent on $c_1, \cdots, c_{l-1}$. In this case we cannot assign an eigenvalue at λ_l as shown by the following result.

Lemma 3.1 : If $\phi_{l,1} = 0$, then λ_l is a transmission zero of the single-input, single-output system $\left(A_l, b_l, k_l^T C_l\right)$.

Proof : Consider the matrix.

$$Z(\lambda_l) = \begin{bmatrix} A_l - \lambda_l I_{n-l+1} & b_l \\ k_l^T C_l & 0 \end{bmatrix} \tag{4.2.21}$$

After performing the plane rotations as described by (4.2.5), we have

$$Z(\lambda_l)\,\bar{Q}_l = \begin{bmatrix} r_{1,1} & r_{1,2}^T & b_{l,1} \\ 0 & R_{2,2} & 0 \\ 0 & k_l^T \bar{C}_{l+1} & 0 \end{bmatrix} \tag{4.2.22}$$

from where it is clear that the first and the last columns of $Z(\lambda_l)\,\bar{Q}_l$ are linearly dependent. Therefore, rank $\left[Z(\lambda_l)\right] < n-l+2$ which in turn implies [12] that λ_l is a transmission zero of $\left(A_l, b_l, k_l^T C_l\right)$.

It is well known that the eigenvalues of a closed-loop system cannot be positioned exactly at the transmission zeros of the system by means of finite gain output feedback. Theoretically, we can position an eigenvalue arbitrarily close to a transmission zero, however, it would require undesirably high feedback gains. We should note here that if λ_l is a transmission zero of $\left(A_l, b_l, C_l\right)$, then it would also be a transmission zero of $\left(A_l, b_l, k_l^T C_l\right)$ but the converse is not necessarily true. If λ_l is a transmission zero of $\left(A_l, b_l, k_l^T C_l\right)$ but not of $\left(A_l, b_l, C_l\right)$, then we can assign the remaining eigenvalues by reordering $\lambda_l, \cdots, \lambda_r$ such that λ_l is assigned in one of the subsequent steps. However, if λ_l is a transmission zero of $\left(A_l, b_l, C_l\right)$, then we cannot assign an eigenvalue at λ_l by means of finite gain output feedback $\dot{k}_l^T$. In this case, the following result implies that an eigenvalue cannot be assigned exactly at λ_l by means of finite gain output feedback applied to the original system (A, b, C).

Lemma 3.2 : Let (A, b, C) be a controllable and observable system. If λ_l is a transmission zero of $\left(A_l, b_l, C_l\right)$ defined in (4.2.2)-(4.2.4), then it is also a transmission zero of (A, b, C).

In order to prove the above Lemma, we need the following result:

Lemma 3.3 : A scalar λ_l is a transmission zero of $\left(A_l, b_l, C_l\right)$ if and only if $\bar{c}_l = 0$.

Proof of Lemma 3.3 : By definition, λ_l is a transmission zero of $\left(A_l, b_l, C_l\right)$ if and only if

$$\text{rank} \begin{bmatrix} A_l - \lambda_l I_{n-l+1} & b_l \\ C_l & 0 \end{bmatrix} < n - l + 2 \qquad (4.2.23)$$

Also,

$$\text{rank} \begin{bmatrix} A_l - \lambda_l I_{n-l+1} & b_l \\ C_l & 0 \end{bmatrix} = \text{rank} \left\{ \begin{bmatrix} A_l - \lambda_l I_{n-l+1} & b_l \\ C_l & 0 \end{bmatrix} \begin{bmatrix} \bar{Q}_l & 0 \\ -w^T \bar{Q}_l & 1 \end{bmatrix} \right\}$$

where $\bar{Q}_l$, and therefore the post-multiplication matrix, has full rank. In the above equation, w^T is chosen such that the (1,1) element of $\left(A_l - \lambda_l I_{n-l+1}\right) \bar{Q}_l$ becomes zero. It is easy to show that this can always be done *e.g.* by letting $w^T \bar{Q}_l = [* \ 0 \cdots 0]$. The vector w^T is of length $n-l$. Therefore,

$$\text{rank} \begin{bmatrix} A_l - \lambda_l I_{n-l+1} & b_l \\ C_l & 0 \end{bmatrix} = \text{rank} \begin{bmatrix} \left(A_l - \lambda_l I_{n-l+1} - b_l w^T\right) \bar{Q}_l & b_l \\ C_l \bar{Q}_l & 0 \end{bmatrix}$$

$$= \text{rank} \begin{bmatrix} 0 & r_{1,2}^T & b_{l,1} \\ 0 & R_{2,2} & 0 \\ \bar{c}_l & \bar{C}_{l+1} & 0 \end{bmatrix} \qquad (4.2.24)$$

In (4.2.24), interchanging the first and the last columns of the matrix, it is easy to see that the rank of the matrix on the left hand side is less than $n-l+2$ if and only if $\bar{q}_l=0$, completing the proof of the Lemma.

Proof of Lemma 3.2 : The transmission zeros of (A,b,C) are invariant under nonsingular state and output coordinate transformations and output feedback [I-4]. Therefore, the transmission zeros of (A,b,C) are the same as those of the system

$$\left(Q_{l-1}^T \cdots Q_0^T\left[A - b\sum_{i=1}^{l-1}k_i^TC\right]Q_0 \cdots Q_{l-1},\ Q_{l-1}^T \cdots Q_0^Tb,\ TCQ_0 \cdots Q_{l-1}\right).$$

Then from (4.2.2)-(4.2.4),

$$\text{rank}\begin{bmatrix} A - \lambda_l I_n & b \\ C & 0 \end{bmatrix} = \text{rank}\begin{bmatrix} A_{l-1} - \lambda_l I_{l-1} & * & b_{l-1} \\ 0 & A_l - \lambda_l I_{n-l+1} & b_l \\ C_{l-1} & C_l & 0 \end{bmatrix} \tag{4.2.25}$$

Since λ_l is a transmission zero of $\left(A_l, b_l, C_l\right)$, by Lemma 3.2, there exists an orthogonal matrix $\bar{Q}_l$ such that $\left(A_l - \lambda_l I_{n-l+1}\right)\bar{Q}_l$ is upper triangular and $C_l\bar{Q}_l = \begin{bmatrix} 0 & \bar{C}_{l+1}\end{bmatrix}$. Using elementary row and column operations, we can reduce the elements b_{l-1}, b_l and C_{l-1} zero without altering the structure of the rest of the matrix. The rank of the matrix on the right hand side of (4.2.25) is given by

$$\begin{aligned}
&\mathit{rank}\begin{bmatrix} I_{l-1} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & I_{n-l} & 0 \\ -C\left(A_{l-1}-\lambda_l I_{l-1}\right)^{-1} & 0 & 0 & I_p \end{bmatrix}\begin{bmatrix} A_{l-1} - \lambda_l I_{l-1} & 0 & * & b_{l-1} \\ 0 & r_{l,l} & * & b_l \\ 0 & 0 & R_{l+1,l+1} & 0 \\ C_{l-1} & 0 & \bar{C}_{l+1} & 0 \end{bmatrix} \\
&= \mathit{rank}\begin{bmatrix} A_{l-1} - \lambda_l I_{l-1} & 0 & * & b_{l-1} \\ 0 & r_{l,l} & * & b_l \\ 0 & 0 & R_{l+1,l+1} & 0 \\ \mathbf{0} & 0 & \bar{C}_{l+1} & 0 \end{bmatrix}\begin{bmatrix} I_{l-1} & 0 & 0 & -C\left(A_{l-1}-\lambda_l I_{l-1}\right)^{-1} \\ 0 & 1 & 0 & r_{l,l}^{-1}b_l \\ 0 & 0 & I_{n-l} & 0 \\ 0 & 0 & 0 & I_p \end{bmatrix} \\
&= \mathit{rank}\begin{bmatrix} A_{l-1} - \lambda_l I_{l-1} & 0 & * & \mathbf{0} \\ 0 & r_{l,l} & * & \mathbf{0} \\ 0 & 0 & R_{l+1,l+1} & 0 \\ \mathbf{0} & 0 & \bar{C}_{l+1} & 0 \end{bmatrix}.
\end{aligned} \tag{4.2.26}$$

This is always possible because the diagonal terms of the upper triangular matrix are all non-zero. After performing the elementary operations, it is clear that the last column of the matrix is 0 thereby implying that the rank is less than $n+1$ which in turn implies that λ_i is a transmission zero of (A, b, C), completing the proof.

Having established the above results, we are in a position to assign the eigenvalues of the state matrix using explicit shifts. However, from the point of view of implementation, it is not very desirable in general because assigning $c-c$ pairs of eigenvalues, we would need to use complex arithmetic. To overcome this limitation, it is preferable to use implicit shifts described in the previous chapter. Since the considerations for forming the implicit shifts are the same for both state as well as output feedback, we will not repeat the details here. The details of forming and employing the implicit (single as well as double-step) shifts in the algorithms are presented in next section.

4.2.2. Theoretical Considerations in EVA for Multi-Input Multi-Output Systems

In this section, we will discuss the case where the number of states of the system is less than the sum of the numbers of inputs and outputs ($n < m + p$), *i.e.*, a sufficient condition required for "almost" arbitrary EVA is satisfied. It will be shown that with the results of Section 4.2.1., the multi-input, multi-output EVA problem can be easily transformed to two single-input, multi-output EVA problems.

In Section 4.2.1., it was shown that if a single-input, p-output system is controllable and observable, we can assign $r\,(\leq p)$ closed-loop eigenvalues arbitrarily close to desired locations in the complex plane. However, a multi-input, multi-output controllable and observable system may not be completely controllable from any one specific input. To overcome this problem, we may need to apply an initial feedback $\hat{K}$ such that the closed-loop system $(A + B\hat{K}C)$ is controllable from one of the inputs (say the first). Having done that, we can apply the results of Section 4.2.1. to the new system and solve the first EVA problem in which we assign $p-1$ eigenvalues. We next show that an initial feedback to make the system controllable from the first input can always be found.

Assume that the given controllable and observable system (A, B, C) with m inputs and p outputs, is in UHF with respect to the first input. If n_c is the dimension of the controllable subsystem $\left(F_{11}, g_1, H_1\right)$, then the following two possibilities can arise:

Case I: $n_c = \text{rank}\left(H_1\right)$. Let us define

$$F = U^T A U \triangleq \begin{bmatrix} F_{11} & F_{12} \\ 0 & F_{22} \end{bmatrix}, \tag{4.2.27a}$$

$$G = U^T B \triangleq \begin{bmatrix} g_{11} & G_{12} \\ 0 & G_{22} \end{bmatrix} \tag{4.2.27b}$$

and

$$H = CU \triangleq \begin{bmatrix} H_1 & H_2 \end{bmatrix} \tag{4.2.27c}$$

where $F_{11} \in \mathbb{R}^{n_c \times n_c}$, $F_{22} \in \mathbb{R}^{(n-n_c)\times(n-n_c)}$ and the matrices G and H are partitioned conformably. From (4.2.27a,b), it is clear that $\left(F, g_{11}\right)$ is not a controllable pair. It should be noted that the system may be completely controllable from the second input, in which case, the EVA algorithm can be applied to $\left(F, g_2, H\right)$. However, we consider here the non-trivial case in which the pairs $\left(F, g_i\right)$, $i=1, \cdots, m$, where g_i corresponds to the i-th column vector of G, are uncontrollable. Assume that the output matrix H is in a lower row echelon form and that F is controllable from the first *two* inputs together. For the sake of illustration, consider a system with $n=6$, $n_c=3$, $m=3$ and $p=4$, such that:

$$F = \begin{bmatrix} * & * & * & * & * & * \\ \circledast & * & * & * & * & * \\ 0 & \circledast & * & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & \circledast & * & * \\ 0 & 0 & 0 & 0 & \circledast & * \end{bmatrix}, \quad G = \begin{bmatrix} \circledast & * & * \\ 0 & * & * \\ 0 & * & * \\ 0 & \circledast & * \\ 0 & 0 & * \\ 0 & 0 & * \end{bmatrix} \text{ and}$$

$$H = \begin{bmatrix} 0 & 0 & 0 & * & * & * \\ 0 & 0 & \circledast & * & * & * \\ 0 & \circledast & * & * & * & * \\ \circledast & * & * & * & * & * \end{bmatrix}$$

where the encircled elements are *non-zero* by assumption. For this system, a feedback of the form $\hat{K} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & * & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$ will always make the (4,3)-rd element of F non-zero *i.e.*, $\left(F_c, g_1, H\right)$ will be in the required UHF where $F_c = F - G\hat{K}H$. Hence the EVA algorithm may be applied to this system.

Case II: $n_c > \text{rank}\left(H_1\right)$. Assume without loss of generality that H has full row rank. The case under consideration will occur under the following conditions:

(a) Rank $\left(H_1\right) = p$ and $\dim\left(F_{11}\right) > p$ *i.e.*, for $n=6$, $n_c=5$ and rank $\left(H_1\right) = 4$, we have

$$F = \begin{bmatrix} * & * & * & * & * & * \\ \circledast & * & * & * & * & * \\ 0 & \circledast & * & * & * & * \\ 0 & 0 & \circledast & * & * & * \\ 0 & 0 & 0 & \circledast & * & * \\ & & & & & \\ 0 & 0 & 0 & 0 & 0 & * \end{bmatrix}, \quad G = \begin{bmatrix} \circledast & * & * \\ 0 & * & * \\ 0 & * & * \\ 0 & * & * \\ 0 & * & * \\ & & \\ 0 & \circledast & * \end{bmatrix}$$

and

$$H = \begin{bmatrix} 0 & 0 & 0 & \circledast & * & * \\ 0 & 0 & \circledast & * & * & * \\ 0 & \circledast & * & * & * & * \\ \circledast & * & * & * & * & * \end{bmatrix}.$$

Then, partitioning the above matrices as in (4.2.27), we find that the triple $\left(F_{11}, g_1, H_1\right)$ meets the requirements for applying the EVA algorithm, hence this occurrence causes no computational difficulty.

(b) Rank $\left(H_1\right) < p$ and the matrix H is in an "echelon" form. The system may have the following structure:

$$F = \begin{bmatrix} * & * & * & & * & * & * \\ \circledast & * & * & & * & * & * \\ 0 & \circledast & * & & * & * & * \\ & & & & & & \\ 0 & 0 & 0 & & * & * & * \\ 0 & 0 & 0 & & \circledast & * & * \\ 0 & 0 & 0 & & 0 & \circledast & * \end{bmatrix}, \quad G = \begin{bmatrix} \circledast & * & * \\ 0 & * & * \\ 0 & * & * \\ & & \\ 0 & \circledast & * \\ 0 & 0 & * \\ 0 & 0 & * \end{bmatrix}$$

and

$$H = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \circledast \\ 0 & 0 & 0 & 0 & \circledast & * \\ 0 & 0 & 0 & \circledast & * & * \\ * & * & * & * & * & * \end{bmatrix}.$$

Then a feedback of the form $\hat{K} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & * \\ 0 & 0 & 0 & 0 \end{bmatrix}$ will "almost always" [2] make the resulting closed-loop system controllable from the first input. As a result of the feedback, the non-zero elements will destroy the upper Hessenberg structure of the state matrix and therefore $F_c = F - G\hat{K}H$ will have to be reduced to an upper Hessenberg form.

It should be noted that if rank $\left(H_1\right) = 0$, *i.e.*, H_1 is a null matrix, then one of the other inputs may be selected to apply the algorithm. Also, in Step II of Algorithm 4.2 given next, when reducing the dual system to its UHF, similar considerations will enable us to get the required unreduced UHF and the algorithm for EVA can be successfully completed.

4.3. ALGORITHMS FOR EVA BY CONSTANT GAIN OUTPUT FEEDBACK

Next we present the algorithms for EVA. The first algorithm is for EVA in single-input, multi-output systems. This algorithm can assign at most p eigenvalues of the closed-loop system. The second algorithm uses the first algorithm to obtain a rank 2 output feedback matrix and assigns $m + p - 1$ eigenvalues of the closed loop system arbitrarily close to the desired locations in the complex plane.

4.3.1. An Algorithm for EVA in Single-Input, Multi-Output Systems

We consider the single-input, multi-output controllable and observable system in (4.2.1) in its UHF. If the desired set of closed-loop eigenvalues contain one or more pairs of *c-c* eigenvalues, then we employ implicit shifts with double steps to avoid the use of complex arithmetic. It is assumed that each desired complex eigenvalue and its conjugate are arranged consecutively. Also, it is assumed without any loss of generality that the output matrix H has full rank, *i.e.*, we can assign p eigenvalues of the closed-loop system arbitrarily close to desired locations in the complex plane. The following algorithm assigns real as well as *c-c* pairs of eigenvalues.

Algorithm 4.1: -(EVA in single-input, multi-output systems using implicit shifts)

Step I: (Initialization)

Set $k^T = k_0^T = 0^T$ $g_1 = g$ $F_1 = F$ $Q = I_n$ $i = 1$ and l = number of eigenvalues to be assigned ($\leq p$)

Step II: (Real Eigenvalues)

If λ_i is complex, go to *Step III*; else set $k_i = 0$ and $T_i = I_p$:

(1) If $i = l = n$, go to Step II-(8); else determine an orthogonal matrix P_i such that

$$\hat{f}_i^T P_i = \pm \| \hat{f}_i \|_2 e_n^T$$

where

$$\hat{f}_i^T = \left[0,0,\cdots,f_{n,n-1},f_{n,n}-\lambda_i \right]$$

is the last row of $F_i - \lambda_i I_n$, $\| \hat{f}_i \|_2 = \left(\hat{f}_i^T \hat{f}_i \right)$ and e_n^T is a vector of length n defined as $[0,0,\cdots,1]$.

Comment : An orthogonal matrix P_i can always be found to accomplish this step [13,14]. The object is to eliminate $f_{n,n-1}$ using $f_{nn} - \lambda_i$.

(2) Set $\tilde{F}_i = P_i^T F_i P_i$, $\tilde{g}_i = P_i^T g_i$ and $\tilde{H}_i = H_i P_i$.

Comment : Note that the transformations represented by P_i are applied to F_i and not to $F_i - \lambda_i I_n$. This is because in forming the matrix P_i the shift has already been accounted for. The matrix $\tilde{F}_i$ has the following structure:

$$\tilde{F}_i = \begin{bmatrix} \tilde{F}_i^{11} & \tilde{F}_i^{12} \\ 0 & \tilde{F}_i^{22} \end{bmatrix} \qquad (4.3.1)$$

where $\tilde{F}_i^{11} \in \mathbb{R}^{(i-1)\times(i-1)}$ is in RSF with its $(i-1)$ eigenvalues appearing along the diagonal at the desired locations. Note that after the similarity transformations P_i have been applied, the matrix $\tilde{F}_i^{22}$ and therefore, $\tilde{F}_i$ is no longer in its upper Hessenberg form since its $(n,n-2)$-th element will be non-zero.

(3) Reduce $\tilde{F}_i$ to an UHM $\hat{F}_i$ by means of plane rotations $P_{i,j}$, $j=1,2, \cdots ,n-i-1$, i.e.,

$$\hat{F} = P^T_{i,n-i-1} \cdots P^T_{i,1}\tilde{F}_i P_{i,1} \cdots P_{1,n-i-1}, \tag{4.3.2a}$$

and transform $\tilde{g}_i$ and $\tilde{H}_i$ as:

$$g_{i+1} = P^T_{i,n-i-1} \cdots P^T_{i,1}\ \tilde{g}_i \tag{4.3.2b}$$

$$\hat{H}_i = \tilde{H}_i P_{i,1} \cdots P_{i,n-i-1}. \tag{4.3.2c}$$

If the i-th column of $\hat{H}_i$ is a zero vector, (λ_i is a transmission zero of the system), go to Step II-(7), else, continue.

Comment : Note that the submatrix $\tilde{F}_i^{11} \in \mathbb{R}^{(i-1)\times(i-1)}$ is not altered by this transformation; only the submatrix $\tilde{F}_{i_{22}}$ is reduced to an unreduced UHF.

(4) Reduce $\hat{H}_i$ to a lower *row-echelon* form.

Comment : Assuming that λ_i is not a transmission zero between the input and the first $(p-i+1)$ outputs of the system $\left(\hat{F}_i, g_{i+1}, \hat{H}_i\right)$, the element $h_{(p-i,i)}$ is non-zero. Note that the transformation T_i does not affect the matrix $\hat{F}_i$ or the vector g_{i+1}.

(5) Determine a feedback vector $\tilde{k}_i^T$ such that the $(i+1,i)$-th element of $\hat{F}_i - g_{i+1}\tilde{k}_i^T H_{i+1}$ becomes zero. A suitable choice of the vector $\tilde{k}_i^T (\in \mathbb{R}^p)$ is $\left[0 \cdots 0\ k_{i,p-i+1}\ 0 \cdots 0\right]$ where the only non-zero element of $\tilde{k}_i^T$ is the $(p-i+1)$-th element.

Comments 1 : From the lower trapezoidal structure of H_{i+1} and knowing g_{i+1}, we have

$$k_{i,p-i+1} = \frac{f_{i+1,i}}{g_{i+1}h_{p-i,i}} \tag{4.3.3}$$

where $f_{i+1,i}$ is the $(i+1,i)$-th element of $\tilde{F}_i$ and g_{i+1} is the $(i+1)$-th element of g_{i+1}.

2 : If the non-zero element of the feedback vector is chosen so as

to reduce the $(i+1,i)$th element of $\hat{F}_i$ to zero, then the (i,i)th element of $\hat{F}_i - g_{i+1}\bar{k}_i^T H_{i+1}$ will be equal to λ_i.

(6) Set $T = T_i T$, $k^T = k^T + \bar{k}_i^T T$, $F_{i+1} = \hat{F}_i - g_{i+1}\bar{k}_i^T H_{i+1}$.
If $i = l$, STOP; else, set $i = i+1$ and go to *Step II.*

(7) Set $F_{i+1} = F_i$, $g_{i+1} = g_i$ and $H_{i+1} = H_i$
If $i = l$, STOP; else, set $i = i+1$ and go to *Step II.*

(8) If the last column vector of H_n is a zero vector (λ_i is a transmission zero of the system), then stop; else, determine a feedback vector $\bar{k}_n^T$ such that the (n,n)th element of $F_n - g_n \bar{k}_n^T H_n$ is equal to λ_n. The vector $\bar{k}_n^T$ will have only the first element $k_{n,1}$ as non-zero and T_n will be the $n \times n$ identity matrix.

Comment : The non-zero element in $\bar{k}_n^T$ is determined as:

$$k_{n,1} = \frac{f_{n,n} - \lambda_n}{g_n h_{1,n}}. \tag{4.3.4}$$

where f_{nn} denotes the (n,n)-th element of F_n; g_n denotes the n-th element of g_n and $h_{1,n}$ denotes the $(1,n)$-th element of H_n. Note that g_n and $h_{1,n}$ are non-zero due to the assumption that the pair (F,g) is controllable and that the rank of $(H) \geq l$ (the number of desired eigenvalues to be assigned).

(9) Set $k^T = k^T + \bar{k}_n^T I_n$ and STOP.

Step III: (Complex-Conjugate pairs of Eigenvalues)

(1) If $i = n-1$, go to Step *III*-(8); else determine an orthogonal matrix P_i such that

$$\hat{f}_i^T P_i = \pm \|\hat{f}_i\|_2 e_n^T$$

where

$$\hat{f}_i^T = [0\ 0 \cdots \hat{f}_{n,n-2}, \hat{f}_{n,n-1}, \hat{f}_{n,n}]$$

is the last row of $(F_i - \lambda_i I_n)(F_i - \lambda_i^* I_n)$, λ_i^* being the complex-conjugate of λ_i. The elements $\hat{f}_{n,n-2}$, $\hat{f}_{n,n-1}$ and $\hat{f}_{n,n}$ are given by

$$\begin{aligned}\hat{f}_{n,n-2} &= f_{n-1,n-2} f_{n,n-1} \\ \hat{f}_{n,n-1} &= f_{n,n-1}\left[f_{n-1,n-1} + f_{n,n} - \left(\lambda_i + \lambda_i^*\right)\right] \\ \hat{f}_{n,n} &= \left(f_{n,n}\right)^2 + f_{n,n-1} f_{n-1,n} - f_{n,n}\left(\lambda_i + \lambda_i^*\right) + \lambda_i \lambda_i^*\end{aligned}$$

where $f_{i,j}$ denotes the (i,j)th element of F_i.

Comment : The elements $\hat{f}_{n,n-2}$, $\hat{f}_{n,n-1}$ and $\hat{f}_{nn}$ are all real since $\lambda_i + \lambda_i^*$ and $\lambda_i \lambda_i^*$ are real. Therefore P_i is real (orthogonal) and can be determined using real arithmetic only.

(2) Set $\tilde{F} = P_i^T F_i P_i$, $\tilde{g}_i = P_i^T g_i$, $\tilde{H} = H_i P_i$ and $\tilde{k}_i = 0$.

Comment : The implicit double shifts corresponding to the desired complex-conjugate pair of eigenvalues have been accounted for in constructing the orthogonal matrix P_i. The shifts are, therefore, applied to F_i. The matrix $\tilde{F}_i$ may be written as:

$$\tilde{F}_i = \begin{bmatrix} \tilde{F}_i^{11} & \tilde{F}_i^{12} \\ 0 & \tilde{F}_i^{22} \end{bmatrix}$$

where $\tilde{F}_i^{11} \in \mathbb{R}^{(i-1)\times(i-1)}$ is in RSF with its eigenvalues appearing along the diagonal at desired locations in the complex plane, $\tilde{F}_i^{12} \in \in \mathbb{R}^{(i-1)\times(n-i+1)}$ has no specific structure. Here again, due to the similarity transformation P_i, the matrix $\tilde{F}_i^{22}$ and therefore $\tilde{F}_i$ is no longer an upper Hessenberg matrix because the elements $(n-1, n-3)$, $(n, n-3)$ and $(n, n-2)$ are non-zero.

(3) Apply plane rotations $P_{i,j}$, $j = 1, 2, \cdots, 3(n-i-1)$ in order to make $\tilde{F}_{22}^{(i)}$ as close to upper Hessenberg as possible, i.e.,

$$\hat{F} = P_{i,3(n-i-1)}^T \cdots P_{i,1}^T \tilde{F}_i P_{i,1} \cdots P_{1,3(n-i-1)}. \tag{4.3.5a}$$

Also, let

$$g_{i+2} = P_{i,3(n-i-1)}^T \cdots P_{i,1}^T \tilde{g}_i \tag{4.3.5b}$$

and

$$\hat{H}_{i+2} = \tilde{H}_i P_{i,1} \cdots P_{i,3(n-i+1)}. \tag{4.3.5c}$$

If the i-th and $(i+1)$-th columns of $\hat{H}_{i+2}$ are zero vectors (λ_i and λ_i^* are *c-c* transmission zeros of the system), go to *Step III*-(7), else, continue.

Comment : Note that the submatrix $\tilde{F}_{11}^{(i)}$ is not affected by the above transformations, further, the submatrix $\tilde{F}_{22}^{(i)}$ cannot be reduced to the UHF by the transformations described in the above step.

(4) Reduce $\hat{H}_i$ to a lower *ɽow-echelon* form *i.e.*,

$$H_{i+2} = T_i \hat{H}_i . \tag{4.3.6}$$

Comment : Assuming that λ_i, λ_i^* are not transmission zeros between the input and the first $(p-i+1)$ outputs of the system $\left(\hat{F}_i, g_{i+1}, \hat{H}_i\right)$, the $(p-i,i)$-th and $(p-i-1,i+1)$-th elements of the matrix H_{i+1} are non-zero.

(5) Determine a feedback vector $\tilde{k}_i^T$ such that the $(i+2,i)$-th and $(i+2,i+1)$-th elements of $\hat{F}_i - g_{i+1} k_i^T H_{i+2}$ are eliminated. The vector $\tilde{k}_i^T$ ($\in \mathbb{R}^p$) is given by

$$\tilde{k}_i^T = \begin{bmatrix} 0\,0 & \cdots & 0 \,|\, k_{i,p-i} \, k_{i,p-i+1} \, 0 \cdots 0 \end{bmatrix}.$$

Comment 1 : The non-zero elements of the feedback vector are computed as

$$k_{i,p-i} = \frac{f_{i+2,i}}{g_{i+2} h_{p-i,i}}$$

and

$$k_{i,p-i+1} = \frac{f_{i+2,i+1}}{g_{i+2} h_{p-i+1,i+1}}$$

2 : The feedback of the kind described above results in a 2×2 matrix in the i-th and $(i+1)$-th rows and columns of the closed-loop matrix $\hat{F}_i - g_{i+2} \tilde{k}_i^T H_{i+2}$ with eigenvalues λ_i and λ_i^*.

(6) Set $T = T_i T$, $k^T = k^T + \tilde{k}_i^T T$ and $F_{i+2} = \hat{F}_i - g_{i+2} \tilde{k}_i^T H_{i+2}$.

If $i = l$, STOP; else, set $i = i+2$ and go to *Step II*.

Comment : In this step, we increment i by 2 because we have assigned two eigenvalues.

(7) Set $F_{i+2} = F_i$, $g_{i+2} = g_i$ and $H_{i+2} = H_i$.

If $i = l-1$, STOP; else, set $i = i+2$ and go to *Step II*.

(8) If the last two columns of H_{n-1} are zero vectors (λ_{n-1} and λ_{n-1}^* are transmission zeros of the system), then STOP; else, determine a feedback vector $\bar{k}_{n-1}^T$ such that the 2×2 matrix in the last two rows and columns of $F_{n-1} - g_{n-1}\bar{k}_{n-1}^T H_{n-1}$ has the desired *c-c* pair of eigenvalues at λ_i and λ_i^*.

Comment : The vector $\bar{k}_{n-1}^T$ will be a vector of length n with only the last two elements being nonzero. The structure of the last two rows and columns of $F_{n-1} - g_{n-1}k_{n-1}^T H_{n-1}$ is given by

$$\begin{bmatrix} \hat{f}_{n-1,n-1} & \hat{f}_{n-1,n} \\ f_{n,n-1} & f_{n,n} \end{bmatrix} - \begin{bmatrix} g_{n-1} \\ 0 \end{bmatrix} [k_{n-1,1} \ \ k_{n-1,2}] \begin{bmatrix} 0 & h_{1,n} \\ h_{2,n-1} & h_{2,n} \end{bmatrix}, \tag{4.3.7}$$

The effect of applying the feedback k_{n-1}^T is to change the first row of the 2×2 matrix above so that by the appropriate choice of the two nonzero elements of the feedback vector, we can ensure that the 2×2 matrix in (4.3.7) has eigenvalues at λ_{n-1} and λ_{n-1}^*.

(9) Set

$$k^T = k^T + \bar{k}_{n-1}^T I_n$$

and STOP.

Remarks about the Algorithm

1. Upon completion of Algorithm 4.1, we get the required feedback vector that assigns l eigenvalues of the closed-loop system arbitrarily close to some desired locations in the complex plane (subject to complex-conjugate pairing). It should be noted that an eigenvalue can be assigned at the desired location if it does not correspond to a transmission zero of the system.

2. Algorithm 4.1 assigns real as well as *c-c* pairs of eigenvalues using real arithmetic only. A single step implicit shift is used to assign a real eigenvalue and one element of the feedback vector is determined. A double step implicit shift is used to assign a *c-c*

pair of eigenvalues and in this step two elements of the feedback vector are determined.

3. As shown in Step II-(8) and Step III-(8), if $i = l = n$ for a real eigenvalue or $i = l - 1 = n - 1$ for a *c-c* pair of eigenvalues, then the eigenvalues are assigned directly since we cannot form implicit shifts in these cases. The feedback gain required to assign the last real eigenvalue directly is given by (4.3.4). The gains for the last *c-c* pair of eigenvalues may be determined as given below. We write (4.3.7) as

$$\begin{bmatrix} \hat{\phi}_{11} & \hat{\phi}_{12} \\ \phi_{21} & \phi_{22} \end{bmatrix} - \begin{bmatrix} \gamma_1 \\ 0 \end{bmatrix} [\kappa_1 \quad \kappa_2] \begin{bmatrix} 0 & \theta_{12} \\ \theta_{21} & \theta_{22} \end{bmatrix}$$

where κ_1 and κ_2 must be determined such that the resulting matrix has the desired eigenvalues at λ_{n-1} and λ^*_{n-1}. The elements κ_1 and κ_2 are given by

$$\kappa_1 = \frac{1}{\phi_{21}\gamma_1\theta_{12}} \left[\lambda_{n-1}\lambda^*_{n-1} + \hat{\phi}_{12}\phi_{21} + \phi_{22}^2 - \phi_{22}\left(\lambda_{n-1} + \lambda^*_{n-1}\right) \right] - \frac{\theta_{22}}{\gamma_1\theta_{12}\theta_{21}} \left[\hat{\phi}_{11} + \phi_{22} - \left(\lambda_{n-1} + \lambda^*_{n-1}\right) \right] \tag{4.3.8}$$

and

$$\kappa_2 = \frac{1}{\gamma_1\theta_{12}} \left[\hat{\phi}_{11} + \phi_{22} - \left(\lambda_{n-1} + \lambda^*_{n-1}\right) \right]. \tag{4.3.9}$$

By the assumptions that H has full rank and that the system is controllable and observable, the elements ϕ_{21}, γ_1, θ_{12} and θ_{21} are non-zero, hence, finite feedback gains κ_1 and κ_2 can always be found. Moreover, from (4.3.8) and (4.3.9), it is clear that the feedback gains can be computed using real arithmetic only.

4.3.2. An Algorithm for EVA in Multi-Input, Multi-Output Systems

We next consider the system triples (F, G, H) defining the state equations of a general multi-input, multi-output system. We assume that the sufficient condition $m + p > n$ is satisfied. It is assumed further that the given system is controllable and observable, and that if the system is not controllable from the first input, then, using a preliminary output feedback, it has been made controllable from the first input. The proposed method is a two step method. The first step assigns $p - 1$ eigenvalues using the first input. The compensated system is in block upper triangular form. To assign the remaining $n-p+1$ eigenvalue, we consider the dual system. The transformations on the dual system as well as the output feedback applied to it is such that the previusly assigned eigenvalues are not affected. The algorithm of the previous section is applied to assign all n eigenvalues arbitrary close to desired locations as follows:

Algorithm 4.2: (EVA in multi-input multi-output systems using implicit shifts)

Step I: (Assign first p-1 eigenvalues)

(1) Set $l = p-1$, where l is the number of eigenvalues to be assigned ($<n$) and p is the rank of H.

(2) Reduce $\left(F, g_1, H\right)$ to its UHF and apply Algorithm 4.1.

Comment : At the end of *Step I*, the matrices will have the following structures:

$$F_{p-1} \triangleq \begin{bmatrix} F_{p-1}^{11} & F_{p-1}^{12} \\ 0 & F_{p-1}^{22} \end{bmatrix}$$

where $F_{p-1}^{11} \in \mathbb{R}^{(p-1)\times(p-1)}$ is a real Schur matrix with $p-1$ eigenvalues at desired locations, $F_{p-1}^{12} \in \mathbb{R}^{(p-1)\times(n-p+1)}$ has no specific form, $F_{p-1}^{22} \in \mathbb{R}^{(n-p+1)\times(n-p+1)}$ is an unreduced upper Hessenberg matrix and matrices $G_{p-1} \triangleq \begin{bmatrix} G_{p-1}^{1} \\ G_{p-1}^{2} \end{bmatrix}$ and $H_{p-1} \triangleq \begin{bmatrix} H_{p-1}^{1} & H_{p-1}^{2} \end{bmatrix}$ are conformably partitioned. The feedback vector $k^T \in \mathbb{R}^p$ is given by

$$k^T = \begin{bmatrix} 0 & k_{p-1}^T \end{bmatrix}.$$

Step II: (Form the dual system)

(1) Form the dual system *i.e.*, set $F = \left(F_{p-1}^{22}\right)^T$, $G = \left(H_{p-1}^{2}\right)^T$, $H = \left(G_{p-1}^{2}\right)^T$, $S = T^T$, $K = [\, k \;\; 0\,]$, $K \in \mathbb{R}^{p\times m}$, $n = n - p + 1$ and $p = m$.

Comment : Note that the system (F, G, H) so formed is not in UHF any more.

Step III: (Assign remaining eigenvalues)

(1) Set $l = n - p + 1$ = the number of remaining eigenvalues to be assigned.

(2) Reduce the single-input, multi-output system (F, g_1, H) to an UHF and apply Algorithm 4.1 to the system to get an output feedback vector k_i^T.

(3) Define the feedback matrix as:

$$K = K + S \begin{bmatrix} k_i^T \\ 0 \end{bmatrix} T \tag{4.3.10}$$

Comment : Step III (3) is required to account for the fact that the system is now a multi-input, multi-output system and therefore a feedback *matrix* must be computed for EVA. The feedback matrix for the "primal" system is given by K^T.

4.4. EVA Assignment by Dynamic Output Feedback

Analytically, the problem of dynamic output feedback can be reduced to that of EVA by constant output feedback for an augmented system [1]. For a given controllable and observable linear multivariable system

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{4.4.1}$$

$$y(t) = Cx(t), \tag{4.4.2}$$

dynamic output feedback can be defined by

$$u(t) = v(t) - Hz(t) - Dy(t) \tag{4.4.3}$$

$$\dot{z}(t) = Fz + Gy(t) \tag{4.4.4}$$

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, $y(t) \in \mathbb{R}^p$, $z(t) \in \mathbb{R}^r$, $v(t) \in \mathbb{R}^m$ and r is the order of the dynamic compensator that assigns all the eigenvalues of the resulting closed-loop system arbitrarily close to $n+r$ desired locations. The closed-loop state matrix is given by

$$\hat{A} = \begin{bmatrix} A - BDC & -BH \\ GC & F \end{bmatrix} \tag{4.4.5}$$

$$= \begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} B & 0 \\ 0 & -I \end{bmatrix} \begin{bmatrix} D & H \\ G & F \end{bmatrix} \begin{bmatrix} C & 0 \\ 0 & I \end{bmatrix}. \tag{4.4.6}$$

From (4.4.6), we note that the same closed-loop state matrix would result if we were to apply a constant gain output feedback $\mu = \nu - \begin{bmatrix} D & H \\ G & F \end{bmatrix} \xi$, to the system

$$\dot{\phi} = \begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix} \phi + \begin{bmatrix} B & 0 \\ 0 & -I \end{bmatrix} \mu \tag{4.4.7a}$$

$$\xi = \begin{bmatrix} C & 0 \\ 0 & I \end{bmatrix} \phi . \tag{4.4.7b}$$

It should be noted that the matrices D, H, G and F define the required dynamic output feedback completely. Unfortunately, we cannot directly use the algorithms described in the previous sections. This is because of the fact that in the augmented system, any plane rotations applied to the lower $r \times r$ zero block will be trivial and as a result, we cannot perform implicit shifts required for the EVA. However, we can reformulate the problem so that we can use the algorithms described in the preceding sections.

We consider the augmented system triple $(\tilde{A}, \tilde{B}, \tilde{C})$ such that,

$$\tilde{A} = \begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix}, \ \tilde{B} = \begin{bmatrix} B & 0 \\ 0 & -I \end{bmatrix}, \text{ and } \tilde{C} = \begin{bmatrix} C & 0 \\ 0 & I \end{bmatrix}. \tag{4.4.8}$$

Also, we assume that (A, b_1) is in UHF and, for the sake of illustration, we assume that it is a controllable pair. It is obvious from equation (4.4.7) that the last r states are uncontrollable from the first m inputs of the augmented system. Consider a constant gain feedback matrix $\tilde{K}$ with the following structure:

$$\tilde{K} = \begin{array}{c} \\ m \\ \\ \\ r \\ \\ \\ \end{array} \left[\begin{array}{ccccccc} 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ \vdots & & \vdots & \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & 1 & \cdots & 0 & 0 \\ \vdots & & \vdots & \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 1 & 0 \\ & p-1 & & & r & & 1 \end{array} \right] P_i . \tag{4.4.9}$$

In (4.4.9), P_i is a permutation matrix that can be chosen to interchange the i-th and p-th rows of $\tilde{C}$. Then, we can establish the following result:

Theorem 4.4 : The single-input pair $(\tilde{A} - \tilde{B} \tilde{K} \tilde{C}, \tilde{b}_1)$ is controllable if and only if the open-loop system (A, b_1, C) does not have a transmission zero at the origin [11,16].

Proof : Assume that the system $(A\,;B\,,C)$ was in UHF with respect to the first input. Further, assume that we selected P_i to interchange the i-th row of C denoted by c_i^T with the p-th row. Then, the closed-loop system $\left(\bar{A}\,,\bar{B}\,,\bar{C}\right)$ has the following structure:

$$\bar{A} = \begin{bmatrix} \bar{A}_{11} & 0 \\ \bar{A}_{21}\,, & \bar{A}_{22} \end{bmatrix} \tag{4.4.10a}$$

where $\bar{A}_{11} \in \mathbb{R}^{n \times n}$ is an unreduced UHM, $\bar{A}_{21}\ (\in \mathbb{R}^{r \times n}\ ) = \begin{bmatrix} c_i^T \\ 0 \end{bmatrix}$ and $\bar{A}_{22}$ $(\in \mathbb{R}^{r \times r}) = \begin{bmatrix} 0 & 0 \\ -I_r & 0 \end{bmatrix}$

$$\bar{B} = \begin{bmatrix} \bar{b}_1 & \bar{B}_{12} & 0 \\ & 0 & -I_r \end{bmatrix} \tag{4.4.10b}$$

and

$$\bar{C} = \begin{bmatrix} C & 0 \\ 0 & I_r \end{bmatrix} \tag{4.4.10c}$$

Next, consider the single-input, single-output system $\left(A\,,b_1,c_i^T\right)$, where $c_i^T = \left[c_{i,1}\ c_{i,2} \cdots c_{i,n}\right]$. If we apply plane rotations with a real shift of 0 to the matrix A as described in Lemma 3.1 then by Lemma 3.3, $\lambda = 0$ is a transmissin zero of the system $\left(A\,,b_1,c_i^T\right)$ if and only if $c_{i,1} = 0$.

We shall prove the theorem by showing that the system $\left(\bar{A}\,,\bar{b}_1,\bar{C}\right)$ has no input-decoupling zero if and only if $\left(A\,,b_1,c_i^T\right)$ has no transmission zero at the origin. From (4.4.10a), we see that the eigenvalues of $\bar{A}$ are the eigenvalues of A together with r eigenvalues at the origin. From the structure of $\bar{A}$ and $\bar{b}_1$ and the assumption that $(A\,,b_1)$ is a controllable pair, it is clear that the eigenvalues of A are not input-decoupling zeros of $\left(\bar{A}\,,\bar{b}_1,\bar{C}\right)$. Now, consider the eigenvalues at the origin. Let plane rotation with a real shift of 0 be applied to the submatrix $\bar{A}_{11}$. Then, we may say that

$$\text{rank}\begin{bmatrix}\tilde{A}_{12} & 0 & b_1 \\ \tilde{A}_{21} & \tilde{A}_{22} & 0\end{bmatrix} = \begin{bmatrix} * & * & \cdots & * & * & 0 & \cdots & 0 & 0 & b_{11} \\ 0 & * & \cdots & * & * & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & * & * & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & \cdots & 0 & * & 0 & \cdots & 0 & 0 & 0 \\ c_{i,1} & c_{i,2} & \cdots & c_{i,n-1} & c_{i,n} & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & -1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & -1 & 0 & 0 \end{bmatrix} \tag{4.4.11}$$

By re-arranging the rows on the right hand side of the above equation, it is easy to see that the rank of the matrix is $(n+r)$ if and only if $c_{i,1} \neq 0$ which, as shown earlier, holds if and only if the system $\left(A, b_1, c_i^T\right)$ does not have any transmission zeros at the origin. By specifying P_i appropriately, c_i^T can be selected to be any row of C. Therefore, the results holds if and only if $\left(A, b_1, C\right)$ does not have any transmission zeros at the origin, thus completing the proof.

Now, since all the conditions for applying the algorithms for EVA by constant gain output feedback have been met, we can proceed as in the previous section to compute a feedback matrix K_1 for the augmented system. The matrix K_1 can then be partitioned as in (4.4.6) to get the required dynamic output feedback equations (4.4.3) and (4.4.4).

Remarks : 1. The modifications made to Algorithm 4.2 for dynamic output feedback case do not affect the properties of the algorithm. However, the algorithm is now applied to a higher order system and therefore, requires more computations.

2. It is necessary that the open-loop system $\left(A, b_1, C\right)$ not have a transmission zero at the origin. Otherwise, it will be cancelled by a pole of the system $\left(\tilde{A}, \tilde{b}_1, \tilde{C}\right)$ at the origin, making it impossible for the augmented system to become controllable from the first input. This, however, is not a restriction because if there is a transmission zero at the origin in $\left(A, b_1, C\right)$, then we can rewrite equation (4.4.6) as

$$\hat{A} = \begin{bmatrix} A & 0 \\ 0 & Y \end{bmatrix} - \begin{bmatrix} B & 0 \\ 0 & -I \end{bmatrix} \begin{bmatrix} D & H \\ G & F-Y \end{bmatrix} \begin{bmatrix} C & 0 \\ 0 & I \end{bmatrix}$$
$$= \begin{bmatrix} A - BDC & -BH \\ GC & F \end{bmatrix}$$

where we can select Y such that the matrix $\begin{bmatrix} A & 0 \\ 0 & Y \end{bmatrix}$ has no eigenvalues at the transmission zeros of (A, b_1, C). Note that the assumption that (A, b_1, C) is controllable and observable implies that there are no cancellations of transmission zeros by poles in (A, b_1, C) [15].

3. It can be checked in a numerically stable manner [16], whether or not, the single-input, multi-output system (A, b_1, C) has a transmission zero at a particular location λ.

4.5. NUMERICAL EXAMPLES

In this section, we illustrate the performance of the algorithms described in this chapter by means of some numerical examples. Here again, the desired closed-loop eigenvalues have been selected for the purpose of illustration only and not to meet any specific design criteria.

Example 1 : For this example, we have taken the data for the 16-th order F100 Turbofan engine whose parameters are given in Table 3.1. Since the system has only 5 inputs and 5 outputs, we can assign $m + p - 1$ $(= 9)$ eigenvalues by means of constant gain output feedback matrix. The desired and the computed eigenvalues of the closed-loop system are given in Table 4.1.

Example 2 : We have selected this example to illustrate EVA algorithm using dynamic output feedback. The system being considered is the 9-th order model of a drumboiler with 2 inputs and 2 outputs [17]. The matrices corresponding to the system triple (A, B, C) are given in Table. 4.2. A dynamic compensator of 6-th order enabled us to assign all the eigenvalues of the augmented system at the desired locations. The parameters of the dynamic compensator are given in Table 4.3. The desired and the computed closed-loop eigenvalues of the augmented system are shown in Table 4.4.

Desired c-l ev's	Computed c-l ev's
-5.75E+02	-5.750000000000001E+02
-1.75E+02	-1.750000000000000E+02
-5.90E+02	-5.900000000000000E+01
-5.00E+02	-5.000000000000001E+01
-2.15E+01	-2.149999999999997E+01
±8.10E+00i	±8.099999999999961E+00i
-1.85E+01	-1.850000000000003E+01
-1.90E+00	-1.899999999999981E+00
-2.60E+00	-2.599999999999996E+00
Don't Care	-4.684197423663909E+01
Don't Care	-3.389256478290264E+01
Don't Care	-2.081856928342824E+01
Don't Care	-2.048046790579636E+01
Don't Care	-7.876749294732868E+00
Don't Care	-6.564310754793838E-01
Don't Care	-2.028103998833246E+00

Table 4.1 Desired and closed loop eigenvalues of F100 Turbo Fan Engine

$$\left[\begin{array}{cccccc}
-3.9300E+00 & -3.1500E-03 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 4.0300E-05 \\
3.6800E+02 & -3.0500E+00 & 3.0300E+00 & 0.0000E+00 & 0.0000E+00 & -3.7700E-03 \\
2.7400E+01 & 7.8700E-02 & -5.9600E-02 & 0.0000E+00 & 0.0000E+00 & -2.8100E-04 \\
-6.4700E-02 & -5.2000E-05 & 0.0000E+00 & -2.5500E-01 & -3.3500E-06 & 3.6000E-07 \\
3.8500E+03 & 1.7300E+01 & -1.2800E+01 & -1.2600E+04 & -2.9100E+00 & -1.0500E-01 \\
2.2400E+04 & 1.8000E+01 & -0.0000E+00 & -3.5600E+01 & -1.0400E-04 & -4.1400E-01 \\
0.0000E+00 & 0.0000E+00 & 2.3400E-03 & 0.0000E+00 & 0.0000E+00 & 2.2200E-04 \\
0.0000E+00 & 0.0000E+00 & 0.0000E+00 & -1.2700E+00 & 1.0000E-03 & 7.8600E-05 \\
-2.2000E+00 & -1.7700E-03 & 0.0000E+00 & -8.4400E+00 & -1.1100E-04 & 1.3800E-05
\end{array}\right.$$

$$\left.\begin{array}{ccc}
0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\
6.3300E-05 & 1.9400E-04 & 0.0000E+00 \\
1.2700E+01 & 4.3100E+01 & 0.0000E+00 \\
9.0000E+01 & 5.6900E+01 & 0.0000E+00 \\
-2.0300E-01 & 0.0000E+00 & 0.0000E+00 \\
0.0000E+00 & -7.1700E-02 & 0.0000E+00 \\
1.4900E-03 & 6.0200E-03 & -1.0000E-10
\end{array}\right]$$

Table 4.2a State Matrix A for Drumboiler model

$$\begin{bmatrix} 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 \\ 1.5600E+00 & 0.0000E+00 \\ 0.0000E+00 & -5.1300E-06 \\ 8.2800E+00 & -1.5500E+00 \\ 0.0000E+00 & 1.7800E+00 \\ 2.3300E+00 & 0.0000E+00 \\ 0.0000E+00 & -2.4500E-02 \\ 0.0000E+00 & 2.9400E-05 \end{bmatrix}$$

Table 4.2b Input Matrix $\bar{B}$ for Drumboiler model

$$\left[\begin{matrix} 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 1.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \end{matrix}\right.$$

$$\left.\begin{matrix} 0.0000E+00 & 0.0000E+00 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 1.0000E+00 \end{matrix}\right]$$

Table 4.2c Output Matrix C for Drumboiler model

$$\begin{bmatrix} 2.3160E+01 & 1.6449E+01 & 1.2034E+01 & 1.7971E+01 & 1.1613E+01 & 2.0405E+01 \\ -1.5240E+02 & -1.2805E+02 & -8.2453E+01 & -9.2432E+01 & -7.6723E+01 & -1.9327E+02 \\ -2.0101E+01 & -1.5874E+01 & -1.3908E+01 & -1.2750E+01 & -1.1868E+01 & -2.4646E+01 \\ 7.1147E+01 & 5.6522E+01 & 4.0941E+01 & 4.3582E+01 & 3.4573E+01 & 8.3770E+01 \\ -2.9681E+00 & 3.0371E+00 & 4.5541E-01 & -1.5071E+00 & 4.7027E+00 & 6.7136E+00 \\ 4.1081E+01 & 3.7004E+01 & 2.7429E+01 & 3.1202E+01 & 2.8934E+01 & 4.9051E+01 \end{bmatrix}$$

Table 4.3a Dynamic Compensator Parameter F in (4.4.6)

$$\begin{bmatrix} 2.0754E+00 & 5.1807E+02 \\ -3.2803E+01 & -3.9781E+01 \end{bmatrix}$$

Table 4.3b Dynamic Compensator Parameter D in (4.4.6)

$$\begin{bmatrix} -3.1480E+02 & 4.5600E+01 & -3.7041E+01 & 1.0065E+03 & 7.9203E+02 & -1.3638E+03 \\ -4.4511E+01 & -3.5509E+01 & -1.9218E+01 & -2.5919E+01 & -2.0249E+01 & -4.5006E+01 \end{bmatrix}$$

Table 4.3c Dynamic Compensator Parameter H in (4.4.6)

$$\begin{bmatrix} 9.3672E+00 & 1.6721E+01 \\ -1.4472E+02 & -1.2939E+02 \\ -1.3142E+01 & -1.0545E+01 \\ 6.1686E+01 & 5.2051E+01 \\ 7.4978E+00 & 2.8774E+00 \\ 4.3265E+01 & 3.3970E+01 \end{bmatrix}.$$

Table 4 3d Dynamic Compensator Parameter C in (4.4.6)

Desired c-l e.v.'s	Computed c-l e.v.'s of Augmented System
-2.100E+01	-2.099999999973079E+01
-5.000E+00	-5.000000006538122E+00
-4.500E+00	-4.500000002309388E+00
-4.000E+00	-3.999999957689151E+00
-3.500E+00	-3.500000040624153E+00
-3.000E+00	-2.999999999675663E+00
-2.500E+00	-2.500000069781103E+00
-2.000E+00	-1.999999906418416E+00
-1.000E+00	-9.999999511102389E-01
-5.500E-01	-5.500001740466275E-01
-4.000E-01	-3.999998888165926E-01
-2.000E-01	-2.000000037274676E-01
-1.300E-01	-1.299999967565668E-01
-2.000E-02	-1.999998996023493E-02
-2.400E-02	-2.400001281753963E-02

Table 4.4 Desired and Computed closed-loop e.v.'s of Augmented System.

4.6. CONCLUDING REMARKS

In this chapter, we presented numerical algorithms for eigenvalue assignment by means of constant gain as well as dynamic output feedback. To the best of our knowledge, this is the first numerically reliable, direct approach for these problems where significant attention has been given to numerical issues, in particular to the reliability of the numerical computations involved. The problem was treated as a converse of the algebraic eigenvalue problem, the underlying principle being the QR decomposition of a matrix and the use of "implicit" shifts. In the eigenvalue computation problem, the shifts converge to the true eigenvalues while in the eigenvalue assignment problem, the shifts are known *a priori*, being the desired closed-loop eigenvalues. The basic idea is to use the output feedback to modify the given state matrix so that it has an eigenvalue corresponding to the specified shift. The algorithms use only orthogonal similarity transformations together with output feedback to assign the desired eigenvalues and therefore, have good numerical properties. The performance of the algorithms has been illustrated by numerical examples.

Apart from presenting a systematic procedure for assigning the eigenvalues of a system by output feedback, the chapter has also established several theoretical results. The proofs for these results are constructive in nature and involve the use of condensed forms which can be computed reliably. The algorithms also serve the purpose of bringing several existing theoretical results for eigenvalue assignment by output feedback under a common framework.

The problem of EVA by means of dynamic output feedback appears to be inherently ill-conditioned. It was observed that in general, the elements of the dynamic output feedback compensator tend to be large. This problem needs further investigation. It should be pointed out that for the algorithms proposed in this chapter, we have used rank 2 output feedback. It would be interesting to see whether the conditioning of the problem can be improved with higher rank output feedback Similar to the case of EVA by means of state feedback, the gains depend on the order in which the eigenvalues are assigned. A further investigation in that direction would be of interest for EVA by output feedback as well.

Similar to EVA by state feedback, the conditioning of the problem of EVA by output feedback is implicitly related to the controllability and observability properties of the open-loop system. Further effort along establishing an explicit relation between the conditioning of the EVA problem with controllability and observability of the open-loop system may lead to a better understanding of the problem and may yield a better computational algorithm.

4.7. References

[1] H. Kimura, "Pole Assignment by Gain Output Feedback", *IEEE Trans. Automat. Contr.*, Vol. AC-20, pp. 509-516, 1975.

[2] E.J. Davison and S.H. Wang, "On Pole Assignment in Linear Multivariable Systems", *IEEE Trans. Automat. Contr.*, Vol. AC-20, pp. 516-518, 1975.

[3] G.S. Miminis and C.C. Paige, "An Algorithm for Pole-Assignment of Time-Invariant Linear Systems", *Int. J. Contr.*, Vol. 35, pp. 341-354, 1982.

[4] R.V. Patel and P. Misra, "Numerical Algorithms for Eigenvalue Assignment by State Feedback", *Proc. IEEE*, Vol. 72, pp. 1755-1764, 1984.

[5] J. Kautsky, N.K. Nichols, P. VanDooren and L. Fletcher, "Robust Pole Assignment in Linear Feedback", *Int. J. Contr.*, Vol. 41, pp. 1129-1155, 1985.

[6] F. Fallside, Ed., *Control Systems Design by Pole-Zero Assignment*, Academic Press, London, 1977.

[7] W.M. Wonham, *Linear Multivariable Control: A Geometric Approach*, Springer Verlag, N.Y., 1979.

[8] T. Kailath, *Linear Systems*, Prentice Hall, Englewood Cliffs, N.J., 1980.

[9] R.V. Patel and N. Munro, *Multivariable System Theory and Design*, Pergamon Press, Oxford, 1982.

[10] P. Misra and R.V. Patel, "Numerical Considerations in Eigenvalue Assignment by Output Feedback", *24th IEEE Conf. on Dec. and Contr.*, pp. 315-316, Ft. Lauderdale, FL, 1985.

[11] R.V. Patel and P. Misra, "A Numerical Algorithm for Eigenvalue Assignment by Output Feedback", in *Computational and Combinatorial Methods in Linear Systems*, C. Byrnes and A. Lindquist, Eds., 1986.

[12] P. Misra and R.V. Patel, "Numerical Algorithms for Eigenvalue Assignment by Constant and Dynamic Output Feedback", to appear in *IEEE Trans. Automat. Contr.*

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

[14] G.W. Stewart, *Introduction to Matrix Computations*, Academic Press, N.Y., 1973.

[15] R.V. Patel, "On Blocking Zeros in Linear Multivariable Systems", *IEEE Trans. Automat. Contr.*, Vol. 33, pp. 239-241, 1986.

[16] R.V. Patel and P. Misra, "A Numerical Test for Transmission Zeros with Applications in Characterizing Decentralized Fixed Modes", *23rd IEEE Contr. and Dec. Conf.*, pp. 1148-1153, Las Vegas, 1984.

[17] E.J. Davison and S.H. Wang, "Properties and Calculation of Transmission Zeros of Linear Multivariable Systems", *IEEE Trans. Automat. Contr.*, Vol. 10, pp. 643-648, 1974.

CHAPTER V

CONTROL OF DECENTRALIZED MULTIVARIABLE SYSTEMS

An interesting application of the results stated in the previous chapter is in the control of "decentralized systems" [1-5]. The problem of stabilization of decentralized systems is considerably more complex than that for "centralized" systems considered earlier. Fundamental to the problem of decentralized stabilization is the concept of "fixed modes". It is well known [3], that a given controllable and observable system can be stabilized by means of a decentralized feedback structure if and only if the system does not have any unstable fixed modes

In this chapter, we examine the conditions under which fixed modes exist. An algebraic characterization of fixed modes of a decentralized linear multivariable systems is presented. It is shown that fixed modes are related to "blocking zeros" [6] of certain subsystems derived from the given decentralized system. A numerically stable algorithm is then presented which enables us to compute fixed modes in a reliable and computationally efficient manner. The algorithm uses numerically efficient and reliable algorithms available in software packages for numerical linear algebra [7-9] and hence can be easily implemented. Next, under the assumption that the given system has no fixed modes, we examine the decentralized eigenvalue assignment problem.

5.1. INTRODUCTION

In recent years there has been considerable interest in the study of decentralized control of large scale linear multivariable systems such as those which arise in developing control strategies for large flexible space structures [1] or multi-machine power systems [2]. The decentralized structure of these systems is a consequence of the constraints that are imposed on the information and control flow within the system, usually because of the locations of various sensors and actuators. By judiciously locating these sensors and actuators, a structure can be chosen for a decentralized controller which makes it considerably simpler to implement than a "centralized" controller.

The structure of a decentralized controller is an important issue in the control of large-scale systems. This is because of the existence of "decentralized fixed modes" (d.f.m.'s) [1-5]. D.f.m's are those modes of the system which are invariant under the implementation of all decentralized controllers having a particular structure. Therefore, if a d.f.m. corresponding to a particular decentralized structure is unstable or has other undesirable characteristics, the decentralized controller will not be able to remedy the situation. One aspect of the design problem, therefore, is to develop methods of determining a structure for a decentralized controller such that there are no d.f.m.'s or no undesirable d.f.m.'s. Consequently, it is of interest to investigate the conditions under which these modes occur, and develop a numerically efficient and reliable method for computing them.

In recent years, several researchers have obtained different characterizations of d.f.m.'s [10-18,26]. Several of these references provide characterizations in terms of transmission zeros of certain subsystems of the given system. The determination of d.f.m.'s by these approaches can be computationally expensive for systems having high order and/or a large number of "stations", since many transmission zero computation tests would be required. In [13], a transfer function characterization is presented. However, it does not provide an efficient and numerically reliable method by which d.f.m.'s may be computed. In [10], the authors give an algebraic characterization which provides valuable insight into the properties of d.f.m.'s and conditions under which they occur. The characterization requires the partitioning of the set of stations into two disjoint subsets and involves a rank test, but as will be discussed later, a direct application of the result to find d.f.m.'s can be computationally expensive. A geometric characterization of d.f.m.'s has been given in [11]. The development of a numerically *reliable* computational technique from this characterization is by no means a trivial task.

One of the most straightforward ways of computing decentralized fixed modes is the method suggested in [3]. This method gives the fixed modes as those eigenvalues of the state matrix which are unaltered when several randomly generated decentralized feedback matrices are applied. The problem can be computationally very expensive for high order systems since it requires the solution of several eigenvalue problems. Further, it can be numerically unreliable as we shall see in Section 5.5. The approach presented here relates the concept of "blocking zeros" [6] of a linear multivariable system to fixed modes of decentralized systems. It is shown how such a characterization leads to a numerically stable and computationally efficient algorithm for computing d.f.m.'s.

The layout of this chapter is as follows: In Section 5.2 some preliminary results from linear algebra and control theory pertaining to decentralized multivariable systems, as well as some definitions that are specific to this chapter alone are presented; Section 5.3.1 uses several existing results to provide an algebraic characterization for d.f.m.'s. In Section 5.3.2, a computationally efficient and numerically reliable technique is proposed to compute the d.f.m.'s. Several examples are given in Section 5.4 and 5.5 to illustrate the use of the proposed algorithm. Finally, in Section 5.6, we present some results on the decentralized eigenvalues assignment problem.

5.2. PRELIMINARIES

We will require the following results for developing a computational method for determination of d.f.m.'s:

Definition 5.2.1 : A linear time-invariant multivariable system described by

$$\dot{x}(t) = Ax(t) + \sum_{i=1}^{N} B_i u_i(t) \tag{5.2.1a}$$

$$y_i(t) = C_i x(t), \quad i=1, \cdots, N \tag{5.2.1b}$$

where $x(t) \in \mathbb{R}^n$, $u_i(t) \in \mathbb{R}^{m_i}$, $y_i(t) \in \mathbb{R}^{p_i}$ and $i=1, \cdots, N$ is called an "N station decentralized system".

Definition 5.2.2 : Given the system (5.2.1), if we define a set of block-diagonal matrices $\mathbf{K}$ as

$$\mathbb{K} = \left\{ K \mid K = \text{block diag. } \left(K_1, \cdots, K_N \right),\ K_i \in \mathbb{R}^{m_i \times p_i} \right\} \tag{5.2.2}$$

then the set of d.f.m.'s of (5.2.1) with respect to $\mathbb{K}$ is defined as

$$\Lambda\left(A, B_i, C_i, \mathbb{K} \right) = \bigcap_{K \in \mathbb{K}} \sigma\left[A + \sum_{i=1}^{N} B_i K_i C_i \right] \tag{5.2.3}$$

where $\sigma(\bullet)$ denotes the set of eigenvalues of the matrix $(\bullet)$.

Remark : Note that every $\lambda \in \Lambda$ *is an eigenvalue of* A. As mentioned in the previous section, various characterizations of d.f.m.'s have been given. The one that is used to develop our algorithm is due to Anderson and Clements [10].

Theorem 5.2.3 : A scalar $\lambda \in \sigma(A)$ is a d.f.m. of the system described by (5.2.1) if and only if for *some partition* of the set $\Omega = 1, \cdots, N$ into disjoint subsets $\Omega_c = \left\{ i_1, \cdots, i_k \right\}$ and $\Omega_o = \left\{ i_{k+1}, \cdots, i_N \right\}$,

$$\text{rank} \begin{bmatrix} \lambda I_n - A & B_{i_1} & \cdots & B_{i_k} \\ C_{i_{k+1}} & 0 & \cdots & 0 \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ C_{i_N} & 0 & \cdots & 0 \end{bmatrix} < n \tag{5.2.4}$$

Proof : For a proof of the above result see [10].

Remark : The statement of the above theorem indicates that if a rank test is performed for all partitions of the set Ω into disjoint subsets Ω_c and Ω_o, it is possible to say conclusively whether or not a given scalar $\lambda \in \sigma(A)$ is a d.f.m. of (5.2.1). When n is large and/or N is large, such a rank test would require an undesirably high amount of computation. Alternatively, if it can be shown somehow that one such partition exists, then the scalar λ in (5.2.4) is a d.f.m. A systematic procedure for checking this is described in the next section. Here we make some further observations from inequality (5.2.4).

1. $\lambda \in \sigma(A)$. This implies that it can be assumed without loss of generality that all possible candidates for d.f.m.'s are known before Theorem 5.2.3 is applied.

2. The mode corresponding to a $\lambda \in \Lambda$ is uncontrollable from the stations $\left\{i_1, \cdots, i_k\right\}$ and unobservable from the stations $\left\{i_{k+1}, \cdots, i_N\right\}$. Therefore, any mode which is controllable *and* observable from one or more stations *cannot* be a d.f.m. Consequently, in order to compute d.f.m.'s, we can eliminate from consideration all those modes which are controllable *and* observable from at least one station. The remaining modes are uncontrollable and/or unobservable from each of the stations. Note that if a particular mode is uncontrollable from *all* the stations or unobservable from *all* the stations, then it is a d.f.m. - it is also a "centralized" fixed mode. In this case, the disjoint partition of Ω which satisfies (5.2.4) consists of Ω and the empty set Φ.

Definition 5.2.4 : A scalar $\lambda \in \mathbb{C}$ is a "blocking zero" [6] of the system (A, B, C) if

$$C \operatorname{adj}\left(\lambda I_n - A\right) B = 0 \tag{5.2.5}$$

where adj(•) denotes the adjoint of the matrix (•). In the above definition, if $\lambda \notin \sigma(A)$, then λ is a blocking zero if

$$C\left(\lambda I_n - A\right)^{-1} B = 0 \tag{5.2.6}$$

Remark : We shall use the definition of blocking zeros to obtain a characterization of d.f.m.'s in the next section.

In subsequent sections, we will assume for simplicity that the system is in USF. All the results with minor modifications can also be stated when the system is in RSF. The latter is, of course, more desirable from the computational point of view since it avoids the use of complex arithmetic.

5.3. CHARACTERIZATION AND COMPUTATION OF D.F.M.'S

In this section, we will investigate the conditions under which the inequality in (5.2.4) is satisfied. It will be shown later that these conditions can be easily used to devise an efficient and numerically stable algorithm to compute the d.f.m.'s of (5.2.1).

5.3.1. Characterization of d.f.m.'s

Consider a system (F, G, H) with $F \in \mathbb{R}^{n \times n}$, $G \in \mathbb{R}^{n \times m}$ and $H \in \mathbb{R}^{p \times n}$. Assume that F has distinct eigenvalues and let $\lambda \in \sigma(F)$ be an uncontrollable *and* unobservable mode of the system. Also, assume without loss of generality that the matrix F is in USF. Then, we can state the following:

Theorem 5.1 : Let $F = \begin{bmatrix} F_{11} & f_{12} \\ 0^T & \lambda \end{bmatrix}$, $G = \begin{bmatrix} G_1 \\ g_2^T \end{bmatrix}$ and $H = \begin{bmatrix} H_1 & h_2 \end{bmatrix}$, where F has distinct eigenvalues and λ is an uncontrollable *and* unobservable mode of (F, G, H). Then

$$rank \begin{bmatrix} \lambda I_n - F & G \\ H & 0 \end{bmatrix} < n$$

if and only if λ is a blocking zero of the system $\left(F_{11}, G_1, H_1\right)$.

Proof : Since λ is an uncontrollable mode of (F, G, H), it follows from the structure of F and G that $g_2 = 0$. By assumption, $\lambda \notin \sigma\left(F_{11}\right)$. Therefore, the matrix $\begin{bmatrix} \lambda I_n - F & G \\ H & 0 \end{bmatrix}$ can be factored as follows:

$$\begin{bmatrix} \lambda I_n - F & G \\ H & 0 \end{bmatrix} = \begin{bmatrix} \lambda I_{n-1} - F_{11} & 0 & 0 \\ 0^T & 1 & 0^T \\ 0 & 0 & I_p \end{bmatrix} \begin{bmatrix} I_{n-1} & 0 & 0 \\ 0^T & 1 & 0^T \\ H_1 & 0 & I_p \end{bmatrix} .$$

$$\begin{bmatrix} I_{n-1} & -\left(\lambda I_{n-1} - F_{11}\right)^{-1} f_{12} & \left(\lambda I_{n-1} - F_{11}\right)^{-1} G_1 \\ 0^T & 0 & 0^T \\ 0 & h_2 + H_1\left(\lambda I_{n-1} - F_{11}\right)^{-1} f_{12} & -H_1\left(\lambda I_{n-1} - F_{11}\right)^{-1} G_1 \end{bmatrix} \qquad (5.3.1)$$

Then,

$$\text{rank}\begin{bmatrix}\lambda I_n - F & G \\ H & 0\end{bmatrix} = n - 1 + \text{rank}\begin{bmatrix}0 & 0^T \\ h_2 + H_1\left(\lambda I_{n-1} - F_{11}\right)^{-1} f_{12} & -H_1\left(\lambda I_{n-1} - F_{11}\right)^{-1} G_1\end{bmatrix} \tag{5.3.2}$$

Next, consider the term $\text{adj}\left(\lambda I_n - F\right)$. Using elementary matrix algebra, we can write

$$\text{adj}\left(\lambda I_n - F\right) = \left[0 \;\; \alpha\right] \tag{5.3.3}$$

where, α is a complex vector of length n, given by

$$\alpha = \begin{bmatrix}\text{adj}\left(\lambda I_{n-1} - F_{11}\right) f_{12} \\ \det\left(\lambda I_{n-1} - F_{11}\right)\end{bmatrix}. \tag{5.3.4}$$

Note that since $\lambda \notin \sigma\left(F_{11}\right)$, $\det\left(\lambda I_{n-1} - F_{11}\right) \neq 0$ which implies that $\alpha \neq 0$. Also, from the identity

$$\left(\lambda I_n - F\right) \text{adj}\left(\lambda I_n - F\right) = \det\left(\lambda I_n - F\right) I_n \tag{5.3.5}$$

and the assumption that $\lambda \in \sigma(F)$, it follows that $\alpha \in \ker\left(\lambda I_n - F\right)$ where ker(•) denotes the kernal (null-space) of the matrix (•). Now, since λ is an unobservable mode of (F, G, H), it follows [19] that there exists a vector $v \neq 0$ such that

$$\left(\lambda I_n - F\right) v = 0 \tag{5.3.6}$$

and

$$Hv = 0. \tag{5.3.7}$$

But the assumptions that $\lambda \in \sigma(F)$ and F ~~has~~ distinct eigenvalues imply that $\text{rank}\left(\lambda I_n - F\right) = n - 1$. Therefore, the dimension of $\ker\left(\lambda I_n - F\right)$ is equal to 1, thereby implying that $v = \alpha\theta$ where $\theta \neq 0$ is some scalar in $\mathbb{C}$. Therefore from (5.3.7), we have $\alpha \in \ker H$ *i.e.*,

$$\left[H_1 \;\; h_2\right]\begin{bmatrix}\text{adj}\left(\lambda I_{n-1} - F_{11}\right) f_{12} \\ \det\left(\lambda I_{n-1} - F_{11}\right)\end{bmatrix} = 0. \tag{5.3.8}$$

Since $\lambda \notin \sigma\left(F_{11}\right)$, we may divide both sides of (5.3.8) by $\det\left(\lambda I_{n-1} - F_{11}\right)$ to get

$$\begin{bmatrix} H_1 & h_2 \end{bmatrix} \begin{bmatrix} (\lambda I_{n-1} - F_{11})^{-1} f_{12} \\ 1 \end{bmatrix} = 0$$

i.e.,

$$H_1(\lambda I_{n-1} - F_{11})^{-1} f_{12} + h_2 = 0. \qquad (5.3.9a)$$

Therefore, from (5.3.2), we get

$$\text{rank} \begin{bmatrix} \lambda I_n - F & G \\ H & 0 \end{bmatrix} = n - 1 + \text{rank} \begin{bmatrix} -H_1(\lambda I_{n-1} - F_{11})^{-1} G_1 \end{bmatrix} \qquad (5.3.9b)$$

which implies that the rank on the left hand side in (5.3.9b) is less than n if and only if $H_1(\lambda I_{n-1} - F_{11})^{-1} G_1 = 0$. The latter, by Definition 5.2.4 is equivalent to λ being a blocking zero of (F_{11}, G_1, H_1), completing the proof of the theorem.

The above result can be easily applied in characterizing d.f.m.'s of (5.2.1). Once the set Ω has been partitioned into disjoint subsets Ω_c and Ω_o, the matrix G will be the partitioned matrix $\begin{bmatrix} G_{i_1} & G_{i_2} & \cdots & G_{i_l} \end{bmatrix}$ and the matrix H will be the partitioned matrix $\begin{bmatrix} H_{i_{l+1}}^T & H_{i_{l+2}}^T & \cdots & H_{i_N}^T \end{bmatrix}^T$.

One way to obtain the partitions Ω_c and Ω_o is to find all the stations from which λ is uncontrollable and all the stations from which it is unobservable. This can be done merely by inspection once A has been reduced to its USM F with λ at position (n,n) ((1,1)) to check for uncontrollability (unobservability). However, the information obtained from the above inspection may not necessarily give a disjoint partition *i.e.*, $\Omega_c \cap \Omega_o \neq \emptyset$.

If there is a station γ such that $\gamma \in \Omega_c \cap \Omega_o$, then for (5.2.4) to hold, λ must necessarily be uncontrollable *and* unobservable from the γ-th station. Let $\Psi = \Omega_c \cap \Omega_o$ be non-empty. To verify the rank condition in (5.2.4), it is necessary to assign each element of Ψ to Ω_c or Ω_o such that $\Omega_c \cap \Omega_o = \emptyset$ and at the same time, the partition should be such that if λ is a d.f.m. of (5.2.1), then the rank condition in (5.2.4) is satisfied.

5.3.2. Computation of d.f.m.'s

The problem of computing the d.f.m.'s of the system described by (5.2.1) can be divided into two smaller problems:

1. Obtaining a set $\bar{\Lambda}$ such that the set of fixed modes $\Lambda \subseteq \bar{\Lambda} \subseteq \sigma(A)$. The set $\bar{\Lambda}$ consists of all possible candidates for d.f.m.'s.
2. Obtaining a disjoint partition of Ω (if it exists) such that the rank condition (5.2.4) is satisfied.

Using the reduction of centralized system to its BUHF and BLHF, we can find the set $\bar{\Lambda} \subseteq \sigma(A)$. The main steps in obtaining the set $\bar{\Lambda}$ consist of repeated application of the algorithm for reduction of a single-station system to its minimal order subsystem. This is done by first reducing the system (A, B, C) to a BUHF (to obtain its controllable subsystem). This would result in the structure: $\left(\begin{bmatrix} F_{11}^{(c)} & F_{12} \\ 0 & F_{22}^{(uc)} \end{bmatrix} \begin{bmatrix} G_1^{(c)} \\ 0 \end{bmatrix} \begin{bmatrix} H_1^{(c)} & H_2 \end{bmatrix} \right)$.

In the next step, when we find the controllable and observable subsystem of the given system, we consider the subsystem $\left(F_{11}^{(c)}, G_1^{(c)}, H_1^{(c)}\right)$. The observable subsystem of $\left(F_{11}^{(c)}, G_1^{(c)}, H_1^{(c)}\right)$ would be in BLHF. Assuming that the transformation for both the reductions are accumulated in an orthogonal matrix T, we have

$$\left(T^T A T, T^T B, C T\right) = \left(\begin{bmatrix} F_{11}^{(o)} & 0 & * \\ * & F_{22}^{(uo)} & * \\ 0 & 0 & F_{33}^{(uc)} \end{bmatrix}, \begin{bmatrix} G_{11} \\ * \\ 0 \end{bmatrix}, \begin{bmatrix} H_{11}^T \\ 0_{nn}^T \\ * \end{bmatrix} \right)$$

where, $\left(F_{11}^{(o)}, G_{11}, H_{11}\right)$ gives the controllable *and* observable subsystem and the remaining subsystem is uncontrollable, unobservable or both. In the case of a decentralized system, we proceed to check the controllability and observability properties from the next station. The state matrix at this stage would be $\begin{bmatrix} F_{22}^{(uo)} & * \\ 0 & F_{33}^{(uc)} \end{bmatrix}$, and the corresponding input and output matrices would be partitioned conformably. Next, we formally state an algorithm for obtaining the set $\bar{\Lambda}$.

Algorithm 5.1: (Determine the set $\bar{\Lambda}$)

Step I: (Initialization)

Set $i=1$, $n_o=0$ and $\bar{A}=A$

Step II: (Controllable subsystem from i-th station)

Reduce $(\bar{A}, B_i, C_i)$ to its BUHF $(F_i^{(c)}, G_i^{(c)}, H_i^{(c)})$ such that $(F_i^{(c)}, G_i^{(c)}, H_i^{(c)})$ contains the subsystem of $(\bar{A}, B_i, C_i)$ which is controllable from the i-th station.

Step III: (Observable subsystem from i-th station)

Reduce the system $(F_i^{(c)}, G_i^{(c)}, H_i^{(c)})$ to its BLHF $(F_i^{(o)}, G_i^{(o)}, H_i^{(o)})$ such that $(F_i^{(o)}, G_i^{(o)}, H_i^{(o)})$ contains the subsystem of $(F_i^{(c)}, G_i^{(c)}, H_i^{(c)})$ which is observable from the i-th station.

Set $n_o=\nu_i+n_o$ where ν_i is the dimension of the observable subsystem $(F_i^{(o)}, G_i^{(o)}, H_i^{(o)})$

Partition the system such that $(\bar{A}, B_j, C_j)$ $j=1, \cdots, N$ correspond to the subsystem which is uncontrollable and/or unobservable from the i-th station, *i.e.*,

$$F = \left[\begin{array}{c:c:c:c:c}
F_1^{(o)} & 0 & & * & \\ \hdashline
* & F_2^{(o)} & 0 & * & \\ \hdashline
 & * & \cdot \quad \cdot \quad \cdot & * & \\ \hdashline
0 & 0 & & F_i^{(o)} & * \\ \hdashline
 & & 0 & 0 & F_{i+1}^{(o)}
\end{array}\right] \tag{5.3.10}$$

where,

$$F_i^{(o)} = \begin{bmatrix} \tilde{F}_{11}^{(o)} & \tilde{F}_{12}^{(o)} & 0 & \cdots & 0 & 0 \\ \tilde{F}_{21}^{(o)} & \tilde{F}_{22}^{(o)} & \tilde{F}_{23}^{(o)} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ \tilde{F}_{r,1}^{(o)} & \tilde{F}_{r,2}^{(o)} & \tilde{F}_{r,3}^{(o)} & \cdots & \tilde{F}_{r,r}^{(o)} & \tilde{F}_{r,r+1}^{(o)} \\ \tilde{F}_{r+1,1}^{(o)} & \tilde{F}_{r+1,2}^{(o)} & \tilde{F}_{r+1,3}^{(o)} & \cdots & \tilde{F}_{r+1,r}^{(o)} & \tilde{F}_{r+1,r+1}^{(o)} \end{bmatrix}$$

The sub-matrix $\tilde{F}_{r,r+1}^{(o)}$ is a non-zero matrix if and only if $\left(F_i^{(c)}, H_i^{(c)}\right)$ is an observable pair [20-22]. If, however, $\tilde{F}_{r,r+1}^{(o)}$ is a zero matrix, then

$$F_i^{(o)} = \begin{bmatrix} \tilde{F}_{11}^{(o)} & \tilde{F}_{12}^{(o)} & 0 & \cdots & 0 & 0 \\ \tilde{F}_{21}^{(o)} & \tilde{F}_{22}^{(o)} & \tilde{F}_{23}^{(o)} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ \tilde{F}_{r-1,1}^{(o)} & \tilde{F}_{r-1,2}^{(o)} & \tilde{F}_{r-1,3}^{(o)} & \cdots & \tilde{F}_{r-2,r-1}^{(o)} & \tilde{F}_{r-1,r}^{(o)} \\ \tilde{F}_{r,1}^{(o)} & \tilde{F}_{r,2}^{(o)} & \tilde{F}_{r,3}^{(o)} & \cdots & \tilde{F}_{r-1,r-1}^{(o)} & \tilde{F}_{r,r}^{(o)} \end{bmatrix}$$

$$G = \begin{array}{c} \nu_1 \\ \\ \nu_2 \\ \vdots \\ \\ \nu_i \\ \\ \end{array} \left[\begin{array}{c|c|c|c|c|c} * & & & & & \\ \cline{1-1} & * & & & & \\ \cline{2-2} & & & * & & \\ \cline{4-4} * & & & & & * \\ & * & \cdots & * & \cdots & \\ \hline G_1^{(o)} & G_2^{(o)} & \cdots & G_i^{(o)} & \cdots & G_N^{(o)} \end{array}\right]$$

$$H = \begin{array}{c} \begin{array}{ccccc} \nu_1 & \nu_2 & \cdots & \nu_i & \end{array} \\ \left[\begin{array}{c|c|c|c|c} * \; 0 \cdots 0 & * & * & * & H_1^{(o)} \\ \hline * & * \; 0 \cdots 0 & * & * & H_2^{(o)} \\ \hline \vdots & & & & \\ \hline \multicolumn{3}{c|}{*} & * \; 0 \cdots 0 & H_i^{(o)} \\ \hline \vdots & & & \vdots & \\ \hline \multicolumn{4}{c|}{*} & H_N^{(o)} \end{array}\right] \end{array}$$

where *'s denote possible non-zero matrices.

The triple $\left(\tilde{A}, B_j, C_j\right)$ is given by

$$\tilde{A} = \begin{bmatrix} \tilde{F}_{r+1,r+1}^{(o)} & * \\ 0 & \tilde{F}_{i+1} \end{bmatrix} \text{ if } \tilde{F}_{r,r+1}^{(o)} \text{ is a zero matrix}$$

$$\tilde{A} = \hat{F}_{i+1} \quad \text{if } \tilde{F}^{(o)}_{r,r+1} \text{ has full row rank}$$
$$B_j = \left[G_1^{(o)} \cdots G_N^{(o)} \right]$$
$$C_j = \left[H_1^{(o)\,T} \cdots H_N^{(o)\,T} \right]^T$$

Step IV: (Termination)

If $n_o = n$ (the dimension of the given decentralized system), go to Step V (Exit 1)

If $i = N$ (the number of stations), go to Step VI (Exit 2)

Set $i = i+1$, go to Step II

Step V: (Exit 1)

$\tilde{\Lambda}$ is an empty set and since $\Lambda \subseteq \tilde{\Lambda}$, the given system has no d.f.m.'s; Stop.

Step VI; (Exit 2)

$\tilde{\Lambda} = \sigma(\tilde{A})$ where $\tilde{A}$, is the sub-matrix containing the modes that are uncontrollable and/or unobservable from all stations; Stop.

Remark : All $\lambda_i \in \tilde{\Lambda}$ need not be d.f.m.'s of the given system. The set $\tilde{\Lambda}$ contains those eigenvalues of the system which are possible candidates for being d.f.m 's. Usually the set $\tilde{\Lambda}$ is a very small subset of $\sigma(A)$.

Having obtained $\tilde{\Lambda}$, we now need to examine each element of $\tilde{\Lambda}$ to determine whether or not it is a d.f.m. of the system. We denote the subsystem obtained in Step III above, after the controllable and observable subsystems from all stations have been removed, by an $\tilde{n}$-th order system $\left(\tilde{A}, \tilde{B}_i, \tilde{C}_i\right)$, $i = 1, \cdots, N$.

To obtain the disjoint partitions Ω_c and Ω_o that also satisfy (2.4), it will be required to compute several *transfer relations* of the form

$$S_{ij} = C_{i1}\left(\lambda I_{n-1} - A'_{11}\right)^{-1} B_{1j} \qquad (5.3.11)$$

where $i, j \in \Omega$ and $i \neq j$. The matrices A_{11}, B_{1j} and C_{i1} above are obtained from the system $\left(A, B_j, C_i\right)$, $i, j = 1, \cdots, N$ as shown below:

$$A = \begin{bmatrix} A_{11} & a_{12} \\ 0^T & \lambda \end{bmatrix}, \; B_j = \begin{bmatrix} B_{1j} \\ b_{2j}^T \end{bmatrix} \text{ and } C_i = \begin{bmatrix} C_{i1} & C_{i2} \end{bmatrix}. \tag{5.3.12}$$

On noting that A is in its USF, we can solve (5.3.11) by first solving $(\lambda I_{n-1} - A_{11}) Z = B_{1j}$ for Z and then computing $S_{ij} = C_{i1} Z$. It is worth mentioning here that a similar method is used to compute frequency response matrices of a centralized system. The details of the method are given in the next chapter. Next, we describe a systematic procedure which will enable us to find the required disjoint partitions of Ω into $\hat{\Omega}_c$ and $\hat{\Omega}_o$. Note that in the algorithm, the tableau need not be constructed, since all the information can be stored in the two sets. However, it makes it easier to understand the algorithm.

Algorithm 5.2: (Disjoint partitions Ω_c and Ω_o)

Step I: (a) Set $q = 1$ and $\Psi = \Phi$

(b) Transform $(\tilde{A}, \tilde{B}_l, \tilde{C}_l)$, $l = 1, \cdots, N$, to a USF such that $\lambda_q \in \tilde{\Lambda}$ in its (1,1) position.

(c) Form a set $\hat{\Omega}_o$ where $\hat{\Omega}_o$ contains l if the first column of the output matrix $\tilde{C}_l$ is a *zero* vector.

Step II: (a) Transform $(\tilde{A}, \tilde{B}_l, \tilde{C}_l)$, $l = 1, \cdots N$, such that $\tilde{A}$ is a USM and λ_q is in its $(\tilde{n}, \tilde{n})$ position.

(b) Form a set $\hat{\Omega}_c$ where $\hat{\Omega}_c$ contains l if the last row of the input matrix $\tilde{B}_l$ is a *zero* vector.

Step III: Set up a tableau such that its columns correspond to the elements of $\hat{\Omega}_o = \left\{ i_1^*, \cdots, i_s^*, i_{s+1}, \cdots, i_\nu \right\}$ and its rows correspond to the elements of $\hat{\Omega}_c = \left\{ i_{s+1}, \cdots, i_\nu, i_{\nu+1}^*, \cdots, i_N^* \right\}$. Let $\Psi = \hat{\Omega}_c \cap \hat{\Omega}_o$ $\left(= \left\{ i_{s+1}, \cdots, i_\nu \right\} \right)$. The tableau will have the following layout:

	i_1^*	i_2^*	● ● ●	i_s^*	i_{s+1}	● ● ●	i_ν
i_{s+1}			● ● ●			● ● ●	
i_{s+2}			● ● ●			● ● ●	
●			●			●	
●			●			●	
●			●			●	
i_ν			● ● ●			● ● ●	
$i_{\nu+1}^*$			● ● ●			● ● ●	
●			●			●	
●			●			●	
●			●			●	
i_N^*			● ● ●			● ● ●	

where the "starred" elements are the ones which correspond to the stations from where λ_q is either controllable but unobservable or uncontrollable but observable.

$$\text{Set } \Omega_c = \left\{ i_1^*, \cdots, i_s^* \right\},\ \Omega_o = \left\{ i_{\nu+1}^*, \cdots, i_N^* \right\} \text{ and } k = 1.$$

Step IV: (a) Corresponding to the k-th (starred) station of Ω_o, form

$$S_{kl} = C_{k1} \left(\lambda_q I_{n-1} - A_{11} \right)^{-1} B_{1l},\ l \in \hat{\Omega}_c$$

(b) For each value of $l \in \hat{\Omega}_c$,

(i) If $S_{kl} \neq 0$ and $l \in \Omega_c$, go to Step V

(ii) If $S_{kl} \neq 0$ and $l \in \Psi$,

Set $\hat{\Omega}_c = \hat{\Omega}_c \ominus \left\{ l \right\}$ (remove l from $\hat{\Omega}_c$)

Set $\Omega_o = \Omega_o \oplus \left\{ l \right\}$ (add l to Ω_o)

Set $\Psi = \Psi \ominus \left\{ l \right\}$, go to Step IV(c).

(iii) If $S_{kl} = 0 \ \forall \ l \in \hat{\Omega}_c$, go to Step IV(c).

(iv). If $\Psi = \Phi$ (empty set), go to Step VI(b)

(c) If all (starred) stations of $\hat{\Omega}_c$ are exhausted, go to Step VI(a). Else, set $k = k + 1$,

go to Step IV(a).

Step V: There is no disjoint partition for which (2.4) is satisfied. Therefore λ_q is not a d.f.m.

Go to Step VII.

Step VI: (a) The disjoint partition satisfying (2.4) is given by Ω_o and $\hat{\Omega}_c$. Therefore λ_q is a d.f.m. Set $\Lambda = \Lambda \oplus \lambda_q$

Go to Step VII.

(b) The disjoint partition satisfying (2.4) is given by Ω_o and Ω_c. Therefore λ_q is a d.f.m. Set $\Lambda = \Lambda \oplus \lambda_q$

Go to Step VII.

Step VII: If $q = \hat{n}$, Stop; else,

Set $q = q + 1$, go to Step I(b).

At the end of the Algorithm 5.2, the set Λ ($\subseteq \bar{\Lambda}$) contains all the eigenvalues of A which are d.f.m.'s of (5.2.1). The steps of Algorithm 5.2 are illustrated by means of an example in next section.

5.4. DISCUSSION OF THE RESULTS

In this section, we will discuss various computational and numerical properties of the proposed algorithms.

5.4.1. General Remarks

1. The characterization provided in the previous section gives us "structural" information about decentralized systems. This is obtained by relating the eigenvalues of the system that are also d.f.m.'s to blocking zeros of certain subsystems derived from the given decentralized system. It should be mentioned that the characterization given above is fairly general and does not impose any restriction on the structure of the decen-

tralized system. Simpler results can, of course, be obtained when the system has certain additional properties, *e.g.*, block diagonal structure for A, or interconnected systems with only input and output matrices in block diagonal form, *etc*.

2. Algorithm 5.1, used to obtain $\bar{A}$, uses only orthogonal (numerically stable) state coordinate transformations and is numerically "backward stable" [22-24]. This is a desirable property from the point of view of application to very high order systems. Note that Algorithm 5.1 and most of the existing algorithms for determination of d.f.m.'s require the knowledge of the eigenvalues of open-loop system. Numerically, this is not desirable because the performance of the algorithm would be affected by the conditioning of the open-loop eigenvalue determination problem.

3. In obtaining the sub-matrix $\bar{A}$, whose eigenvalues are the possible candidates for d.f.m.'s, only numerically stable algorithms are used. Moreover, since the order of $\bar{A}$ is in general very small compared to the order of the original state matrix A, the conditioning of the eigenvalue problem of $\bar{A}$ would, in the worst case, be the same as that of A. Hence, from the point of view of the accuracy of determining the set of d.f.m.'s, the use of Algorithm 5.1 is very attractive.

4. It should be pointed out that since the matrix A is real, all its complex eigenvalues occur in conjugate pairs. The same is true for the blocking zeros of (5.2.1). In Algorithm 5.2, the use of complex arithmetic would be necessary to examine the controllability and observability of a particular mode and also to compute the equality in (5.3.11). However, the use of complex arithmetic can be easily avoided by using the RSF instead of the USF. It can be easily shown *e.g.*, see [15], that if in the triple (F,G,H), F is a RSM, with a $c-c$ pair of eigenvalues (λ,λ^*) in the first 2×2 block of F, then (λ,λ^*) is unobservable if and only if the first two columns of the corresponding output matrix H are *zero vectors.* A similar result can be stated for controllability of (λ,λ^*).

To avoid the use of complex arithmetic in (5.3.11), the real and imaginary parts of the expression in (5.3.11) can be obtained separately as described below. The approach used is similar to that used for computing frequency response from state-space models as discussed in Chapter VI.

Consider the triple (F_{11}, G_{1j}, H_{i1}) and let $\lambda = \alpha + j\beta$. The transfer relation in (5.3.11) can be evaluated by solving

$$[(\alpha + j\beta)I_{n-1} - F_{11}]Z = G_{1j} \tag{5.4.1}$$

for Z and then computing

$$S_{ij} = H_{i1}Z \tag{5.4.2}$$

Let $Z = Z_1 + jZ_2$, then, equating the real and imaginary parts on both sides of (5.4.1), we get

$$\begin{bmatrix} \alpha I_{n-1} - F_{11} & -\beta I_{n-1} \\ \beta I_{n-1} & \alpha I_{n-1} - F_{11} \end{bmatrix} \begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} = \begin{bmatrix} G_{1j} \\ 0 \end{bmatrix} \tag{5.4.3}$$

From (5.4.3), it can be easily shown that

$$Z_1 = \frac{1}{\beta}[\alpha I_{n-1} - F_{11}]Z_2 \tag{5.4.4}$$

$$\frac{1}{\beta}\left[(\alpha I_{n-1} - F_{11})^2 - \beta^2 I_{n-1}\right] Z_2 = G_{1j}\,. \tag{5.4.5}$$

The system of linear equations (5.4.5) can be solved by carrying out an LU decomposition of $\left[(\alpha I_{n-1} - F_{11})^2 - \beta^2 I_{n-1}\right]$ and subsequently solving the triangular systems of linear equations. Z_1 is then easily evaluated from (5.4.4). Knowing Z_1 and Z_2, the real and imaginary parts of S_{ij} in (5.3.11) may be evaluated as

$$\begin{aligned} \text{Re } S_{ij} &= H_{i1}Z_1 \\ \text{Im } S_{ij} &= H_{i1}Z_2 \end{aligned}$$

5. From (5.3.9), it is clear that if the given system has distinct eigenvalues and λ satisfies (5.3.9a), then the rank in (5.3.9b) is less than n if and only if $\left(\lambda_q I_{n-1} - F_{11}\right)^{-1} G_1$ lies in the null space of H_1. Therefore, if the dimension of the null space of H_1 is zero, then λ_q cannot be a d.f.m. A preliminary rank test by Householder triangularization or using the singular value decomposition of H_1 may be used to determine the rank of H_1. The transfer relations in (5.3.11) may then be evaluated if and

only if the null space of H_1 has non-zero dimension.

6. The breakdown of the operations count required for the proposed algorithms is as follows:

(i) Obtaining the matrix $\tilde{A}$ *i.e.*, eliminating all controllable and observable modes from various stations: This requires $O\left(n^3\right)$ operations.

(ii) Finding the eigenvalues of $\tilde{A} \in \mathbb{R}^{\tilde{n} \times \tilde{n}}$: The total number of operations required here will usually be insignificant because in general $\tilde{n} << \eta$. But for the case when $\tilde{n}$ is large, $O\left(\tilde{n}^3\right)$ operations will enable us to find the set of eigenvalues which are also the possible candidates for d.f.m 's.

(iii) Obtaining the partition $\hat{\Omega}_c$ and $\hat{\Omega}_o$: Since we need to rearrange the eigenvalues of $\tilde{A}$ to check for uncontrollability and/or unobservability, obtaining the partition would require $O\left(\tilde{n}^3\right)$ operations.

(iv) Finally, evaluating the transfer relations S_{ij} will require $O\left(n^3\right)$ operations.

The total number of operations required to compute d.f.m.'s will be approximately $O\left(n^3\right)$ with a constant multiplier ~5. However, if the approach in [3] or [10] is used, the count will be considerably higher because the former approach requires solutions of several algebraic eigenvalue problems for matrices of the same order as that of the decentralized system, and the latter requires several rank tests on systems of order greater than n.

5.4.2. Examples

We will now illustrate Algorithm 5.2 by an example of a decentralized system with $\hat{\Omega}_c = \left\{1,2,4,5,6,7\right\}$, $\hat{\Omega}_o = \left\{1,2,5,6,7\right\}$, $\Omega_c = \left\{4\right\}$ and $\Omega_o = \left\{3\right\}$ Assume that $C_{31}\left(\lambda_q I_{n-1} - A_{11}\right)^{-1} B_{14} = 0$ *i.e.*, the condition in Step IV-b(i) is not met. Given below are the tableau at various stages:

Step IV-a Evaluate $\left[S_{31}\ S_{32}\ S_{34}\ S_{35}\ S_{36}\ S_{37}\right]$

Let $S_{31}, S_{36} \neq 0$.

IV-b(ii) $\hat{\Omega}_c = \left\{2,4,5,7\right\}$, $\Psi = \left\{2,5,7\right\}$, $\Omega_c = \left\{4\right\}$ and $\Omega_o = \left\{3,1,6\right\}$

	4*	2	5	7
3*				
1*		S_{12}		
6*				
2				
5				
7				

IV-a Corresponding to station 1, let $S_{12} \neq 0$ as indicated above.

IV-b(ii) $\hat{\Omega}_c = \left\{4,5,7\right\}$, $\Psi = \left\{5,7\right\}$, $\Omega_c = \left\{4\right\}$ and $\Omega_o = \left\{3,1,2,6\right\}$. The tableau now has the following layout:

	4*	5	7
3*			
1*			
6*			
2*			
5			
7			

IV-a Assume that corresponding to station 2, all S_{kl}'s are zero

IV-c Set $k = k+1$, go to IV(a)

IV-a Assume that all S_{kl}'s corresponding to station 6 are zero

VI-a λ_q is a d.f.m. and the disjoint partition $\Omega_o = \left\{1,2,3,6\right\}$ and $\Omega_c = \left\{4,5,7\right\}$ satisfies the rank condition (5.2.4).

As can be seen from the above considerations, if either S_{24} or S_{64} were non-zero, then λ_q would not be a d.f.m. If S_{65} and S_{67} were non-zero, however, S_{54} and S_{74} were zero,

λ_q would be a d.f.m. with $\Omega_o = \left\{1,2,3,5,6,7\right\}$ and $\Omega_c = \left\{4\right\}$. The algorithm ends when a disjoint partition is found or else when a conclusion is reached that no disjoint partition (for the value of λ_q under consideration) satisfying (5.2.4) exists.

5.5. NUMERICAL EXAMPLES

In this section, we consider two numerical examples to illustrate the proposed algorithms.

Example 1 : Consider a 3 station decentralized system. The matrices describing the system are given in Table 5.1. The state matrix has eigenvalues at $\left\{-2,-1.5,-1.0,3.0,2.5,2.0,1.5,1.0\right\}$. At the conclusion of Algorithm 5.1, it is found that $\tilde{\Lambda} = \left\{2.0\right\}$ *i.e.*, only λ=2.0 is a possible d.f.m. Next, on applying Algorithm 5.3.2, it is found that λ=2.0 is unobservable from stations 1 and 2 and uncontrollable from stations 1 and 3. Therefore $\hat{\Omega}_o = \left\{i_1,i_2\right\}$ and $\hat{\Omega}_c = \left\{i_1,i_3\right\}$. Following the steps of Algorithm 5.2, it is found that $S_{21} = C_{21}\left(\lambda_q I_{n-1} - A_{11}\right)^{-1} B_{11}$ and $S_{23} = C_{21}\left(\lambda_q I_{n-1} - A_{11}\right)^{-1} B_{13}$ are both zero matrices for λ_q=2.0. The elements of the matrices S_{12} and S_{23} are given in Table 5.2. They are of the order of 10^{-16} and can be safely assumed to be zero. Therefore, we have the partition $\Omega_c = \left\{i_1,i_3\right\}$ and $\Omega_c = \left\{i_2\right\}$ which are disjoint and satisfy the condition in (5.2.4) implying that λ = 2.0 is a d.f.m. of the given system.

$$\left[\begin{matrix}
-2.9023E+00 & 4.9023E+00 & 3.6981E+00 & -1.0917E+00 & -1.3396E+00 & -3.8336E+00 \\
1.4576E+01 & 8.6636E+00 & 4.4715E+00 & -6.1636E+00 & -1.0319E+00 & 1.6921E+00 \\
-1.3017E+01 & 5.2428E+00 & 6.8087E+00 & 3.6585E-01 & -1.2503E+00 & -9.4017E+00 \\
-9.3585E+00 & 9.3585E+00 & 9.2334E+00 & -1.2498E+00 & -1.1024E+00 & -1.0211E+01 \\
1.0902E+01 & 1.0338E+01 & 6.5778E+00 & -6.5275E+00 & -1.4633E+00 & -1.2775E+00 \\
7.4390E+00 & 6.0263E+00 & 4.3652E+00 & -4.2282E+00 & -1.6595E+00 & -1.1370E+00 \\
4.5910E+00 & 7.4755E+00 & 4,9822E+00 & -3.6650E+00 & -2.2821E+00 & -2.5444E+00 \\
1.6853E+01 & 5.7854E+00 & 2.2048E+00 & -5.7854E+00 & -1.7986E+00 & 3.5806E+00
\end{matrix}\right.$$

$$\left.\begin{matrix}
1.3545E-01 & 1.3545E-01 \\
-1.1852E+01 & 5.2806E-01 \\
5.5930E+00 & -1.8385E-01 \\
9.7734E-01 & 9.7734E-01 \\
-1,0989E+01 & 1.3914E+00 \\
-5.9165E+00 & 6.8663E-01 \\
-5.1261E+00 & 4.7701E-01 \\
-1.1474E+01 & 1.9062E+00
\end{matrix}\right]$$

Table 5.1a State Matrix A

$$\left[\begin{array}{ccc|ccc|}
-6.2881E+00 & -8.3841E+00 & -1.0480E+01 & 2.4061E+01 & 8.8694E+00 & 7.8089E+00 \\
6.9893E+00 & 9.3191E+00 & 1.1649E+01 & 1.6406E+01 & 1.9285E+01 & 8.9405E+00 \\
-1.3548E+01 & -1.8064E+01 & -2.2580E+01 & 2.5208E+01 & 1.8083E+01 & 3.1479E+01 \\
-5.1983E+00 & -6.9311E+00 & -8.6639E+00 & 2.3850E+01 & 1.9948E+01 & 1.7032E+01 \\
-7.6482E+00 & -1.0198E+01 & -1.2747E+01 & 1.0157E+01 & 9.9392E+00 & 4.7007E+00 \\
-5.8995E+00 & -7.8660E+00 & -9.8326E+00 & 1.9488E+01 & 2.6586E+01 & 3.1634E+01 \\
-6.2881E+00 & -8.3841E+00 & -1.0480E+01 & 2.9748E+01 & 2.5045E+01 & 1.7088E+01 \\
1.3601E+00 & 1.8135E+00 & 2.2669E+00 & 1.7739E+01 & 1.7020E+01 & 1.4477E+01
\end{array}\right.$$

$$\left.\begin{matrix}
-2.6200E+00 & -3.1440E+00 & -1.3100E+01 \\
2.9122E+00 & 3.4946E+00 & 1.4561E+01 \\
-5.6449E+00 & -6.7739E+00 & -2.8224E+01 \\
-2.1660E+00 & -2.5992E+00 & -1.0830E+01 \\
-3.1868E+00 & -3.8241E+00 & -1.5934E+01 \\
-2.4581E+00 & -2.9498E+00 & -1.2291E+01 \\
-2.6200E+00 & -3.1440E+00 & -1.3100E+01 \\
5.6672E-01 & 6.8007E-01 & 2.8336E+00
\end{matrix}\right]$$

Table 5.1b Input Matrices $\left[\hat{B}_1 \mid B_2 \mid \hat{B}_3\right]$

$$\left|\begin{matrix}
-2.6324E+00 & -8.8165E-01 & 2.2165E-01 & 8.8165E-01 & 3.0600E-01 & -1.1033E+00 \\
6.9654E-01 & -6.9654E-01 & -6.3869E-01 & 8.7892E-02 & 6.9897E-02 & 6.5073E-01 \\
1.2513E+00 & 7.4134E-01 & 3.5784E+00 & -2.6605E-01 & -9.9435E-02 & -3.6871E+00
\end{matrix}\right.$$

$$\left.\begin{array}{rr} 1.2231E+00 & 5.2765E-01 \\ -1.2046E-02 & -1.2046E-02 \\ -3.7007E+00 & 2.9408E+00 \end{array}\right]$$

$$\left[\begin{array}{rrrrrr} 9.7288E-01 & -6.8240E-01 & 8.3212E-01 & 7.3755E-02 & 5.1544E-02 & -8.0594E-01 \\ 6.3385E-01 & 1.2532E+00 & 1.8749E+00 & -7.7794E-01 & -2.2269E-01 & -1.4717E+00 \\ 7.2518E-02 & -7.2518E-02 & -4.0037E-01 & 6.0481E-01 & -1.5815E-01 & 5.7008E-01 \end{array}\right.$$

$$\left.\begin{array}{rr} -1.7267E+00 & 1.4404E+00 \\ -7.4462E-01 & 1.1140E+00 \\ -1.6970E-01 & -1.6970E-01 \end{array}\right];$$

$$\left[\begin{array}{rrrrrr} 9.5268E-02 & -6.8710E-02 & 6.6781E-03 & 9.0577E-02 & -1.0730E-01 & 1.5351E-01 \\ -2.2525E-01 & 9.9213E-02 & 1.7152E-01 & 1.4171E-01 & -4.7781E-02 & -9.4807E-02 \\ -2.3147E-01 & 1.2077E-01 & 2.0929E-01 & 1.1203E-02 & -2.8055E-02 & -8.5839E-02 \end{array}\right.$$

$$\left.\begin{array}{rr} 1.5610E-01 & 4.7215E-02 \\ 1.2588E-01 & 6.3608E-02 \\ 3.3578E-01 & -5.9450E-02 \end{array}\right]$$

Table 5.1c Output Matrices C_1, C_2 and C_3.

$$\left[\begin{array}{rrr|rrr} -1.1102d-16 & -1.1102d-16 & -3.3307d-16 & -8.3267d-17 & -5.5511d-17 & -2.2204d-16 \\ -2.2204d-16 & -4.4409d-16 & -4.4409d-16 & -1.1102d-16 & -1.1102d-16 & 0.0000d+00 \\ 2.2204d-16 & 0.0000d+00 & 4.4409d-16 & 1.1102d-16 & 1.1102d-16 & 4.4409d-16 \end{array}\right]$$

Table 5.2 Transfer relations $\left[S_{21} \middle| S_{23}\right]$ in Example 1.

Example 2 : In this example, we will illustrate the difficulty that may be encountered in deciding whether or not a particular eigenvalue is a d.f.m. using the test proposed in [3]. For instance, if the open-loop eigenvalue problem is ill-conditioned then, depending on how a "random" feedback matrix affects the closed-loop eigenvalues, some of the eigenvalues may change. By inspection of the eigenvalues of the open-loop and closed-loop system, it may not always be possible to say conclusively whether or not there are any d.f.m.'s. The characterization and computational algorithms presented here enable us to conclude with greater certainty if a certain mode is fixed or not. The data for this example is given in Table 5.3. For several randomly generated values of the feedback matrices with their elements multiplied by factors ranging from 10^{-2} to 10^{2}, it was found that certain eigenvalues of the system do not "move" appreciably. However, as the magnitude of the elements of the feedback matrices were gradually increased to 10^{10}, the eigenvalues changed completely. Table 5.4 compares typical values of some of the open-loop eigenvalues with those of the closed-loop eigenvalues for 3 representative sets of values of feedback gains: $k_{ij} \sim 10^{-2}$, $k_{ij} \sim 10^{2}$ and $k_{ij} \sim 10^{10}$ where the k_{ij} are the elements of the randomly generated feedback matrices.

It should be noted that the eigenvalues at -3.0 and 2.5 do not change appreciably for the $k_{ij} \sim 10^{-2}$ and 10^{2} but are altered completely for $k_{ij} \sim 10^{10}$. In contrast, corresponding to the eigenvalue at -4.6, we have eigenvalues at -4.59579, -4.59402 and -4.46691 for $k_{ij} \sim 10^{-2}, 10^{2}$ and 10^{10} respectively. With reference to the eigenvalue at 8.0, we see that for $k_{ij} \sim 10^{-2}$, there is an eigenvalue at 8.0120 while for $k_{ij} \sim 10^{2}$, we have an eigenvalue at 7.99998. Such observations with this and several other examples suggest that some difficulties could arise in computing the set of d.f.m.'s using the characterization given in (5.2.3). Applying the algorithms proposed above, it was found conclusively that the system does not have any d.f.m.'s. This was further confirmed by performing the rank test (using the singular value decomposition) on the system matrix in (5.2.4).

6.0000E+00	-2.1500E+00	8.4500E+00	-2.5500E+00	4.0000E-01	1.7500E+00	1.0200E+01	0.0000E+00	-1.0200E+01	-8.0500E+00
3.0500E+01	-3.4050E+01	1.2850E+01	-4.4050E+01	4.1000E+00	1.3550E+01	3.5300E+01	-1.5500E+01	-1.0900E+01	4.6500E+00
-7.5000E+00	6.5000E+00	1.1000E+01	1.4000E+01	-7.5000E+00	-6.5000E+00	7.5000E+00	0.0000E+00	0.0000E+00	-1.4000E+01
1.2000E+01	-6.4000E+00	4.2000E+00	-1.2800E+01	4.0000E-01	6.0000E+00	1.0200E+01	0.0000E+00	-1.0200E+01	-3.8000E+00
2.9500E+01	-3.3000E+01	-2.0000E+00	-4.7000E+01	8.0000E+00	1.7500E+01	1.5500E+01	-1.5500E+01	0.0000E+00	1.7500E+01
3.1200E+01	-3.8600E+01	1.2100E+01	-4.4100E+01	5.5000E+00	1.8100E+01	3.3200E+01	-1.5500E+01	-1.0200E+01	5.4000E+00
3.0300E+01	-3.2100E+01	1.0200E+01	-4.6500E+01	2.5000E+00	1.6200E+01	3.2300E+01	-1.5500E+01	-1.0900E+01	5.7000E+00
-6.2000E+00	2.0000E-01	1.5100E+01	9.3000E+00	-9.1000E+00	-3.1000E+00	1.2000E+01	-7.5000E+00	0.0000E+00	-1.2200E+01
1.9800E+01	-2.8550E+01	1.2350E+01	-3.2450E+01	3.9000E+00	1.2650E+01	2.5000E+01	-1.5500E+01	-5.0000E+00	3.5500E+00
3.8000E+00	2.1500E+00	1.3450E+01	-1.2500E+00	-8.5000E+00	-2.5500E+00	1.9800E+01	0.0000E+00	-1.0900E+01	-1.6050E+01

Table 5.3a State Matrix A in Example 2.

1.4100E+00	-7.0000E-01	-4.8920E+01	-4.0000E-02
1.2401E+02	1.2639E+02	3.5000E-01	-4.0900E+01
2.0800E+00	2.1700E+00	-1.0300E+00	0.0000E+00
1.4100E+00	-7.0000E-01	-4.9010E+01	-1.8300E+00
1.2593E+02	1.2993E+02	9.0000E-02	-4.2770E+01
1.2942E+02	1.3202E+02	-5.0180E+01	-4.1020E+01
1.2401E+02	1.2577E+02	5.0000E-02	-4.2690E+01
2.0700E+00	1.5400E+00	-1.3300E+00	-4.0990E+01
1.2950E+02	1.3148E+02	-1.2400E+00	-4.1020E+01
-1.9200E+00	-4.1600E+00	3.1000E-01	-1.7100E+00

Table 5.3b Input Matrix $B = \left[B_1 | B_2 \right]$ in Example 2.

$$\left[\begin{array}{cccccc} -4.7323E+01 & 2.3526E+01 & -2.6854E+01 & 2.1844E+01 & 1.6820E+00 & -2.3577E+01 \\ 4.2880E+01 & -6.2423E+01 & 1.9768E+01 & -6.4265E+01 & -6.7000E-02 & 2.1385E+01 \\ \\ 1.7593E+01 & -1.8059E+01 & 1.2830E+00 & -2.6374E+01 & 8.9410E+00 & 8.7710E+00 \\ 7.3712E+01 & -7.4850E+01 & 2.1835E+00 & -1.1067E+02 & 3.7831E+01 & 3.6946E+01 \end{array}\right.$$

$$\left.\begin{array}{cccc} -5.0687E+01 & 0.0000E+00 & 4.9056E+01 & 2.7161E+01 \\ 4.1270E+01 & -4.0996E+01 & -1.5800E-01 & 2.1269E+01 \\ \\ 1.0105E+01 & -2.2000E-02 & -9.7580E+00 & 8.0050E+00 \\ 3.8950E+01 & -5.1000E-02 & -3.8877E+01 & 3.5917E+01 \end{array}\right].$$

Table 5.3c Ouput Matrix $C = \left[C_1^T \mid C_2^T \right]^T$ in Example 2.

some o/l e.v.'s	c/l e.v.'s for $k_{ij} \sim 10^{-2}$	c/l e.v.'s for $k_{ij} \sim 10^{2}$	c/l e.v.'s for $k_{ij} \sim 10^{10}$
	-5.359871211893602d+00	1.981705713622234d+02	2.007855000236994d+14
	±3.066671782119345d+00i	±5.349405101541089d+04i	±5.349407732726706d+16i
	4.909457853960178d+00	4.794988863438237d+00	8.994344754265372d+00
	3.095864118165678d+00	- 2.272479821115828d+00	±2.495595121512005d+01i
5.9	5.833571910641285d+00	5.806693718096567d+00	- 6.761211902258197d-01
-4.6	- 4.595795866440305d+00	- 4.594925581006383d+00	- 4.457406647868712d+01
-6.0	- 5.995408391604500d+00	- 6.003566998696624d+00	- 4.752607649352673d+04
-3.0	- 2.999789534354356d+00	- 3.000821692878285d+00	4.101839349739446d+01
8.0	8.012032299929436d+00	7.999957971270422d+00	- 4.466914156228984d+00
2.5	2.499967133489732d+00	2.500010816079302d+00	1.240053279603918d+01

Table 5.4 Closed-loop eigenvalues for $K_{ij} \sim 10^{-2}$, 10^2 10^{10}

A natural question that arises from Example 2 is what zero threshold value (*i.e.*, numerical value for "zero") should be used in the algorithms proposed in the previous section. Among the numerical techniques used in the above algorithms, the maximum error is accumulated in the reduction of the state matrix to its RSF and hence an error bound on this reduction [23,24] may be used to define the zero threshold value.

5.6. DECENTRALIZED EVA PROBLEM

In this section, we consider the problem of control of decentralized linear multivariable systems using dynamic output feedback between the sensors and actuators of each station. Under the assumption of certain controllability and observability properties of the system, it is shown that a feedback control law that stabilizes the given system can always be found. These results are presented as extensions of the problem of eigenvalue assignment by dynamic output feedback in single-station systems. It should be pointed out that there are still some theoretical issues which remain to be solved in the decentralized EVA problem. However, the issues discussed in this section can lead to an efficient computational procedure for assigning the eigenvalues of the given decentralized system.

5.6.1. Problem Statement

For the systems described by (5.2.1), the problem of EVA that we investigate is to determine a decentralized control defined by local dynamic output feedback controllers

$$u_i(t) = v_i(t) - \phi_i(t) \tag{5.6.1a}$$
$$\phi_i(t) = H_i z_i(t) + J_i y_i(t) \tag{5.6.1b}$$
$$\dot{z}_i(t) = F_i z_i(t) + G_i y_i(t) \tag{5.6.1c}$$

where $\phi_i(t) \in \mathbb{R}^{m_i}$, $z_i(t) \in \mathbb{R}^{q_i}$, such that the state matrix of the resulting closed-loop system has eigenvalues at any desired locations in the complex plane (subject to $c-c$ pairing).

The only methods available currently for solving the problem is that proposed by DeCarlo *et al.* [25] and Corfmat and Morse [11]. The continuation method is mathematically sound, however, it requires non-linear programming approach for solving the problem. Computationally it could be extremely demanding. The approach used in [11] is to convert the decentralized EVA problem to a "centralized" EVA problem by applying *constant decentralized feedback* to make the system controllable from a single station. Dynamic output feedback is then applied to this station to assign all the closed-loop eigenvalues at the desired locations. Computationally, the approach is not very attractive, especially for high order systems, because the mechanism of EVA is concentrated at one station. The proposed method shares the burden of EVA between various stations. The procedure is sequential in nature in that each station assigns a number of additional eigenvalues without altering the ones that have been assigned by the preceding stations. For each station, the algorithm for EVA by output feedback, proposed in the previous chapter is used to assign the additional eigenvalues.

5.6.2. Preliminary Considerations

In general, for a controllable and observable single-station, multi-input, multi-output system, output feedback between any input-output pair affects all the eigenvalues of the system. For a multi-station system, unless the system is "diagonally decoupled" or can be diagonally decoupled, an output feedback from any given station will affect all those modes that are controllable and observable from that station. This could in turn affect the controllability and observability properties of the system from the other stations. The problem of decentralized control, therefore, is a two part problem. First, from a given set of stations, we assign a desired set of closed-loop eigenvalues and second, we must ensure that the feedback at any other station preserves these eigenvalues and possibly assigns some additional eigenvalues.

For simplicity of presentation, assume that we have a two station system described by

$$\dot{x}(t) = Ax(t) + B_1u_1(t) + B_2u_2(t) \tag{5.6.2a}$$

$$y_1(t) = C_1x(t) \tag{5.6.2b}$$

$$y_2(t) = C_2x(t) \tag{5.6.2c}$$

where $u_i(t) \in \mathbb{R}^{m_i}$, $y_i(t) \in \mathbb{R}^{p_i}$. Further, assume that the system has no centralized or decentralized fixed modes. This assumption is by no means a restriction on the strategy to be proposed, since under the given decentralized structure, these modes are unaffected by decentralized output feedback and hence cannot be relocated. Let the dimensions of the controllable and observable subsystems from the first and second stations be n_1 and n_2 respectively. Note that under the assumptions above, if we denote the set of all modes that are controllable and observable from the i-th station as $\left\{ n_i \right\}$, then $\left\{n_1\right\} \cup \left\{n_2\right\} \sim \left\{n_1\right\} \cap \left\{n_2\right\} = \left\{n\right\}$. Therefore, all those modes that are uncontrollable and/or unobservable from the first station would be controllable and observable from the second station and *vice versa.*

Under the above assumptions, the following two situations can arise:

Case 1 - $\left\{n_1\right\} \cap \left\{n_2\right\} = \emptyset$: Controllable and observable sub-systems that are block diagonally decoupled or those which can be block diagonally decoupled, fall under this category. This situation is easy to resolve. An output feedback from one station will not affect the uncontrollable and/or unobservable modes from that station. Since the same modes are controllable and observable from the other station, we can design local dynamic compensators of appropriate order as discussed in Chapter V, that would assign the closed-loop eigenvalues arbitrarily close to the desired locations in the complex plane (subject to complex-conjugate pairing)

Example 1 : To illustpate the above case consider the system described by

$$\dot{x}(t) = \begin{bmatrix} 5 & 0 & 0 & 0 & 0 \\ 1 & 3 & 0 & 1 & 0 \\ 0 & 0 & 2 & 1 & 0 \\ 1 & 0 & 0 & 4 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix} x(t) + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} u_1(t) + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \\ 0 & 0 \end{bmatrix} u_2(t) \tag{5.6.3a}$$

$$y_1(t) = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix} x(t) \tag{5.6.3b}$$

$$y_2(t) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix} x(t). \tag{5.6.3c}$$

For the system (5.6.3) described above, $\left\{n_1\right\} = \left\{1,2\right\}$ and $\left\{n_2\right\} = \left\{3,4,5\right\}$. The output feedback from the first station affects only two of the eigenvalues (located at 1 and 2), while the output feedback from the second station affects the remaining three eigenvalues at 3,4 and 5. Therefore we can design a compensator to relocate the eigenvalues such that there is no interaction between the two stations *i.e.*, the eigenvalues assigned by output feedback from either of the station will not affect those assigned by the other station. For example with constant gain feedback matrices $K_1 = \begin{bmatrix} -2 & 6 \\ 0 & -4 \end{bmatrix}$ and $K_2 =$

$\begin{bmatrix} -20.71875 & 16.40625 \\ -19.68750 & 27.56250 \end{bmatrix}$, we can assign the closed-loop eigenvalue of the system at -1, $-2, -3, -4$ and -5 where the output feedback from the first station assigns eigenvalues at -1 and -2 without affecting the other three eigenvalues and the output feedback from the second station assigns the remaining three eigenvalues at -3, -4 and -5, without affecting the previously assigned ones.

Case 2 - $\left\{n_1\right\} \cap \left\{n_2\right\} \neq \emptyset$: This is perhaps, a more realistic situation. It is easy to see that if we assign (say) n_1 eigenvalues by dynamic output feedback from the first station, then, while assigning the remaining eigenvalues from the second station, we would also affect all the eigenvalues of the set $\left\{n_1\right\} \cap \left\{n_2\right\}$. In the worst case when the system is controllable and observable from both stations, feedback from the second station will affect all the previously assigned eigenvalues.

Example 2 : If the data in Example 1 is modified slightly to the following:

$$\dot{x}(t) = \begin{bmatrix} 5 & 0 & 0 & 0 & 0 \\ 1 & 3 & 0 & 1 & 0 \\ 0 & 0 & 2 & 1 & 0 \\ 1 & 0 & 0 & 4 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix} x(t) + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} u_1(t) + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \\ 0 & 0 \end{bmatrix} u_2(t) \tag{5.6.4a}$$

$$y_1(t) = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix} x(t) \tag{5.6.4b}$$

$$y_2(t) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix} x(t) \tag{5.6.4c}$$

where, we have changed the (3,1) element of the input matrix of the first station from 0 to 1, giving $\left\{n_1\right\} = \left\{1,2,3\right\}$ and $\left\{n_2\right\} = \left\{3,4,5\right\}$. Both stations can now stabilize *at least* three eigenvalues by means of local output feedback. Note that after the feedback from the first station the system may become completely controllable and/or observable from the second station. Therefore, it is necessary to sacrifice some freedom in designing the compensator in order to retain the previously assigned modes. By means of an output feedback $K_1 = \begin{bmatrix} -16 & -43 \\ 8 & -23 \end{bmatrix}$, we can assign the three controllable and observable modes from the first station at -1, -2 and -3. Then in order to preserve these assigned values and assign the remaining two eigenvalues at -4 and -5, a dynamic compensator of second order will enable us to assign the complete set of eigenvalues at the desired locations. The parameters of one such dynamic compensator are given by the following equations:

$$u_2(t) = v_2(t) - \phi_2(t) \qquad (5.6.5a)$$

$$\phi_2(t) = \begin{bmatrix} 5.1084 & 2.0055 \\ -10.858 & -4.4890 \end{bmatrix} z_2(t) + \begin{bmatrix} 2.4441 & 3.5018 \\ -11.477 & -7.9749 \end{bmatrix} y_2(t) \qquad (5.6.5b)$$

$$\dot{z}_2(t) = \begin{bmatrix} 25.306 & 10.630 \\ -4.4961 & -5.9269 \end{bmatrix} z_2(t) + \begin{bmatrix} 19.291 & 19.585 \\ -9.9409 & -9.8286 \end{bmatrix} y_2(t) \qquad (5.6.5c)$$

For the low order, two station examples considered above, one could easily analyze which modes can be assigned from which station and proceed with the design. However, in general, when N becomes considerably larger, a systematic procedure must be devised that would enable us to assign the eigenvalues at the desired locations by means of a decentralized compensator. In the next section, we develop such a procedure.

5.6.3. Main Results

Consider a single-station system $(A, \overline{B}, \overline{C})$ where $A \in \mathbb{R}^{n \times n}$, $\overline{B} \in \mathbb{R}^{n \times m}$ and $\overline{C} \in \mathbb{R}^{p \times n}$. Assume that the input matrix $\overline{B}$ and the output matrix $\overline{C}$ are block partitioned as $\overline{B} = \begin{bmatrix} B_1 \, B_2 \cdots B_N \end{bmatrix}$ and $\overline{C}^T = \begin{bmatrix} C_1^T & C_2^T & \cdots & C_N^T \end{bmatrix}$. If the total dynamic compensation from all stations to achieve some desired set of closed-loop eigenvalues requires a dynamic compensator of order $r = \sum_i^N r_i$, then the problem of EVA may be restated as follows. For the system

$$\begin{bmatrix} \dot{x}(t) \\ \dot{z}(t) \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} + \begin{bmatrix} \overline{B} & 0 \\ 0 & -I_r \end{bmatrix} \begin{bmatrix} u(t) \\ \phi(t) \end{bmatrix} \qquad (5.6.6a)$$

$$\begin{bmatrix} y(t) \\ z(t) \end{bmatrix} = \begin{bmatrix} \overline{C} & 0 \\ 0 & I_r \end{bmatrix} \begin{bmatrix} x(t) \\ z(t) \end{bmatrix}, \qquad (5.6.6b)$$

where $z(t) \in \mathbb{R}^r$, find a constant gain feedback matrix $K \in \mathbb{R}^{(m+r)\times(p+r)}$ of the form

$$K = \begin{bmatrix} J_1 & & & H_1 & & \\ & \ddots & & & \ddots & \\ & & J_N & & & H_N \\ G_1 & & & F_1 & & \\ & \ddots & & & \ddots & \\ & & G_N & & & F_N \end{bmatrix} \tag{5.6.7}$$

such that the resulting augmented closed-loop system has its $n+r$ eigenvalues at desired locations. Permuting the (block) rows and (block) columns of the feedback matrix K, we can easily obtain $\overline{K} = block\ \ diag.\ K_i$ with the following structure:

$$K_i = \begin{bmatrix} J_i & H_i \\ G_i & F_i \end{bmatrix}, i=1, \cdots, N. \tag{5.6.8}$$

Therefore, if we can determine the order of the compensators *a priori*, we can replace the problem of EVA by means of decentralized dynamic output feedback by N problems of EVA by means of single station dynamic output feedback. It should be pointed out here that these single station output feedback problems *may* or *may not* be independent of each other as was shown by examples in the previous section. Further, it may be possible that *all* stations considered independently may not form a disjoint partition, however, grouped together as $\left\{i_1, \cdots i_{r_1}\right\}, \left\{i_{r_1+1}, \cdots i_{r_2}\right\}, \cdots, \left\{i_{r_{s-1}+1}, \cdots i_{r_s}\right\}$ where $\sum_{i=1}^{s} r_i = N$, might still form disjoint partitions. The above partition is formed from the stations that *do not* affect each other when a local dynamic output feedback is applied from them (Case 1 of the previous section). From the definition of partitions, it is clear that no two partitions have a common station. Once we have obtained such partition, if it exists, then we have several decentralized EVA problems described in Case 2 of the previous section.

The algorithm for EVA in decentralized system then consists of the following two major steps:

1. Obtaining the disjoint partitions $\left\{i_1, \cdots i_{r_1}\right\}, \left\{i_{r_1+1}, \cdots i_{r_2}\right\}, \cdots$

$\left\{ i_{r_{s-1}+1}, \cdots i_{r_s} \right\}$ if they exist.

2. Designing a dynamic output feedback from each of the above partitions such that collectively, the feedback from the various stations inside each partition enables us to control the complete decentralized system.

The first step above is a preliminary examination of the given system. The partitions can be obtained in a systematic manner by examining the controllability and observability of the mode of open-loop the system from each station. This is easily achieved by using Algorithm 5.2.

Formally, we may state the algorithm for finding the disjoint sets $\left\{ i_1, \cdots i_{r_1} \right\}$, $\left\{ i_{r_1+1}, \cdots i_{r_2} \right\}$, . . . , $\left\{ i_{r_{s-1}+1}, \cdots i_{r_s} \right\}$ for the given decentralized system (A, B_i, C_i), $i=1, \cdots, N$, as follows:

Algorithm 5.6.1 : (Find Disjoint Partition)

Step I : (Initialization)

(1) Find a unitary transformation U_0 such that $\left(F^0, G_i^0, H_i^0\right) = \left(U^{0\,H} A U^0, G_i^0, U^{0\,H} B_i, H_i^0\right)$ is in UHF.

(2) Set n =order of the system.

(3) Set r =0

Step II : (Check controllability and observability of eigenvalues)

(1) Determine a unitary matrix U^r using the algorithm EXCHNG [24] such that $F^r = U^{r\,H} F^{r-1} U_p$ with λ_r (the r-th eigenvalue of the system) at (1,1) location.

(2) Complete the similarity transformation on the system, setting $\overline{G}_i^r = U^{r\,H} G_i^{r-1}$ and $\overline{H}_i^r = H_i^{r-1} U^r$ $i=1, \cdots, N$.

(3) Examine the first column of the matrices $\overline{H}_i^r$. If the first column of the output matrix of a certain station is a *zero* vector, λ_r is unobservable from that station. Store information in an array as shown in the tableau below (o in a block stands for observable modes while c stands for controllable modes).

(4) Determine a unitary matrix V^r such that $F^r = V^{r\,H} \overline{F}^r V^r$ with λ_r at $(n\,,n\,)$ location.

(5) Complete the similarity transformation on the system, setting $G_i^r = V_{r\,,H} \overline{G}_i^r$ and $H_i^r = \overline{H}_i^r V_r \;\; i=1, \cdots ,N$.

(6) Examine the last row of matrices G_i^r. If it is a *zero* vector, λ_r is uncontrollable from the i-th station. Store information in an array (for the sake of illustration, the array given below is for a 8-th order 3 station system):

	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8
Station 1	co	co		c		o		
Station 2	o			co	co	co		
Station 3			co				co	co

(7) If $r = n$, GO TO Step III, else set $r = r+1$, GO TO Step II(1)

Step III: (Find Disjoint Partitions if they exist)

(1a) Set Σ = set of all stations to be partitioned.

(1b) Set $s=1,\; j=1$,

(2) Form a set Σ_j corresponding to all stations from which λ_s is controllable and/or observable, assign the remaining stations to set Σ_{j+1}.

(3)

If $\Sigma_{j+1} = \emptyset$, GO TO Step IV (1).

(4) If λ_s is controllable and/or observable from any station $\in S_i$; $i=1, \cdots ,n$, Set $\Sigma_j = \Sigma_j \oplus S_i$ and $\Sigma_{j+1} = \Sigma_{j+1} \ominus S_i$.

(5) If $\Sigma_{j+1} = \emptyset$, GO TO Step IV (2).

If $s < n$, set $s = s+1$, GO TO (4)

If $s = n$, Set $n =$ number of modes in Σ_{j+1}, $\Sigma =$ all stations in Σ_{j+1}.

Set $j = j+2$, GO TO (1b).

Step IV: (Exit)

(1) No disjoint partition exists.

(2) The sets Σ_i, $i = 1, \cdots, j+1$ are the required disjoint partitions.

At the end of the algorithm, we will have the required disjoint partitions. It is worth noting that the EVA problem has now been decoupled to $j+1$ EVA problems of the kind described in Case 2 of the previous section.

We next show how EVA can be carried out from within each of the partitions. The method is best illustrated by means of an example. We consider a partition that has three stations, (A, B_i, C_i), $i = 1,2,3$, such that the output feedback from one station affects controllability and observability of all other stations. Assume that the system is of 10-th order and each station has 2 inputs and 2 outputs. The following steps then give the required dynamic output feedback.

Step I:

(1) Find controllable and observable subsystem $(A_{11}^1, B_{11}^1, C_{11}^1)$, from the first station (observable followed by controllable). Let the dimension of minimal-order subsystem be 4.

(2) Augment the controllable and observable subsystem and solve the constant gain output feedback problem for the augmented system $\left(\begin{bmatrix} A_{11}^1 & 0 \\ 0^T & 0 \end{bmatrix}, \begin{bmatrix} B_{11}^1 & 0 \\ 0^T & -1 \end{bmatrix}, \begin{bmatrix} C_{11}^1 & 0 \\ 0^T & 1 \end{bmatrix} \right)$.

(3) Define the parameter of resulting dynamic compensator by partitioning K_1 as $\begin{bmatrix} J_1 & h_1 \\ g_1^T & f_1 \end{bmatrix}$.

Comment: After applying the feedback, the augmented system is of order 11.

Step II :

(1) Find controllable and observable subsystem $\left(A_{11}^{2}, B_{11}^{2}, C_{11}^{2}\right)$, of the *augmented* closed-loop system from the second station (observable followed by controllable). Let the dimension of minimal-order subsystem be 8 (say) with three of the eigenvalues being the same as the ones assigned from the previous station.

Comment: A fifth order dynamic compensator would enable us to assign all eigenvalues of the minimal-order subsystem.

(2) Augment the controllable and observable subsystem and solve the constant gain output feedback problem for the augmented system $\left(\begin{bmatrix} A_{11}^{2} & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} B_{11}^{2} & 0 \\ 0 & -I_5 \end{bmatrix}, \begin{bmatrix} C_{11}^{2} & 0 \\ 0 & I_5 \end{bmatrix}\right)$.

(3) Define the parameter of resulting dynamic compensator by partitioning K_2 as $\begin{bmatrix} J_2 & H_2 \\ G_2 & F_2 \end{bmatrix}$.

Comment 1 : After applying the feedback, the augmented system is of order 16.

2 : Note that output feedback from the second station would affect all the controllable and observable eigenvalue that may have been previously assigned by the first station. Therefore these eigenvalues must be assigned from the second station as well.

3 : The augmented closed-loop system now has: 13 eigenvalues where they were assigned from the second station, 2 eigenvalues at the location where they were assigned from the first station and 1 eigenvalue at the same location as the open-loop system.

Step III :

(1) Find controllable and observable subsystem $\left(A_{11}^{3}, B_{11}^{3}, C_{11}^{3}\right)$, of the *augmented* closed-loop system from the second station (observable followed by controllable). Let the dimension of minimal-order subsystem be 3 (say) with one of the eigenvalues being the same as that of open loop system.

Comment: Note that $m_3+p_3 > 3$. Therefore a constant gain output feedback would be sufficient to assign all remaining eigenvalues.

(2) Find a constant gain output feedback matrix K_3 to assign the eigenvalues of the 3-rd order system $\left(A_{11}^{3}, B_{11}^{3}, C_{11}^{3}\right)$.

At the end of the third step, the augmented closed-loop system (16-th order), will have all its eigenvalues at the desired locations. Note that if from the final station, the dimension of the controllable and observable subsystem was greater than 3, we would require a dynamic output feedback.

The algorithm presented above employs repeated application of (1) Algorithms for computation of minimal-order subsystems of a given system and (2) computation of dynamic output feedback. Both of these algorithms use orthogonal state coordinate transformations and therefore, have good numerical properties.

5.7. CONCLUDING REMARKS

In this chapter, we presented a new characterization of d.f.m's of linear multivariable systems. Based on the characterization, an efficient and reliable method was proposed to compute d.f.m.'s. The computational method is numerically stable and uses software available in various scientific programming packages such as IMSL, EISPACK, LINPACK, *etc*. It was shown that the proposed method has significant advantages over existing methods from the point of view of computational efficiency and numerical stability. Numerical experiments carried out so far suggest that this approach is numerically more reliable than existing methods for computing d.f.m.'s.

For the systems that do not have d.f.m.'s, a computational procedure for eigenvalue assignment by means of decentralized output feedback was presented. The algorithm for decentralized EVA can be considered to be a generalization of the results in the previous chapter. In the decentralized EVA problem, unless the stations are block decoupled, as discussed in Section 5.6.2, an output feedback from one station would affect some or all of the eigenvalues assigned from other stations. As a first step in solving the decentralized EVA, we divided all the stations into several partitions such that feedback from any station in one partition would not affect the eigenvalues assigned from a station in other partitions. The second step consists of systematically assigning the eigenvalues from the stations within a partition. Computationally the decentralized EVA consists of repeated applications of reduction of a system to its minimal order sub-system, followed by applying the algorithm for dynamic output feedback in a single station system.

5.8. REFERENCES

[1] G.S. West-Vukovich, E.J. Davison and P.C. Hughes, "The Decentralized Control of Large Flexible Space Structures", *IEEE Trans. Autmat. Control*, Vol. AC-29, pp. 866-879, 1984.

[2] E.J. Davison and N. Tripathi, "The Optimal Decentralized Control of a Large Power System: Load and Frequency Control", *IEEE Trans. Automat. Control*, Vol. AC-23, pp. 312-323, 1978.

[3] S.H. Wang and E.J. Davison, "On Stabilization of Decentralized Control Systems", *IEEE Trans. Autmat. Control*, Vol. AC-18, pp. 473-478, 1973.

[4] V.A. Armentano and M.G. Singh, "A Procedure to Eliminate Decentralized Fixed Modes", *IEEE Trans. Autmat. Control*, Vol. AC-27, pp. 258-263, 1982.

[5] E.J. Davison and Ü. Özgüner, "Characterization of Decentralized Fixed Modes for Interconnected Systems", *Automatica*, Vol. 19, pp. 169-182, 1983.

[6] R.V. Patel, "On Blocking Zeros in Linear Multivariable Systems", *IEEE Trans. Autmat. Control*, Vol. AC-31, pp. 239-241, 1986.

[7] B.T. Smith, *et al*, *Matrix Eigensystems Routines - EISPACK Guide*, Lecture notes in Comp. Sci., Vol. 6, Springer Verlag, N.Y., 1976.

[8] B.S. Garbow, *et al.*, *Matrix Eigensystems Routines - EISPACK Guide Extension*, Lecture notes in Comp. Sci., Vol. 51, Springer Verlag, N.Y., 1977.

[9] J.J. Dongarra, *et al.*, *LINPACK Users' Guide*, Philadelphia, SIAM, P.A., 1979.

[10] B.D.O. Anderson and D.J. Clements, "Algebraic Characterization of Fixed Modes in Decentralized Control", *Automatica*, Vol. 17, pp. 703-712, 1981.

[11] J.P. Corfmat and A.S. Morse, "Decentralized Control of Linear Multivariable Systems", *Automatica*, Vol. 12, pp. 479-495, 1976.

[12] E.J. Davison and S.H. Wang, " A Characterization of Decentralized Fixed Modes in Terms of Transmission Zeros", *IEEE Trans. Autmat. Control*, Vol. AC-30, pp. 81-85, 1985.

[13] B.D.O. Anderson, "Transfer Function Description of Decentralized Fixed Modes", *IEEE Trans. Autmat. Control*, Vol. AC-27, pp. 1176-1182, 1982.

[14] H.. Seraji, "On Fixed Modes in Decentralized Control Systems", *Int. J. Control*, Vol. 35, pp. 775-784, 1982.

[15] R.V. Patel and P. Misra, "A Numerical Test for Transmission Zeros with Applications in Characterizing Decentralized Fixed Modes", *Proc. 23rd IEEE Conf. on Dec. and Control*, Las Vegas, 1984.

[16] M. Tarokh, "Fixed Modes in Multivariable Systems using Constrained Controllers", *Automatica*, Vol. 21, pp. 495-498, 1985.

[17] P. Misra and R.V. Patel, "Characterization and Computation of Decentralized Fixed Modes of Multivariable Systems", *Proc. American Contr. Conf.*, pp. 427-432, Seattle, WA, 1984.

[18] P. Misra and R.V. Patel, "Characterization and Numerical Computation of Decentralized Fixed Modes of Multivariable Systems", accepted for publication in *Automatica*.

[19] H.H. Rosenbrock, *Computer-Aided Control Systems Design*, Academic Press, London, 1974.

[20] C.C. Paige, "Properties of Numerical Algorithms Related to Computing Controllability", *IEEE Trans. Autmat. Control*, Vol. AC-26, pp. 130-138, 1981.

[21] R.V. Patel, "Computation of Minimal-Order State-Space Realizations and Observability Indices using Orthogonal Transformations", *Int. J. Control*, Vol. 33, pp.

227-246, 1981.

[22] P. Van Dooren, "The Generalized Eigenstructure Problem in Linear Systems Theory", *IEEE Trans. Autmat. Control*, Vol. AC-26, pp. 111-129, 1981.

[23] G.W. Stewart *Introduction to Matrix Computations*, Academic Press, N.Y., 1973.

[24] J.H. Wilkinson *The Algebraic Eigenvalue Problem*, Oxford University Press, 1965.

[25] S. Richter and R. DeCarlo, "A Homotopy Method for Eigenvalue Assignment Using State Feedback", *IEEE Trans. Autmat. Control*, Vol. AC-29, pp. 148-158, 1984.

[26] M. Vidyasagar and N. Viswanadham, "Algebraic Characterization of Decentralized Fixed Modes and Pole Assignment", *Proc. 21st IEEE Conf. on Dec. and Control*, Florida, 1982.

CHAPTER VI

TRANSFER FUNCTION AND FREQUENCY RESPONSE MATRICES OF MULTIVARIABLE SYSTEMS

In this chapter, we consider the problems of computation of transfer function and frequency response matrices of linear multivariable systems described by their state-space equations. A determinant identity is used to compute the above matrices which play an important role in frequency domain analysis and design of linear multivariable systems [1-6]. The algorithms proposed here are considerably faster and at least as accurate as the more reliable ones in the literature. The layout of the chapter is as follows: Section 6.1 states the problems of computation of transfer function and frequency response matrices and contains an overview of some of the existing methods for solving them. Section 6.2 introduces a determinant identity and describes the theoretical basis of the algorithms. Computation of transfer function matrices of linear multivariable systems together with various computational issues and numerical examples is considered in Section 6.3. Section 6.4 contains the algorithm and numerical examples for evaluating the frequency response matrices. A brief discussion of the results is presented in Section 6.5.

6.1. INTRODUCTION

Many so called classical and modern control system design methods for linear time-invariant systems *e.g.* see [1-6], use frequency response characteristics and transfer function matrices to design controllers which achieve desired stability and robustness properties for the resulting closed-loop systems. Efficient and accurate computation of frequency response and transfer function matrices is therefore of considerable importance. We consider a linear time-invariant, multivariable system described by its state-space equations:

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{6.1.1a}$$

$$y(t) = Cx(t) + Eu(t) \tag{6.1.1b}$$

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$ and $y(t) \in \mathbb{R}^p$. The transfer function matrix $W(s)$ of the system (A,B,C,E) is given by

$$W(s) = C\left(sI_n - A\right)^{-1}B + E \tag{6.1.2}$$

and it's frequency response matrix $W(j\omega)$ by

$$W(j\omega) = C\left(j\omega I_n - A\right)^{-1}B + E. \tag{6.1.3}$$

One direct method of computing $W(s)$ is to determine the resolvent matrix $\left(sI_n - A\right)^{-1}$. However, when the order of the system is high, this approach is prone to numerical round-off errors. Using an alternative approach proposed in [7,8], which we shall refer to as the *pole-zero* approach, the coefficients of the denominator polynomial of the (i,j)-th element of $W(s)$ are determined by computing the eigenvalues [9,10] of the state matrix of the controllable and observable subsystem corresponding to the (j,i)-th input-output pair. The coefficients of the corresponding numerator polynomial are obtained by solving a generalized eigenvalue problem [11,12] and finding a constant multiplier. The accuracy of this scheme depends on the accuracy of the computed eigenvalues and generalized eigenvalues. For systems with ill-conditioned eigenvalue problems or ill-conditioned generalized eigenvalue problems, the computed coefficients of the numerators and denominators of transfer function elements can be very inaccurate. These difficulties can be avoided by computing the coefficients directly [13,14]. In subsequent sections, we present an efficient and reliable technique for doing this.

Computation of frequency response matrices usually requires evaluation of $W(j\omega_k)$ at a large number of frequencies ω_k, $k=1, \cdots, N$. If the system description is given in terms of the transfer function matrix $W(s)$, then the computation of frequency response is a relatively simple matter. However, if the state-space description (A,B,C,E) is given, then the problem is not so straightforward computationally. Obtaining frequency response by first converting the state-space description to a transfer function description is justifiable only when the initial cost of computing the transfer function matrix is offset by the number of frequencies [15,16] at which the frequency response is desired. From the operations count in Section 6.4., it is possible to determine approximately when a direct determination of frequency response would be more economical than computing the transfer function matrix followed by evaluating the transfer function matrix at various frequencies. In [17], a method for computing the frequency response was proposed which, starting from a given state-space description, determines the frequency response matrix at a given value of ω by first reducing the state matrix to an upper Hessenberg matrix and then solving a system of n simultaneous linear equations. Depending on the number of frequencies at which the frequency response matrix is desired, the methods described in subsequent sections are comparable to or more efficient than the method in [17].

6.2. PRELIMINARY CONSIDERATIONS

Here we introduce an important determinant identity and show how it can be used for developing our algorithms.

Fact 6.1 : For a single input, single output system (A, b, c^T), we may write [5,13-16],

$$\det\left(sI_n - A + b\ c^T\right) = \det\left(sI_n - A\right) + c^T \operatorname{adj}\left(sI_n - A\right) b. \tag{6.2.1}$$

Further,

$$c^T \left(sI_n - A\right)^{-1} b = \frac{c^T \operatorname{adj}\left(sI_n - A\right) b}{\det\left(sI_n - A\right)}$$
$$= \frac{\det\left(sI_n - A + b\ c^T\right)}{\det\left(sI_n - A\right)} - 1. \tag{6.2.2}$$

Now, in (6.1.2), the (i,j)-th element of $W(s)$ is given by

$$w_{ij}(s) = c_i^T \left(sI_n - A\right)^{-1} b_j + e_{ij} \tag{6.2.3}$$

which may be written, using (6.2.2) as

$$w_{ij}(s) = \frac{\det\left(sI_n - A + b_j c_i^T\right)}{\det\left(sI_n - A\right)} - 1 + e_{ij}. \qquad (6.2.4)$$

Similar results can be stated when s has been assigned a value $j\omega$.

Fact 6.2 : The determinant of a matrix whose k-th column can be expressed as a sum of column vectors $a_k + \hat{a}_k$, may be written as [10],

$$\det\left(a_1\ a_2 \cdots a_k + \hat{a}_k \cdots a_n\right) = \det\left(a_1\ a_2 \cdots a_k \cdots a_n\right) + \det\left(a_1\ a_2 \cdots \hat{a}_k \cdots a_n\right). \qquad (6.2.5)$$

A similar result can also be stated for the matrices whose k-th row can be represented as sum of two row vectors.

Fact 6.3 : The characteristic polynomial of an upper Hessenberg matrix A can be computed efficiently using the following recurrence relations [18]:

Define $u_n(s)$=1.0 and compute the vector $\left[u_1(s)\ u_2(s)\ \cdots\ u_n(s)\right]$ by using the recurrence relation:

$$u_k(s) = \frac{1}{a_{k+1,k}}\left[\left(s - a_{k+1,k+1}\right)u_{k+1}(s) - \sum_{l=k+1}^{n} a_{k+1,l+1}u_{l+1}(s)\right], \qquad (6.2.6a)$$

the characteristic polynomial is then given by

$$p(s) = \left[\left(s - a_{11}\right)u_1(s) - \sum_{r=2}^{n} a_{1,r}\,u_r(s)\right]. \qquad (6.2.6b)$$

6.3. Computation of Transfer Function Matrices

The proposed method determines one element of the transfer function matrix at a time. Assume that a controllable and observable subsystem $\left(A, b_j, c_i^T, e_{ij}\right)$ is in its UHF. Then (6.2.4) may be written as

$$w_{ij}(s) = \frac{\det\left(\bar{A} + b_j c_i^T\right) - \det\left(\bar{A}\right)}{\det\left(\bar{A}\right)} + c_{ij} \qquad (6.3.1)$$

where $\tilde{A} = \left(sI_n - A\right)$. Note that because of the structure of b_j, the product $b_j c_i^T$ will have only its first row as a non-zero row, given by $b_{1j} c_i^T$. Next, we define $\hat{A}$ as the matrix obtained from $\tilde{A}$ by replacing its first row by $b_{1j} c_n^T$, where b_{1j} is the first element of b_j. Then, using (6.2.5), we have

$$\begin{aligned} w_{ij}(s) &= \frac{\det\left(\hat{A}\right)}{\det\left(\tilde{A}\right)} + e_{ij} \qquad (6.3.2) \\ &= \frac{\hat{n}_{ij}(s) + e_{ij}\, d_{ij}(s)}{d_{ij}(s)} \\ &\triangleq \frac{n_{ij}(s)}{d_{ij}(s)} \end{aligned}$$

where $\hat{n}_{ij}(s) \triangleq \det\left(\hat{A}\right)$ and $d_{ij}(s) \triangleq \det\left(\tilde{A}\right)$.

6.3.1. An Algorithm for Computing Transfer Function Matrices

The determinant identity described above will be used in Algorithm 6.1 to compute the umerator and denominator polynomials of individual elements of the transfer function matrix. However, since the single-input, single-output subsystems $\left(A, b_j, c_i^T, e_{ij}\right)$ may not be controllable and/or observable for each input-output pair, we must remove the input and/or output decoupling zeros before computing the "transfer function elements" corresponding to that input-output pair. The algorithm to compute the transfer function matrix therefore consists of (1) removal of input and/or output decoupling zeros for the (j,i)-th input-output pair and (2) computation of the transfer function element corresponding to the controllable and observable subsystem of $\left(A, b_j, c_i^T, e_{ij}\right)$. The first step ensures that there will be no cancellation in the numerator and denominator polynomials of the (i,j)-th element of the transfer function matrix. An algorithm for computation of a transfer function matrix may be formally described as follows:

Algorithm 6.1: (Transfer Function of Linear Multivariable Systems)

Step I: [Initialization]

(1) Set $i=0$, $n_c=0$, $n_o=0$.

(2) Set $i=i+1$, $j=0$, n =order of the given multi-input, multi-output systems.

(3). Set $\left(\hat{F},\hat{G},\hat{h}^T,e^T\right) = \left(A,B,c_i^T,e_i^T\right)$.

Step II: [Remove output decoupling zeros]

(1) Reduce the single-input, multi-output "dual" system $\left(\hat{F}^T,\hat{h},\hat{G}^T\right)$ to UHF.

Comment : This can always be done by means of an orthogonal state coordinate transformation matrix $U \in \mathbb{R}^{n \times n}$ such that

$$U^T \hat{F}^T U = \begin{bmatrix} \hat{F}^{(o)} & * \\ 0 & \hat{F}^{(uo)} \end{bmatrix},$$

$$U^T \hat{h} = \begin{bmatrix} \hat{h}^{(o)} \\ 0 \end{bmatrix}$$

and

$$\hat{G}^T U = \left[\hat{G}^{(o)} \; \hat{G}^{(uo)}\right]$$

(2) Partition the system and set $\left(\bar{F},\bar{G},\bar{h},e\right) = \left(\hat{F}^{(o)},\hat{G}^{(o)\,T},\hat{h}^{(o)\,T},e\right)$ and $n_o = \dim\left(\hat{F}^{(o)}\right)$.

Comment : Note that the dimension of the observable system is easily determined by inspecting the sub-diagonal of the upper Hessenberg matrix $U^T \hat{F}^T U$. The partition is performed if any of the elements on the sub-diagonal is zero. For a completely observable system, $n_o = n$.

Step III: [Remove input decoupling zeros]

(1) Set $j=j+1$ and $\left(\bar{F},\bar{g},\bar{h}^T,e\right) = \left(\bar{F},\bar{g}_j,\bar{h}^T,e_{ij}\right)$.

(2) Reduce the system $\left(\bar{F},\bar{g},\bar{h}^T,e\right)$ to UHF by means of an orthogonal transformation matrix $\bar{U} \in \mathbb{R}^{n_o \times n_o}$ such that

$$U^T \tilde{F} U = \begin{bmatrix} \tilde{F}^{(c)} & * \\ 0 & \tilde{F}^{(uc)} \end{bmatrix},$$

$$U^T \tilde{g} = \begin{bmatrix} \tilde{g}^{(c)T} & 0^T \end{bmatrix}^T$$

and

$$\tilde{h}^T U = \begin{bmatrix} \tilde{h}^{(c)T} & \tilde{h}^{(uc)T} \end{bmatrix}.$$

(3) Partition the system and set $(F, g, h^T, e) = (\tilde{F}^{(c)}, \tilde{g}^{(c)}, \tilde{h}^{(c)T}, e)$ and $n_c = \dim(\tilde{F}^{(c)})$.

Comment : Note that at this stage we have isolated the controllable and observable subsystem of (A, b_j, c_i^T, e_{ij}) and therefore, we are in a position to evaluate the numerator and denominator polynomials of the corresponding subsystem.

Step IV: (Compute the transfer function element $w_{ij}(s)$)

(1) Compute the vector $\left[u_1(s)\ u_2(s)\ \cdots\ u_{n_c}(s)\right]$ using the recurrence relation:

$$u_k(s) = \frac{1}{f_{k+1,k}}\left[\left(s - f_{k+1,k+1}\right)u_{k+1}(s) - \sum_{l=k+1}^{n_c} f_{k+1,l+1}u_{l+1}(s)\right]$$

where $u_{n_c}(s) = 1.0$ and $f_{i,j}$ is the (i,j)-th element of F.

(2) Compute the denominator polynomial from

$$d_{ij}(s) = \left[\left(s - f_{11}\right)u_1(s) - \sum_{r=2}^{n_c} f_{1,r}u_r(s)\right].$$

(3) Evaluate $\hat{n}_{ij}(s)$ as

$$\hat{n}_{ij}(s) = \left[g_1 \sum_{r=1}^{n_c} h_r u_r(s)\right].$$

where h_r is the r-th element of h and g_1 is the first (the only nonzero) element of g.

(4) Compute the numerator polynomial from

$$n_{ij}(s) = \hat{n}_{ij}(s) + e_{ij}d_{ij}(s)$$

and set

$$w_{ij}(s) = \frac{n_{ij}(s)}{d_{ij}(s)}.$$

```
(5) If $j < m$, go to Step III - (1), else,
    If $i < p$ and $j = m$, go to Step I - (2), else,
    STOP .
```

At the end of the Algorithm, we will have the desired transfer function matrix $W(s)$.

6.3.2. Discussion

Algorithm 6.1 overcomes several problems associated with existing methods for computing transfer function matrices of linear multivariable systems. In this section, we discuss various properties and implementation issues associated with the algorithm.

(1). To obtain the minimal order subsystems corresponding to each input-output pair, only orthogonal similarity transformations are used. This is very desirable from the numerical point of view, because the round-off errors incurred by the use of finite precision arithmetic are well bounded for orthogonal transformations [9]. This aspect of computation of transfer function matrices has also been discussed in detail in [7,8].

(2). Unlike the pole-zero method, the proposed method is *direct*. It should be pointed out that the pole-zero method obtains the minimal order subsystem corresponding to each input-output pair in a similar manner. However, having obtained the minimal order subsystem, it is necessary to solve the *algebraic eigenvalue problem* for the corresponding state matrix to obtain the denominator polynomial, followed by a *generalized eigenvalue problem* to obtain the numerator polynomial. The methods for solving the algebraic and generalized eigenvalue problems are *iterative* in nature and are, therefore, computationally much more expensive. Moreover, if these problems for a given system are *ill-conditioned*, then the transfer function matrix computed using this approach may lead to inaccurate results.

The proposed algorithm works directly from the unreduced UHF of the minimal order system and therefore does not suffer from the shortcomings discussed above. Since the algorithm computes the determinant of a matrix in a single step, it is computationally inexpensive. Note that in the algorithm, we need to form products of the elements of the first subdiagonal of an upper Hessenberg matrix. If these elements are extremely small (large), one could run into floating point underflow (overflow) [9]. While implementing the algorithm, one can store the elements of $p_r(s)$ in a product form until the final step, thereby avoiding the floating point overflow or underflow.

(3). The bulk of the computational effort is involved in finding the minimal order subsystem for each input-output pair. From the algorithm, it is clear that we require p reductions of single output systems to their lower Hessenberg forms, followed by m reductions of the observable subsystems to their upper Hessenberg forms for each output. For a single input, single output system, these reductions require approximately $\frac{5}{3}n^3+8n^2 + \frac{5}{3}n_o^3+8n_o^2$ floating point operations, where n_o is the dimension of the observable subsystem from the specified output. The final step for computing the two determinants can be completed in approximately $\frac{1}{6}n_c^3+n_c^2$ floating point operations, where n_c is the dimension of the minimal order subsystem for a specified input-output pair. It should be pointed out that the pole-zero method *also* requires the initial reduction to Hessenberg form to obtain the minimal order subsystems and subsequently needs to solve the computationally expensive algebraic eigenvalue as well as generalized eigenvalue problems to obtain the elements of the transfer function matrix. It is safe to say that Algorithm 6.1 has significant advantage in computational cost over the pole-zero technique.

(4). The proposed method and the existing methods used for computing the coefficients of numerator and denominator polynomials of the elements of transfer function matrices are based on entirely different principles. Therefore, one cannot directly compare the performance of the two approaches. However, the computational costs discussed above and the numerical examples given below illustrate that the proposed method has significant advantages over the existing methods.

6.3.3. Numerical Examples

Here, we illustrate the accuracy of the proposed algorithm by means of a numerical example. All computations were carried out on a VAX 11/780 using double precision as well as single precision. The accuracy of the coefficients computed using single precision is shown by the underlined digits.

In this example we consider the 9-th order boiler model [11] with 2 inputs and 2 outputs. This is an extremely ill-conditioned system with eigenvalues ranging from $\sim 10^{-10}$ to ~ 10. The parameters of the system are given in Table 4.2. The pole-zero approach in this case will not be reliable because forming the coefficients with eigenvalues as far apart as mentioned will lead to cancellation of several significant places. The coefficient w_{11}, computed using the proposed algorithm with double precision and single precision shows that the algorithm is quite reliable. The underlined digits in the columns with coefficients computed using single precision give an idea of the accuracy of the algorithm (Table 6.1 and 6.2). For the sake of illustration, only the coefficients of $w_{11}(s)$ have been shown.

Powers of s	Double Precision	Single Precision
6	2.096991388799999d+02	$\underline{2.096991}$405487061d+02
5	2.240810395996802d+03	$\underline{2.2408103}$79028320d+03
4	8.058142994149724d+03	$\underline{8.058142}$883300781d+03
3	1.054129331589858d+04	$\underline{1.0541293}$21289063d+04
2	2.667315391676311d+03	$\underline{2.6673153}$07617188d+03
1	1.470991471959572d+02	$\underline{1.470991}$382598877d+02
0	1.189248279895743d+00	$\underline{1.189248}$189330101d+00

Table 6.1 Numerator polynomial of $w_{11}(s)$

Powers of s.	Double Precision	Single Precision
8	1.000000000000000d+00	1.000000000000000d+00
7	1.089330000000039d+01	1.089329999685287d+01
6	4.255744585600835d+01	4.255744528770447d+01
5	6.701215774012081d+01	6.701215744018555d+01
4	3.338350289656865d+01	3.338350939750671d+01
3	6.338532196643809d+00	6.338539093732834d+00
2	4.170298739139534d-01	4.170327670872211d-01
1	5.786203563399542d-03	5.786365509266034d-03
0	2.266130444584768d-05	2.266192825572944d-05

Table 6.2 Denominator polynomial of $w_{11}(s)$

6.4. Computation of Frequency Response Matrices

The method for computation of frequency response matrices described in this section determines one row of the matrix at a time *i.e.*, we evaluate the frequency response of multi-input, single-output systems $\left(A, B, c_i^T\right)$, $i = 1, \cdots, p$ where c_i^T is the i-th row of C. Note that for the sake of clarity, we have dropped the matrix E from the system description. It can be easily incorporated in each final frequency response matrix by a simple addition. In evaluating the frequency response, each triple described above is first reduced to the condensed form described in (2.3.3). This reduction is done only once for a given system.

The (i,l)-th element of the frequency response matrix $W(j\omega)$ in (6.1.3) with $E = 0$, is given by

$$w_{il}(j\omega) = c_i^T \left(j\omega I_n - A\right)^{-1} b_l$$

where b_l is the l-th column of B. Using the determinant identity (6.2.2), we can write

$$w_{il}(j\omega) = \frac{\det\left(j\omega I_n - A + b_l c_i^T\right)}{\det\left(j\omega I_n - A\right)} - 1 \tag{6.4.1}$$

Equation (6.4.1) can be further written as

$$w_{il}(j\omega) = \frac{\det\left(j\omega I_n - A + b_l c_i^T\right) - \det(j\omega I_n - A)}{\det(j\omega I_n - A)}$$

$$= \frac{\left(\tilde{a}_1, \cdots, \tilde{a}_{n-1}, \tilde{a}_n + b_l c_{in}\right) - \det\left(\tilde{A}\right)}{\det\left(\tilde{A}\right)} \tag{6.4.2}$$

where $\tilde{A} = (j\omega I_n - A)$; $\tilde{a}_k$ is the k-th column of $\tilde{A}$ and c_{in} is the n-th element of c_i and is the only non-zero element in that vector. Then, using the determinant identity in (6.2.5), the above expression can be simplified to

$$w_{il}(j\omega) = \frac{\det\left(\tilde{a}_1, \cdots, \tilde{a}_{n-1}, b_l c_{in}\right)}{\det\left(\tilde{A}\right)} = \frac{\det\bar{A}}{\det A} \tag{6.4.3}$$

$$= \frac{\bar{n}_{il}}{\tilde{d}} \tag{6.4.4}$$

From the LU decomposition of $\tilde{A} = \tilde{L}\,\tilde{U}$, we have $\det\left(\tilde{A}\right) = \det\left(\tilde{U}\right) = \prod_{r=1}^{n} \tilde{u}_{rr}$. It is perhaps worth pointing out the matrices $\tilde{A}$, $\tilde{L}$ and $\tilde{U}$ are respectively an UHM, a unit lower bi-diagonal matrix and upper triangular matrix. Moreover, changing the last column of $\tilde{A}$ affects only the last column of $\tilde{U}$. Therefore (6.4.4) gives

$$\begin{aligned} w_{il}(j\omega) &= \frac{\bar{n}_{il}}{\tilde{d}} \\ &= \frac{\bar{u}_{nn} \prod_{r=1}^{n-1} \tilde{u}_{r,r}}{\tilde{u}_{nn} \prod_{r=1}^{n-1} \tilde{u}_{r,r}} \\ &= \frac{\bar{u}_{nn}}{\tilde{u}_{nn}}. \end{aligned} \tag{6.4.5}$$

In (6.4.5), $\tilde{u}_{nn}$ is the (n,n)-th element of the matrix $\tilde{U}$ and $\bar{u}_{nn}$ is the (n,n)-th element of the matrix $\bar{U}$ in the LU decomposition of $\bar{A}$. Therefore, for evaluating the frequency response, instead of computing the determinants as in (6.4.2), we only need to find the ratio in (6.4.5). A small saving in computation can be achieved by noting that $\bar{u}_{nn} = \hat{u}_{nn} c_{in}$, where $\hat{u}_{nn}$ is the (n,n)-th element of the matrix $\hat{U}$ in the LU decomposition of $\hat{A} = \left(\tilde{a}_1, \cdots, \tilde{a}_{n-1}, b_l\right)$ The (i,l)-th element of the frequency response matrix is, therefore, given by

$$w_{il}(j\omega) = \frac{\hat{u}_{nn}}{\bar{u}_{nn}} c_{in} . \tag{6.4.6}$$

6.4.1. An Algorithm for Computing Frequency Response Matrices

This algorithm utilizes the fact that if a single-output system is reduced to the condensed form described in (2.3.3), then, one can easily partition the system into observable and unobservable subsystems. Since the unobservable subsystem does not contribute to the frequency response, it can be removed. An algorithm based on the above discussion may be formally given as follows.

Algorithm 6.2: (Frequency Response Matrices of Linear Multivariable Systems).

Step I: [Initialization]

(1) Set $i=0$, $k=0$, $l=0$ and $n_o=0$.

(2) Set $i=i+1$, $n=$order of the system (A,B,C).

(3) Set $\left(\hat{F},\hat{G},\hat{h}^T\right) = \left(A,B,c_i^T\right)$.

Step II: [Remove output decoupling zeros]

(1) Find an orthogonal transformation matrix $U \in \mathbb{R}^{n\times n}$ such that

$$\hat{h}^T U = \left[0\,0 \cdots 0\,\hat{h}_{1n}\right]$$

and set $\hat{h}^T=\hat{h}^T U$, $\hat{F}=U^T\hat{F}U$ and $\hat{G}=U^T\hat{G}$.

(2) Find an orthogonal transformation matrix $V \in \mathbb{R}^{n\times n}$ such that $V^T\hat{F}V$ is a upper Hessenberg matrix and the structure of $\hat{h}$ is preserved.

$$V^T\hat{F}V = \begin{bmatrix} \hat{F}^{(uo)} & * \\ 0 & \hat{F}^{(o)} \end{bmatrix},$$

$$V^T\hat{G} = \begin{bmatrix} \hat{G}^{(uo)} \\ \hat{G}^{(o)} \end{bmatrix}$$

and

$$\hat{h}^T V = \left[0^T \quad \hat{h}^{(o)\,T}\right]$$

(3) Partition the system and set $\left(F,G,h^T\right) = \left(\hat{F}^{(o)},\hat{G}^{(o)},\hat{h}^{(o)\,T}\right)$ and

$n_o = \dim \left(\hat{F}^{(o)}\right)$.

Comment : Note that the dimension of the observable system is easily determined by inspecting the first super-diagonal of the lower Hessenberg matrix $\hat{V}^T \hat{F} V$. For a completely observable system, $n_o = n$.

Step III: [Compute the frequency response vector $w_i(s)$]

(1) Set $k = k+1$

(2) Perform LU decomposition of $\left(j\omega_k I - F^{(i)}\right)$, denote by $\tilde{L}^{(k)} \tilde{U}^{(k)}$ and set $\tilde{u}_{nn}^{(k)} = (n,n)$-th element of $\tilde{U}^{(k)}$.

(3) Set $l = l+1$

(4) Replace the last column of $\left(j\omega_k I - F^{(i)}\right)$ by $g_l^{(i)}$ and denote it by $\hat{F}^{(i)}$, perform LU decomposition of $\hat{F}^{(i)}$ and set $\hat{u}_{nn}^{(k,l)} = (n,n)$-th element of $\hat{U}^{(k,l)}$

(5) Set $w_{il}(j\omega_k) = \dfrac{\hat{u}_{nn}^{(k,l)}}{\tilde{u}_{nn}^{(k)}} c_{in}$.

(6) If $l < m$ go to *Step III* (3), else

If $l = m$ and $k < N$, go to *Step III* (1), else

If $l = m$, $k = N$ and $i < p$, go to *Step I* (2), else

If $l = m$, $k = N$ and $i = p$, STOP.

At the end of the algorithm, we get the required frequency response matrix $W(j\omega)$ at N desired values of ω_k.

6.4.2. Discussion

(1) The triples $\left(F^{(i)}, G^{(i)}, h_i^T\right)$ are in the special condensed form described in (2.3.3). As shown in that section, only the last element of h_i^T is non-zero. Consequently, forming $\left(j\omega_k I_n - F^{(i)} + g_l^{(i)} h_i^T\right)$ retains the upper Hessenberg structure of $F^{(i)}$. Also, the matrix $\hat{U}^{(k,l)}$ differs from $\tilde{U}^{(k)}$ in only its last column. This enables us to compute the

frequency response matrix without actually calculating the determinants in (6.4.3), thereby, reducing the number of operations.

(2) The LU decomposition of an upper Hessenberg matrix $\hat{F}$ requires only $\frac{1}{2}n^2$ floating point operations. Once the lower sub-diagonal of $\hat{L}$ in the LU decomposition of $\hat{F}$ is known, subsequent evaluations of $\hat{u}_{nn}^{(k,l)}$ for all g_l, $l = 1, \cdots, m$, require only nm extra operations.

(3) An error analysis of the LU decomposition of a matrix F with L being a unit lower bidiagonal matrix yields

$$LU = F + \Delta F$$

where L and U are exact for a slightly perturbed matrix F. The elements of the error matrix ΔF satisfy [9,10]

$$|\Delta F_{ij}| \leq n\,\pi\beta\gamma 10^{-t}$$

where n is the order of the matrix, π is some constant of order unity, β is the largest element of the matrix F and $\gamma \leq 2^{n-1}$. Although 2^{n-1} appears to be a rapidly growing function, in practice for upper Hessenberg matrices, large growth factors γ are almost never encountered. Moreover, if inner products in the LU decomposition are accumulated in double precision, the factor n can be replaced by 1. The discussion above does not permit us to make a strong statement about the stability of the algorithm, but for all practical purposes, the results obtained from using the proposed algorithm will, in general, be very reliable.

(4) The problem of evaluating the frequency response matrix can be divided into p (=the number of outputs) independent sub-problems. Therefore, p processors may be employed to compute the frequency response. This will reduce the actual time of computation significantly. If, however, only one processor is used and if the number of inputs is smaller than the number of outputs, then computing the frequency response of the dual system would enable further reduction in computational effort. This will

become clear from the operationas count given later in this section.

(5) It is worth mentioning that any technique for efficient evaluation of the determinants of Hessenberg matrices may be used to determine frequency response matrices *e.g.* Hyman's method and its variations [9,19]. But such methods may run into floating point overflows as the value of ω increases. The proposed method as well as the method in [17] do not suffer from this drawback.

(6) The algorithm proposed above uses complex arithmetic. The use of complex arithmetic can be avoided by making minor modifications to the algorithm. The real and imaginary parts can be computed independently as described below.

Consider the triple (A, B, C); its frequency response matrix can be obtained by solving

$$(j\omega I - A)Z = B \tag{6.4.7}$$

for Z and then computing

$$G(j\omega) = CZ \tag{6.4.8}$$

Let $Z = Z_1 + jZ_2$, then equating the real and imaginary parts on both side of (6.4.7), we get

$$\begin{bmatrix} -A & -\omega I \\ \omega I & -A \end{bmatrix} \begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} = \begin{bmatrix} B \\ 0 \end{bmatrix}. \tag{6.4.9}$$

It is easy to see that in (6.4.9),

$$Z_1 = \frac{1}{\omega} A Z_2 \tag{6.4.10}$$

and

$$Z_2 = -\omega I \left(\omega^2 I + A^2\right)^{-1} B. \tag{6.4.11}$$

The term $\hat{G}\left(\omega^2\right) = \left(\omega^2 I + A^2\right)^{-1} B$ can be evaluated by applying Algorithm 6.2 to the system $\left(-A^2, B, I\right)$. Then,

$$Z_1 = -A\hat{G}\left(\omega^2\right)$$

and

$$Z_2 = -\omega \hat{G}\left(\omega^2\right).$$

Note that only real arithmetic is used in computing both Z_1 and Z_2. The frequency response is then given by $CZ_1 + jCZ_2$. Although the above approach uses only real arithmetic, in forming the matrix A^2, significant amount of information can be lost, and therefore, this alternative should be used only as the last resort.

(7) In the transformed triples $\left(F^{(i)}, G^{(i)}, h_i^T\right)$, the matrix $F^{(i)}$ is an unreduced upper Hessenberg matrix and $h_{in} \neq 0$ if and only if F is completely observable from the i-th output. However, if that is not the case, then $F^{(i)}$ will have a block upper triangular structure and the system equations may be rewritten as:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} F_{11} & F_{12} \\ 0 & F_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} u$$

$$y_i = \begin{bmatrix} 0 & h_i^T \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

where F_{22} is an unreduced upper Hessenberg matrix and $h_i^T = \begin{bmatrix} 0 \; 0 \cdots h_{in} \end{bmatrix}$. The observable subsystem is $\left(F_{22}, G_2, h_i^T\right)$ and the frequency response is given by

$$W(j\omega) = h_i^T \left(j\omega I - F_{22}\right)^{-1} G_2.$$

Since the system being considered now has order equal to the dimension of F_{22}, the computational effort is accordingly reduced.

6.4.3. Operations Count

Next we compare the operations count for several efficient methods for computing frequency response matrices. We consider three methods: (1) the method in [17], (2) the method proposed in this section and (3) by first computing the transfer function matrix and then evaluating it at various desired frequencies.

Method in [17]: In this method, for the given triple (A, B, C), the matrix A is transformed to an upper Hessenberg matrix while matrices B and C have no specific structure. An LU decomposition of $(j\omega I - A)$ is carried out and Z is obtained from $UZ = L^{-1}B$, where U and L are respectively upper triangular and unit lower bidiagonal triangular matrices. The frequency response for one value of ω is then given by $W(j\omega) = CZ$. When efficiently implemented, the above steps, together with an initial reduction of A to an upper Hessenberg form and the corresponding transformations on B and C, require approximately $\frac{5}{3}(n+m+p)n^2$ (real) and $\frac{1}{2}\left[(p+1)n^2+2nmp\right]N$ (complex) floating point operations (flops) for N values of ω.

Proposed Method: The proposed method requires an initial reduction of several multi-input, single-output systems to a condensed form. This reduction requires approximately $\frac{5}{3}(n+m+1)n^2p$ (real) operations. For each value of frequency, evaluation of $\tilde{u}_{nn}^{(i)}$ in Algorithm 6.2 requires $\frac{1}{2}n^2$ operations and subsequent m values of $\tilde{u}_{nn}^{(i)}$ for all inputs require a total of nm operations. This is done for each triple $\left(A, B, c_i^T\right)$. Therefore, $W(j\omega)$ can be evaluated in approximately $\frac{5}{3}(m+n+1)n^2p$ (real) and $\frac{p}{2}\left(n^2+2nm\right)N$ (complex) flops for N value of ω. Further saving can be achieved by considering the dual system if $p > m$, as can be easily seen from the expression above.

Considering remark (7), we note that the operations count given for Algorithm 6.2 above, corresponds to the case when the system is observable from each of the outputs. However, this is usually not the case for very high order systems.If a system is not observable from the i-th output, the frequency response calculations are carried out on a lower order subsystem and a significant saving in the computational effort can be achieved. To illustrate the above point, consider a 40-th order system with 5 inputs and 5 outputs and 100 frequency values. If each of the outputs can observe only 20 states, the method in [17] requires approximately 2,453,000 flops compared to 1,213,000 flops required by the proposed method.

By Evaluating the Transfer Function Matrix: Evaluating the transfer function matrix requires approximately $\frac{5}{3}\left(n^3+n_c^3\right)m + 8\left(n^2+n_c^2\right)mp + \left(\frac{1}{6}n_o^3+n_o^2\right)mp$ flops as discussed in Section 6.3. Consider the example introduced above where only 20 states are observable from each of the outputs. Further, assume that only 15 states are controllable from each of the inputs. Then evaluating the transfer function matrix will require approximately 560,000 flops. Evaluation of frequency u response matrices for 100 different values of ω requires a further n_c^2mpN flops. For the example under consideration, this is approximately 540,000 flops, giving a total of 1,100,000 flops. Note that this approach becomes extremely efficient if the number N is very large, because once the transfer function is known, it requires a very small number of computations for evaluating it at different values of ω.

The operations counts given above are only representative. In practice, frequency response matrices may be computed with slightly less or more computational effort: depending on the controllability and observability properties of the system under consideration.

6.4.4. Numerical Example

For the 9-th order boiler model considered in Section 6.3, the frequency response was calculated using the three approaches mentioned above. The frequency response was first calculated in double precision using the proposed method as well as the method in [17]. The results agreed up to the 15-th significant digit. We shall call the frequency response calculated in double precision as the "Actual" response. Next, the value for the (1,1) element of the frequency response matrix was obtained in single precision using the proposed method, using the transfer function obtained in the previous section and using the method in [17] for a selected number of frequencies The results for the three methods are shown in Tables 6.3-6.6 below. The underlined digits indicate the accuracy of the computations.

Actual	-1.764752693176270d+02 +7.363956117630005d+01i
via TFM	-1.764752655029297d+02 +7.363956165313721d+01i
Proposed	-1.764752702713013d+02 +7.363956117330102d+01i
Method [17]	-1.764752664566040d+02 +7.363956403732300d+01i

Table 6.3 (1,1) Element of Frequency Response Matrix for ω=1

Actual	-2.125151613822383d+00 +6.456438212270109d-02i
via TFM	-2.125151574611664d+00 +6.456438358873129d-02i
Proposed	-2.125151604413986d+00 +6.456437520682812d-02i
Method [17]	-2.125151515007019d+00 +6.456438917666674d-02i

Table 6.4 (1,1) Element of Frequency Response Matrix for ω=10

Actual	-2.097382268402725d-02 +4.376591209620528d-05i
via TFM	-2.097382233478129d-02 +4.376591778054717d-05i
Proposed	-2.097382280044258d-02 +4.376592733024154d-05i
Method [17]	-2.097382198553532d-02 +4.376593096822035d-05i

Table 6.5 (1,1) Element of Frequency Response Matrix for ω=100

Actual	-2.096995310680062d-04 +4.350784624412990d-08i
via TFM	-2.096995276588132d-04 +4.350785731688234d-08i
Proposed	-2.096995312967920d-04 +4.350783444628803d-08i
Method [17]	-2.096995267493185d-04 +4.349920956769893d-08i

Table 6.6 (1,1) Element of Frequency Response Matrix for ω=1000

6.5. CONCLUSIONS

Using a determinant identity, computationally efficient methods for determining the transfer function and frequency response matrices of linear multivariable systems were presented. The properties and performance of the proposed methods were illustrated by means of extremely ill-conditioned practical examples. Although some of the steps of the algorithms do not use orthogonal transformations, the accuracy of the computational results was very good. These algorithms would find applications in both classical frequency response methods for analysis and design of systems as well the new "factoriztion approach". In the latter, the algorithm for computing transfer function matrices would be especially useful because the methods in the factorization approach frequently need to evaluate the transfer function matrices.

6.6. REFERENCES

[1] I. Horowitz, *Synthesis of Feedback Systems*, Academic Press, N.Y., 1963.

[2] H.H. Rosenbrock, *Computer-Aided Control Systems Design*, Academic Press, London, 1974.

[3] I. Postlethwaite and A.G.J. MacFarlane, *Complex Variable Methods for Linear Multivariable Feedback Systems*, Taylor & Francis Ltd., London, 1980.

[4] D.H. Owens , "Multivariable Root-Loci: An Emerging Design Tool", *IEE Int. Conf. on Control*, Warwick, 1981.

[5] R.V. Patel and N. Munro, *Multivariable System Theory and Design*, Pergamon Press, Oxford, 1982.

[6] M. Vidyasagar, *Control Systems Synthesis: A Factorization Approach*, MIT Press, Cambridge, 1985.

[7] A. Varga and V. Sima, "Numerically Stable Algorithm for Transfer Function Matrix Evaluation", *Int. J. Contr.*, Vol 33, pp. 1123-1133, 1981.

[8] A. Emami-Naeini and P. VanDooren, "On Computation of Transmission Zeros and Transfer Functions", *IEEE Contr. and Dec. Conf.*, Las Vegas, pp. 51-55, 1984.

[9] J.H. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford University Press, 1965.

[10] G.W. Stewart, *Introduction to Matrix Computation*, Academic Press, N.Y., 1973.

[11] E.J. Davison and S.H. Wang, "Properties and Calculation of Transmission Zeros of Linear Multivariable Systems", *IEEE Trans. Automat. Contr.*, Vol. 10, pp. 643-648, 1974.

[12] A. Emami-Naeini and P. VanDooren, "Computation of Zeros of Linear Multivariable Systems", *Automatica*, Vol. 18, pp. 415-430, 1982.

[13] P. Misra and R.V. Patel, "Computation of Transfer Function Matrices of Multivariable Systems", to appear in *Automatica*

[14] P. Misra and R.V. Patel, "A Novel Approach for Computing Transfer Function Matrices of Multivariable Systems", *IEEE Int. Symp. Circuits and Syst.*, Philadelphia, pp. 296-299, June 4-7, 1987.

[15] P. Misra and R.V. Patel, "A Determinant Identity with Application in Computing Frequency Response Matrices", to appear in *'Linear Algebra in Signals, Systems and Control'*, B.N. Datta, *Ed.*, SIAM, 1987. *Ed.*

[16] P. Misra and R.V. Patel, "A Computational Method for Frequency Response of Multivariable Systems", *IEEE Contr. and Dec. Conf.*, Ft. Lauderdale, FL, pp. 1248-1249, 1985.

[17] A.J. Laub, "Efficient Multivariable Frequency Response Computations", *IEEE Trans. Automat. Contr.*, Vol. AC-26, pp. 407-408, 1981.

[18] R.V. Patel, "Computational Algorithms for Pole Assignment in Linear Multivariable Systems", *2nd IFAC Symp. on 'Computer-Aided Design of Multivariable Technological Systems'*,, Purdue, IN, pp. 79-89, 1982.

[19] M.A. Hyman, "Eigenvalues and Eigenvectors of General Matrices", *12th National Meeting of A.C.M.*, Houston, TX, 1957.

CHAPTER VII

CONCLUSIONS AND FUTURE WORK

This thesis has been concerned with the important problem of translating elegant theoretical developments in multivariable system theory and design, into computationally reliable algorithms. The main difficulty with many of the existing techniques in the control system literature is their limitation to specific problems or to "low order" systems. Reliability of algorithms becomes an important issue when the algorithms are applied to "high-order" and/or ill-conditioned problems. The main feature of the algorithms presented in this thesis is that they are based on sound principles from numerical linear algebra. As far as possible, orthogonal transformations are used in the computations. Since these transformations are numerically well-conditioned, they are very much suited for computer implementations. The numerical problems of rounding errors are significantly reduced by use of these transformations and therefore, the computed results are, in general, more reliable than those obtained when non-orthogonal transformations are used.

The main contributions of the research described in this thesis have been to provide numerically reliable algorithms for several problems that are commonly encountered in the analysis and design of linear multivariable systems. The results presented in this thesis and related issues that may benefit from further investigation are summarized next:

In Chapter III, numerical algorithms for solving the EVA problem for multi-input systems by means of state feedback was considered. Since the state feedback required to carry out EVA in multi-input systems is not unique, an interesting problem that needs further investigation is whether this degree of freedom can be exploited to improve the numerical performance of the EVA algorithm. Another related aspect is to determine how the EVA task should be distributed among various inputs. The feedback gains can vary by several orders of magnitude, depending on the input from which it is computed. Our experience with many numerical experiments suggests that this is closely related to the controllability properties of the corresponding single-input system. However, a rigorous mathematical treatment is needed to determine an "optimum" distribution of the task of EVA between various inputs.

The results of EVA by state feedback were generalized to EVA by means of constant as well as dynamic output feedback in Chapter IV. The conditioning of the output feedback EVA problem (like te state feedback EVA problem) appears to be related to the structural properties of the given system, such as controllability and observability. A rigorous study of this should prove interesting. The proposed algorithms for EVA by output feedback use rank-2 output feedback matrices. Perhaps the conditioning of the problem can be improved by using higher rank output feedback. Preliminary numerical results seem to reflect this. A systematic study of higher rank output feedback might lead to better computational algorithms.

Based on a new characterization of d.f.m.'s, Chapter V presented a reliable numerical algorithm for determining them. A computational method for EVA in decentralized systems was formalized. Several problems in decentralized control need further investigation. The characterization presented in this chapter assumes an accurate knowledge of open-loop eigenvalues of the system. It would be interesting to see whether fixed modes can be characterized without the knowledge of open-loop eigenvalues. Another interesting problem would be to determine the minimum increase in information exchange between various stations, such that the resulting system has no decentralized fixed modes. In the decentralized EVA problem, an open theoretical as well as computational problem is to determine the minimum order of the dynamic compensator for "almost" arbitrary EVA..

Finally, in Chapter VI, the problems of computation of transfer function and frequency response matrices of linear multivariable systems described by their state-space equations, were considered. A determinant identity was used to compute the above matrices. Some of the transformations used in these algorithms are not orthogonal. However, the numerical performance of the algorithms has been found to be very good. The reason for this would perhaps be revealed by carrying out an error analysis of these.

It is worth mentioning that quite a few of the results presented in this thesis can be extended to multivariable systems in descriptor form:

$$\begin{aligned} \Sigma\dot{x}(t) &= \Phi x(t) + \Gamma u(t) \\ y(t) &= \Theta x(t) + \Xi u(t). \end{aligned}$$

For example, when Σ has full rank, a converse of the generalized eigenvalue problem (QZ algorithm) can be applied for arbitrary EVA by means of state feedback. The case when Σ is not of full rank and corresponding output feedback problem should be interesting problems for further investigation.