POLE-PLACEMENT WITH MINIMUM EFFORT FOR LINEAR MULTIVARIABLE SYSTEMS

by

Naser F. Al-Muthairi

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in
Electrical Engineering

APPROVED:

S. Bingulac, Chairman

D. W. Kuse

R. A. McCoy

H. F. VanLandingham

May, 1988
Blacksburg, Virginia
POLE-PLACEMENT WITH MINIMUM EFFORT FOR LINEAR MULTIVARIABLE SYSTEMS

by
Naser F. Al-Muthairi
S. Bingulac, Chairman
Electrical Engineering

(ABSTRACT)

This dissertation is concerned with the problem of the exact pole-placement by minimum control effort using state and output feedback for linear multivariable systems. The novelty of the design lies in obtaining a direct transformation of the system matrices into a modified controllable canonical form. Two realizations are identified, and the algorithms to obtain them are derived. In both cases, the transformation matrix has some degrees of freedom by tuning a scalar or a set of scalars within the matrix. These degrees of freedom are utilized in the solution to reduce further the norm of the state feedback matrix. Then the pole-placement problem is solved by minimizing a certain functional, subject to a set of specified constraints.

A non-canonical form approach to the problem is also proposed, where it was only necessary to transform the input matrix to a special form. The transformation matrix, in this method, has larger degrees of freedom which can be utilized in the solution. Moreover, a new pole-placement method based on the non-canonical approach is derived. The solution, in this method, was made possible by solving the Lyapunov matrix equation.

Finally, an iterative algorithm for pole-placement by output feedback is extended so as to obtain an output feedback matrix with a small norm. The extension has been
accomplished by applying the successive pole shifting method. Two schemes for the pole shifting are proposed. The first is to successively shift the poles through straight paths starting from the open loop poles and ending at the desired poles, whereas the second scheme shifts the poles according to a successive change of their characteristic polynomial coefficients.
I am grateful to The Creator for everything I have accomplished.

I would like to express my sincere appreciation to Dr. S. Bingulac, my advisor, for his guidance, encouragement and valuable advice throughout the course of this research. Special thanks to Dr. H. F. VanLandingham for gaining from his experience and knowledge. Also the time and effort of my advisory committee members, Dr. D. K. Lindner, Dr. D. W. Luse, and Dr. R. A. McCoy are deeply appreciated.

I am indebted to Kuwait University for providing the financial support which made my graduate education possible.

Finally, I wish to express warmest appreciation to my family, specially my parents, my brother, my sisters, my wife, and my children, for their constant encouragement and many sacrifices throughout my educational program.
### Table of Contents

**INTRODUCTION** ................................................................. 1  
1.1 Motivation and Survey of Previous Work ..................................... 1  
1.2 Problem Formulation ......................................................... 5  
1.3 Dissertation Outline ....................................................... 7  

**OTHER STATE-FEEDBACK CONTROLLERS WITH MINIMUM EFFORT** ........................................ 9  
2.1 Introduction ................................................................. 9  
2.2 Unity-Rank Feedback Method .................................................... 10  
2.3 Ramar's Method ............................................................... 12  
2.4 Successive Pole Shifting Method .............................................. 15  
2.5 Schmidt's Method ............................................................. 19  
2.6 Modal-Control Methods ....................................................... 23  
2.6.1 Control Effort Minimization with Respect to f .................................. 24  
2.6.2 Control Effort Minimization with Respect to f and g .......................... 26  

**SOME CONTROLLABLE CANONICAL FORMS** .............................................. 27  
3.1 Introduction ................................................................. 27
# Table of Contents

3.2 Luenberger's Canonical Forms ........................................ 28
    3.2.1 Non-uniqueness of Luenberger's Canonical Forms .......... 32
    3.2.2 Example 3.1 ........................................... 33

3.3 Modified Canonical Forms ............................................ 37
    3.3.1 Definitions ............................................. 38
    3.3.2 Intermediate Canonical Forms .................................. 41
    3.3.3 Modified Controllable Canonical Forms ....................... 42
    3.3.4 Direct Determination of the Modified Canonical Forms .......... 44

APPROXIMATE METHOD FOR MINIMUM EFFORT POLE-PLACEMENT ............... 45
4.1 Modified Canonical Forms ............................................ 46
    4.1.1 The First Realization ....................................... 47
    4.1.2 The Second Realization ..................................... 51

4.2 Pole-Placement with Minimum Effort .................................. 55
4.3 Non-Canonical Form Approach ........................................ 61
    4.3.1 Pole-placement by Non-canonical Approach .................. 63

4.4 Examples .......................................................... 68
    4.4.1 Example 4.4.1 ........................................... 68
    4.4.2 Example 4.4.2 ........................................... 70
    4.4.3 Example 4.4.3 ........................................... 77
    4.4.4 Example 4.4.4 ........................................... 80
    4.4.5 Example 4.4.5 ........................................... 85

POLE-PLACEMENT BY OUTPUT FEEDBACK ........................................ 88
5.1 Historical Background ................................................ 88
5.2 An Iterative Algorithm .............................................. 96
    5.2.1 Algorithm Development ...................................... 97
    5.2.2 Derivatives of Characteristic Polynomials .................. 99
5.3 Extended Iterative Algorithm ......................................... 101
5.4 Examples ................................................................ 105
  5.4.1 Example 5.1 ................................................... 105
  5.4.2 Example 5.2 ................................................... 108
  5.4.3 Example 5.3 ................................................... 110

Summary and Conclusions .............................................. 113

6.1 Work In Retrospect ................................................ 113
6.2 Directions for Further Research ................................. 116

Proof of Algorithm 4.1 .................................................. 118

Proof of Lemma 4.2.1 ................................................... 121

BASIC Program ....................................................... 122
  C.1 Program Listing ................................................... 122
  C.2 Program Output ................................................... 131

References .................................................................. 137

Vita ........................................................................... 144
List of Illustrations

Figure 1. The control signals of the closed-loop system of example 4.4.1. .... 71
Figure 2. The output signals of the closed-loop system of example 4.4.1. ..... 72
Figure 3. The norm of the matrix T and the norm of the generalized inverse of
        (TB) versus the variable scalar of example 4.4.3. ................ 78
Figure 4. The norm of the matrix T versus the variable scalar of example 4.4.4. 82
Figure 5. The norm of the generalized inverse of (TB) versus the variable scalar
        of example 4.4.4. ............................................ 83
Figure 6. The Listing of the BASIC Program. ............................. 123
Figure 7. A Sample of the Program Output. ............................... 132
Chapter 1

INTRODUCTION

1.1 Motivation and Survey of Previous Work

In most physical systems, there is a constraint on the amplitude of the plant’s input which the system can tolerate. In general, the norm of the feedback gain matrix is a suitable measure for the maximal input amplitude of a regulator utilizing a state variable feedback. This is particularly true if it is assumed that the state variables are always scaled with respect to their maximal values. Another reason for choosing a feedback matrix with small gain is to reduce the effect of the errors introduced by a measurement noise or by quantization errors.

Several methods have been proposed for single-input, single-output (SISO) control systems to obtain a state feedback vector with minimal norm. Given a desired region in the eigenvalue plane to which the eigenvalues of the system should be
shifted, Ackermann [1] gave a state feedback method which works in the $K$ space such that a state feedback vector with minimal norm can be obtained. However, this method is practical only when two eigenvalues of the control system need to be shifted. Kreisselmeir and Steinhauser [2] proposed a rule of thumb to move the system poles within a given region in the complex plane such that large control amplitudes can be avoided. The idea was to minimize the poles' displacements. This minimization was done by compensating for a pole displacement to the left by moving another pole to the right or by decreasing the imaginary parts of a pair of complex poles. Zeiske [3] described a method to decrease the norm of the feedback vector by shifting the system poles within a prescribed area, in such a way that a certain function is minimized. However, the latter method is only applicable for plants modeled in Frobenius state space form. Schmidt [4] extended Zeiske's method to include plants in the general state space description. However, since there is no degree of freedom in the SISO pole-placement problem by state feedback, it was required in the previous methods to shift the poles of the open-loop system to a pre-assigned region in the complex plane, where given design objectives can be met.

On the other hand, for an nth-order multi-input, multi-output (MIMO) linear time-invariant system with m-inputs, where an exact pole assignment is required by a state feedback, there are $n \times (m - 1)$ degrees of freedom available to satisfy additional performance criteria. An interesting problem is to utilize these degrees of freedom in designing a state variable feedback for the exact pole-assignment problem where the norm of the state feedback matrix is required to be minimal.

In the literature, several methods exist for solving this problem. Fallside and Seraji [5] and Seraji [6] used a unity rank controller where there are only $m - 1$ de-
degrees of freedom available. Ramar and Gourishankar [7] found a state feedback matrix with unrestricted rank by minimizing a certain functional with respect to $n \times (m - 1)$ elements of the feedback matrix. Furthermore, it is shown in [7] the advantage of using the unrestricted rank controller over the unity rank controller. However, for systems of order four or higher, since the functional in [7] is not smooth, it is tedious and sometimes impossible to search for the minimum value of the functional with respect to all the variables. Preuß [8;9] developed a successive method for pole shifting. In this approach, the multilinear set of the pole-assignment equations is linearized by doing only small pole shiftings. The whole desired pole placement is achieved by the sum of the many shiftings. Schmidt [10] combined his method in [4] with Preuß’s [8;9] by calculating the positions where the closed-loop poles should be fixed within an allowable area in the complex plane, and then using the pole successive method. Godbout and Jordan [74] and Kouvaritakis and Cameron [75] developed a class of methods which uses the modal control via dyadic pole-placement technique. Their idea was to utilize the modal structure of the system to move some of the system’s modes to desired locations, and at the same time incorporate design constraints on the control magnitude.

On the other hand, there have been some advances in the representation of MIMO systems in the state space. In particular, Bingulac, Padilla and Padilla [11] presented modified canonical forms for the linear multivariable systems. These realizations are modified versions of Luenberger’s canonical forms [12]. Because of its great significance, the latter forms have been further examined by Wolovich [13] and in other references. Unlike the canonical forms presented by Passeri and Herget [14], Sinha and Rozsa [15] and Nelson and Stear [16] which are only applicable to systems with specific structural properties, the realizations in [11] are applicable to any controlla-
ble and observable systems. In addition, they have the advantage of producing a system matrix with a minimum number of nonzero and nonunity parameters, which can be determined from the structural properties of the system (Bingulac [17], Padilla, Padilla and Bingulac [18]). The proposed modified canonical forms use the controllability and observability matrices directly, and, unlike the other existing realization procedures, they do not require any reordering. These realizations have exhibited satisfactory properties and have been used by Bingulac and Farias [19] in the identification and the input-output minimal realization of linear multivariable systems.

The objective of this dissertation is to use the modified canonical forms presented in [11] to solve the exact pole-placement problem for multivariable systems, so that a state feedback matrix with minimal norm can be obtained. To achieve this goal, a direct method was obtained to transform the system matrix to the modified controllable canonical form. The transformation matrix, in the proposed method, has some degrees of freedom by tuning a certain scalar, or a set of scalars, within the matrix. Then, one should choose a desired system matrix which is of the same structure as the transformed system matrix and which also has the desired eigenvalues. This choice can be made possible by minimizing a certain functional with $m \times n$ elements under a set of $n$ constraints.

Another part of the dissertation is to extend an iterative output feedback pole-placement algorithm so that an output feedback matrix with minimum norm can be obtained. The detailed discussion of this topic will be presented in Chapter 5.
1.2 Problem Formulation

Consider the following controllable realization:

\[
\{A, B\}
\]  

(1.2.1)

of the linear, time-invariant, multivariable system modeled by:

\[
q x(t) = A x(t) + B u(t)
\]  

(1.2.2)

where \( q \) denotes a time derivative or a unit time advance for the continuous or the discrete time representation, respectively. The vectors \( x(t) \) and \( u(t) \) are termed the \( n \times 1 \) state vector and the \( m \times 1 \) input vector, respectively. \( A \) and \( B \) are matrices of appropriate dimensions with \( B \) of maximal rank \( m \leq n \). This latter assumption implies that all the inputs are mutually independent which is usually the case in practice.

Let the state feedback law be given by:

\[
u(t) = K x(t)\]

(1.2.3)

where \( K \) is an \( m \times n \) constant matrix, and also a member of a family, \( \kappa \), of \( m \times n \) constant matrices.

The state representation of the closed-loop system becomes:

\[
q x(t) = (A + B K) x(t)
\]

(1.2.4)
If the set of desired closed-loop eigenvalues is given by \( \Lambda^d = \{\lambda_1^d, \ldots, \lambda_n^d\} \), then the state feedback matrix, \( K \), should satisfy the following condition:

\[
\Lambda(A + BK) = \Lambda^d
\]  

(1.2.5)

where \( \Lambda(\cdot) \) denotes the eigenvalues of the argument matrix.

Then, we may define the problem of the multivariable pole-placement by a minimum effort state feedback as follows:

given a controllable pair \( \{A, B\} \) and a set \( \Lambda^d \), find the state feedback matrix \( K' \) which satisfies:

\[
||K'|| = \min_{\forall K \in \kappa} ||K||
\]  

(1.2.6)

where \( \kappa \) is the family of the state feedback matrices which satisfies Eq. (1.2.5) and \( ||\cdot|| \) is the following matrix norm:

\[
||K|| = \left( \sum_{i=1}^{m} \sum_{j=1}^{n} k_{ij}^2 \right)^{1/2}
\]  

(1.2.7)

On the other hand, an output feedback pole-placement algorithm will be discussed in chapter 5. In this case, Eq. (1.2.5) will be replaced by the following relation:

\[
\Lambda(A + BK C) = \Lambda^d
\]  

(1.2.8)

where the matrix \( C \) is the \( p \times n \) output matrix, and \( p \) being the dimension of the output vector.

1 "Eigenvalue" and "pole" are used synonymously.
1.3 **Dissertation Outline**

The structure of this dissertation is as follows: Chapter 2 reviews some of the existing methods which utilize the degrees of freedom in the pole-placement problem for multivariable systems so that a state feedback matrix with minimum norm can be obtained. In particular, five methods reported in the literature are carefully examined, and their drawbacks are outlined. This study indicates that the pole-placement problem with minimum effort does not yet have an optimal solution.

Chapter 3 presents a summary of two types of canonical forms which have great significance in our approach. These two canonical forms are: Luenberger's canonical forms [12] and Bingulac's modified canonical forms [11]. It has been also shown that the selection process of the \( m \) row vectors needed to construct the transformation matrix for Luenberger's canonical form is not unique. For the particular canonical form examined, it has been demonstrated that a certain linear combination of these \( m \) vectors can be used to construct the transformation matrix. A proposition has been given to describe the freedom in the selection process.

Chapter 4 derives modified canonical forms for the linear multivariable systems. The transformation matrix in the proposed method has some degrees of freedom which can be used to reduce further the norm of the state feedback matrix. Then, a solution to the pole-placement problem with minimum effort is outlined. The solution is obtained by minimizing a certain functional subject to a set of specified constraints. Furthermore, a non-canonical approach to the problem is investigated. In the latter approach, it is not necessary to transform the system matrices to a certain canonical
form. However, the only requirement is to transform the input matrix to a special form. Then, a pole-placement method by state feedback is proposed in which a Lyapunov matrix equation should be solved. Numerous examples have been given to demonstrate the various developed algorithms.

Chapter 5 gives a review of an iterative algorithm for pole-placement by output feedback. Then, this algorithm is extended so that a minimum norm feedback matrix can be obtained. The extension is achieved by applying the successive shifting procedure. Two schemes for shifting the poles successively are proposed; and several examples are given.

Finally, Chapter 6 contains a summary of the contributions of this dissertation and conclusions about the various proposed algorithms. Some suggestions for further researchs are also included.
Chapter 2

OTHER STATE-FEEDBACK CONTROLLERS WITH MINIMUM EFFORT

2.1 Introduction

For an $n$'th-order multi-input, multi-output (MIMO) linear time-invariant system with $m$-inputs, where an exact pole assignment is required by a state feedback, there is an infinite number of matrices satisfying this requirement. This non-uniqueness of the state feedback matrix indicates that it is possible to satisfy other design objectives besides positioning the closed-loop poles. Many researchers have tried to utilize the degrees of freedom inherited in the multivariable system to satisfy other objectives such as steady-state characteristics, zero assignment, integrity, etc. The objective of this chapter is to review some of the existing methods for pole-placement
In multivariable systems which utilize these degrees of freedom to obtain state feedback matrix with minimum norm.

### 2.2 Unity-Rank Feedback Method

In this section, the unity rank controller developed by Fallside and Seraji [15] and Seraji [6] is presented. The feedback matrix, in this method, is constrained to have unity rank by defining it in the following dyadic form:

\[ K = qk \]  

(2.2.1)

with \( q \) and \( k \) being \( mx1 \) and \( 1xn \) vectors, respectively.

The feedback matrix in this structure has the advantage that it can be calculated from linear equations. This simplicity is introduced because the open-loop system, with feedback matrix as in Eq. (2.2.1), will be reduced to the following equivalent single-input system:

\[ \dot{x} = Ax + (Bq)u \]

(2.2.2)

In general, \( q \) can be chosen arbitrarily provided that the single-input system in Eq. (2.2.2) is completely controllable. Then, it was shown in [5] that the state feedback vector can be obtained by:

\[ k^T = M^{-1}(a - d) \]

(2.2.3)
where \(a\) and \(d\) are \(n\)-column vectors containing the coefficients of the open-loop and the closed-loop characteristic polynomials, respectively, and \(M\) is an \(n \times n\) constant matrix obtained from the coefficients of the \(n \times 1\) transfer function matrix from the input to the states of the single-input system in Eq. (2.2.2). Since this system is ensured to be completely controllable, the matrix \(M\) is always non-singular. Furthermore, it was shown that this method introduces \((m - 1)\) degrees of freedom in addition to the assignment of the closed-loop poles. This amount of freedom exists because of the arbitrary choice of the vector \(q\).

The general procedure in using \(q\) is as follows:

I. Specify an initial \(q\), and determine the feedback matrix, \(K\), which assigns the closed-loop poles.

II. Set up an error function \(E(q)\) between a certain performance aspect of the closed-loop system and its desired value.

III. Increment \(q\) to minimize \(E(q)\), and then update \(q\).

IV. Repeat the whole procedure until the minimization of \(E(q)\) is obtained.

The proposed procedure has been used in [6] to choose a certain \(q\) so that the magnitudes of some of the feedback matrix elements do not exceed specified limits. In addition, the procedure can be used to obtain a state feedback matrix with minimal norm by defining the error function to be in the following form:
Although the unity-rank method has considerable simplicity, the resulting closed-loop systems have poor disturbance rejection properties (Daniel [42]). Therefore, in the following sections, a review will be given for several methods which produce state feedback matrices with full rank.

2.3 Ramar's Method

In this section, the method of unrestricted rank controllers proposed by Ramar and Gourishankar [7] is reviewed. The state feedback matrix, in this approach, is of unrestricted rank; therefore, there are \( n \times (m - 1) \) degrees of freedom available to satisfy additional design objectives besides placing the closed-loop poles. Since \( n \times (m - 1) \) is greater than \( m - 1 \), additional performance criteria can be satisfied more closely by using a full-rank feedback controller than with a unity rank controller. The feedback control in Eq. (1.2.3) can be written as:

\[
\bar{u} = \bar{K} x
\]  

(2.3.1)

\[
\hat{u} = \hat{k} x
\]  

(2.3.2)

where

\[ E(q) = \sum_{i=1}^{m} \sum_{j=1}^{n} k_{ij}^{2}(q) \]  

(2.2.4)
\[ \bar{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{q-1} \\ u_{q+1} \\ \vdots \\ u_m \end{bmatrix} \quad ; \quad \hat{u} = u_q \] (2.3.3)

and

\[ \bar{K} = \begin{bmatrix} k_1 \\ k_2 \\ \vdots \\ k_{q-1} \\ k_{q+1} \\ \vdots \\ k_m \end{bmatrix} \quad ; \quad \hat{k} = k_q \] (2.3.4)

\(u_i\)'s are the elements of \(u\) and \(k_i\)'s are the rows of \(K\).

Applying the control in Eq. (2.3.1) to the open-loop system, we obtain the following equivalent single-input system:

\[ \dot{x} = \bar{A}(\bar{K}) x + \bar{b} \hat{u} \] (2.3.5)
where

\[
\bar{A}(K) = A + B K ; \quad \bar{B} = [b_1, b_2, \ldots, b_{q-1}, b_{q+1}, \ldots, b_m] ; \quad \hat{b} = b_q
\]

\(b_i\)'s are the columns of \(B\).

The single controller \(\hat{u} = K x\) is designed to achieve the desired closed-loop poles, while the \((m - 1) \times n\) matrix \(\bar{K}\) is utilized to minimize the norm of the overall feedback matrix \(K\). An expression for the vector \(\hat{k}\) which assigns the desired closed-loop poles is given by:

\[
\hat{k} = - \{P(\bar{K})\}_n \bar{A}(\bar{K}) - d P(\bar{K}) \tag{2.3.6}
\]

where \(P(\bar{K})\) is the matrix which transforms the pair \((\bar{A}(\bar{K}), \hat{b})\) to its phase-variable (companion) form \((P\bar{A}P^{-1}, Pb)\), \(d\) is the row vector containing the coefficients of the desired characteristic polynomial, i.e., \(d = [d_n, d_{n-1}, \ldots, d_1]\), and \(\{P\}_n\) is the \(n^{th}\) row of \(P\). An algorithm for computing the transformation matrix \(P\) can be found in many of the control literatures.

Now, using Eq. (2.3.6), the matrix \(K\) for pole assignment can be written in terms of \(\bar{K}\) only. Then, a functional similar to the one in Eq. (2.2.4) can be minimized iteratively with respect to \(\bar{K}\). Once \(\bar{K}\) is obtained for minimum feedback norm, \(\hat{k}\) can be calculated using Eq. (2.3.6).

Several examples have been given in [7] which show clearly that the results obtained by using full-rank controllers are better than those obtained by using unity-rank controllers. However, the proposed method has several drawbacks. One of them is
that the amount of the symbolic computations involved in the method makes it dif- 
cult to apply for higher order systems. Another drawback is that in applying the de- 
sign procedure to systems of order four or higher especially if there is a large number 
of inputs, it is tedious and sometimes impossible to search for the minimal value of 
the functional with respect to all the variables. This difficulty occurs because the 
functional in this method is not smooth. Also, it is noted that a different choice of the 
row vector \( \hat{k} \) would result in a different norm of \( K \). However, it is not obvious which 
one of the \( m \) row vectors should be chosen so that the resultant feedback matrix has 
the minimal norm.

2.4 Successive Pole Shifting Method

The successive pole shifting method, as proposed by Preuß [8;9], solves the 
pole-assignment problem approximately and, at the same time, utilizes the degrees 
of freedom to achieve a minimum norm solution. The idea behind this method is to 
linearize the multilinear set of the pole-assignment equations by doing only small 
pole shiftings; then the whole desired pole placement is achieved by the sum of the 
many shiftings.

Consider the closed-loop characteristic polynomial given by:

\[
d(s) = \det(sI - A - BK) \\
= s^n + d_{n-1}s^{n-1} + \cdots + d_0
\]  

(2.4.1)
Therefore, each of the closed-loop polynomial coefficients is a nonlinear function of the elements of the feedback matrix $K$. In fact, this nonlinear function is of special type called multilinear (i.e., linear with respect to the elements of each column of $K$). Furthermore, it can be shown that each of these coefficients can be expressed as:

$$d_i = a_i + L_i(K) + N_i(K) \quad i = 0, 1, \ldots, n - 1 \quad (2.4.2)$$

where $a_i$ is the $i$'th coefficient of the open-loop characteristic polynomial, $L_i(K)$ and $N_i(K)$ are a linear and a multilinear function of the elements of $K$, respectively.

If only small pole shiftings are required, which means the following assumption holds:

$$|k_{ij}| < 1 \quad i = 1, \ldots, m \ ; j = 1, \ldots, n \quad (2.4.3)$$

then, the multilinear part, $N_i(K)$, of Eq. (2.4.2) can be neglected compared with the linear part, $L_i(K)$, and Eq. (2.4.2) becomes:

$$d_i = a_i + L_i(K) \quad i = 0, 1, \ldots, n - 1 \quad (2.4.4)$$

Furthermore, if the elements of the matrix $K$ are arranged as the following vector:

$$k = [k_{11}, k_{12}, \ldots, k_{mn}]^T \quad (2.4.5)$$

then, the multilinear relations in Eq. (2.4.4) can be rewritten in the following compact form:

$$M k = (d - a) \quad (2.4.6)$$
where the $n$-dimensional vector $(d - a)$ is defined as follows:

$$(d - a) = [d_0 - a_0, \ldots, d_{n-1} - a_{n-1}]^T$$

and the $n \times mn$ matrix, $M$, contains the coefficients of the linear equations in (2.4.4), and it depends only on the plant's parameters. A method for calculating $M$ will be given later.

Finally, using the Moore-Penrose pseudoinverse $M^+$ of $M$, the solution of Eq. (2.4.6) will be:

$$k = M^+ (d - a) \quad (2.4.7)$$

It should be pointed out that the pseudoinverse solution is the logical choice because it minimizes simultaneously the error norm $\|Mk - (d - a)\|$ and the norm $\|k\| = \sqrt{k^T k}$.

In practice, one should apply the method repeatedly, starting again and again from the new closed-loop poles. Then, if the partial feedback matrix obtained at each step is $K_i$, the total feedback matrix will be:

$$K = \sum_{i=1}^{L} K_i \quad (2.4.8)$$

where $L$ is the total number of successive pole displacements. Furthermore, experience with this method has shown that the total feedback matrix has usually a low norm.
The matrix $M$ can be calculated using the following procedure. We may rewrite Eq. (2.4.6) as follows:

$$\begin{bmatrix} k_{11} \\ k_{12} \\ \vdots \\ k_{mn} \end{bmatrix} = \begin{bmatrix} d_0 - a_0 \\ d_1 - a_1 \\ \vdots \\ d_{n-1} - a_{n-1} \end{bmatrix}$$ (2.4.9)

where $m_i$ is the $i$'th column of $M$, and the integer $y = mn$. Then, if we choose the vector $k$ to have 1 at the $p$'th position and zero elsewhere, the left-hand side of Eq. (2.4.9) will be the column $m_p$. Therefore, the $p$'th column of $M$ can be determined by subtracting the coefficients of the open-loop characteristic polynomial from the coefficients of the closed-loop polynomial as obtained from Eq. (2.4.1). However, in this case, the feedback matrix $K$ used in Eq. (2.4.1) is a result of rearranging the vector, which contains 1 at the $p$'th position, into a matrix. Then, if we repeat the previous procedure by positioning the one in another location in the vector $k$, we can obtain another column of $M$. Finally, repeating the procedure again, all columns of $M$ can be obtained.

In conclusion, the successive pole shifting method solves the pole-assignment problem approximately. It shifts the system poles, lying at undesired locations, step by step to the left until they have reached a more favorable region in the complex plane; however, it is not clear whether the method can place the closed-loop poles at exact desired locations. Furthermore, the size of the displacement within a single step can only be found by using an interactive computer program.
2.5 Schmidt's Method

In this section, Schmidt's method [10] is outlined. In previous results, Zeiske [3] and Schmidt [4] developed a method for single-input single-output (SISO) systems to calculate the locations of some of the closed-loop poles to be fixed within a pre-scribed area so that the resultant feedback vector would have small gains. Then, the method has been extended in [10] to MIMO systems by combining it with the successive pole shifting method. The development of the new method can be clarified as follows.

Let's rename the elements of the vector $k$ in Eq. (2.4.5) by the following:

$$k = [k_1 \ k_2 \ ... \ k_l]^T$$

where $l = nm$.

Then, if we write the pseudoinverse of $M$ in the form:

$$M^+ = \begin{bmatrix} m_{11} & ... & m_{1n} \\ \vdots & \ddots & \vdots \\ m_{l1} & ... & m_{ln} \end{bmatrix}$$

the solution shown in Eq. (2.4.7) can be written as:

$$\begin{bmatrix} k_1 \\ \vdots \\ k_l \end{bmatrix} = \begin{bmatrix} m_{11}(d_0 - a_0) + ... + m_{1n}(d_{n-1} - a_{n-1}) \\ \vdots \\ m_{l1}(d_0 - a_0) + ... + m_{ln}(d_{n-1} - a_{n-1}) \end{bmatrix}$$
Hence, each element of the vector $k$ can be constructed as the following sum:

$$k_r = \sum_{v=1}^{n} m_{rv} (d_{v-1} - a_{v-1}) ; \quad r = 1, \ldots, l \quad (2.5.4)$$

It is clear from Eq. (2.5.4) that the elements $k_r$ can only be reduced further by a suitable choice of the $d_i$'s.

A functional similar to Eq. (1.2.7) may be chosen. However, neglecting the square root operation and introducing a factor of $1/2$, then substituting the values of $k_r$'s from Eq. (2.5.4), we obtain the following functional:

$$J = \frac{1}{2} \sum_{r=1}^{l} \left[ \sum_{v=1}^{n} m_{rv} (d_{v-1} - a_{v-1}) \right]^2 \quad (2.5.5)$$

The functional in Eq. (2.5.5) should be minimized with respect to the coefficients $d_i$'s. In other words, some of the closed-loop poles lying in the desired prescribed area should be shifted within this area so as to minimize the functional in Eq. (2.5.5), whereas, the poles lying outside the desired area should be shifted successively in its direction. Therefore, the closed-loop polynomial should be partitioned into two polynomials in the following form:

$$d(s) = z(s) q(s) = \sum_{j=0}^{p} z_j s^j \sum_{\zeta=0}^{n-p} q_\zeta s^\zeta \quad (2.5.6)$$

The roots of $q(s)$ should be chosen as the desired closed-loop poles which must be shifted toward the desired area. The polynomial $z(s)$ must be chosen so as to mini-
mize the functional in Eq. (2.5.5). Since the polynomial \( d(s) \) is normalized, we have \( d_n = z_p = q_{n-p} = 1 \). The remaining coefficients can be found using the result in [3]:

\[
d_i = \sum_{j=0}^{p} z_j q_{i-j} ; \quad q_i = 0 \quad \text{for} \quad i < 0 \quad \text{and} \quad i > n - p
\]  

(2.5.7)

Finally, substituting Eqs. (2.5.6) and (2.5.7) in Eq. (2.5.5), the desired functional becomes:

\[
J = \frac{1}{2} \sum_{r=1}^{l} \left[ \sum_{v=1}^{n} m_{rv} \left( \sum_{j=0}^{p} z_j q_{v-j-1} - a_{v-1} \right) \right]^2
\]  

(2.5.8)

All the terms and coefficients in Eq. (2.5.8) are fixed except \( z_j \)'s. Therefore, the minimization of the functional in (2.5.8) would give the coefficients \( z_j \)'s; then, the desired closed-loop polynomial coefficients \( d_j \)'s can be found using Eq. (2.5.6). Finally, the desired feedback matrix can be obtained using Eq. (2.4.7).

However, it should be noted that all the previous calculations should be repeated in every step of the successive pole shifting. Hence, the total feedback matrix is the sum of all the partial feedback matrices as per Eq. (2.4.8).

Solving the minimization problem in Eq. (2.5.8) yields the following set of linear equations:

\[
Qz = f
\]  

(2.5.9)

where \( z = [z_0 \ldots z_{p-1}]' \), and \( Q \) takes the following form:
\[
Q = \begin{bmatrix}
q_{00} & q_{01} & \cdots & q_{0,p-1} \\
q_{01} & q_{11} & & \\
\vdots & & \ddots & \\
q_{p-1,0} & \cdots & & q_{p-1,p-1}
\end{bmatrix}
\]

(2.5.10)

The elements of \( Q \) can be determined by:

\[
q_{mn} = \sum_{r=1}^{l} \left[ \sum_{v=0}^{n-p} m_{r,(v+m+1)} q_v \right]^2 ; \quad m = 0, \ldots, p - 1
\]

(2.5.11)

\[
q_{mj} = \sum_{r=1}^{l} \left[ \left( \sum_{v=0}^{n-p} m_{r,(v+1)} q_v \right) \left( \sum_{v=0}^{n-p} m_{r,(v+m+1)} q_v \right) \right] ; \quad m = 0, \ldots, p - 1
\]

\[
; \quad j = 0, \ldots, p - 1 ; \quad m \neq j
\]

(2.5.12)

\[
q_{jm} = q_{mj} ; \quad m = 0, \ldots, p - 1 ; \quad j = 0, \ldots, p - 1
\]

(2.5.13)

and the elements of the vector \( f \) are:

\[
f_m = \sum_{r=1}^{l} \left[ \left( \sum_{v=1}^{n} m_{r,2v-1} - \sum_{v=0}^{n-p-1} m_{r,(v+p+1)} q_v \right) \left( \sum_{v=0}^{n-p} m_{r,(v+m+1)} q_v \right) \right] ; \quad m = 0, \ldots, p - 1
\]

(2.5.14)

In summary, Schmidt's method requires that some of the closed-loop poles should be shifted arbitrarily within the allowable area. Hence, exact pole assignments are not possible. Furthermore, it may happen that the matrix \( Q \) is singular, or the poles of the polynomial \( z(s) \) are outside the allowable area. Also, the amount of computations required by this method is clearly excessive.
2.6 Modal-Control Methods

The methods in this class use the modal control via dyadic pole-placement technique developed by Simon and Mitter [76] and further refined by Retallack and Macfarlane [77]. The principle of modal control utilizes the modal structure of the system to move some of the system's modes to desired locations and, at the same time, incorporates design constraints on the control magnitude. In this section, the method of Kouvaritakis and Cameron [75] is reviewed as an example of the class.

Suppose that the unity-rank controller takes the form:

\[ K = fg^T \]  

(2.6.1)

where \( f \) and \( g \) are \( m \) and \( n \)-dimensional vectors, respectively. Here, two problems are considered. First, a functional similar to Eq. (2.2.4) is minimized over \( f \) by making the preserved modes unobservable. Next, an approach is considered which distributes the degrees of freedom between \( f \) and \( g \) by also making some of the preserved poles uncontrollable. However, the method moves one real pole or a complex conjugate pair of poles at one time. Therefore, an iterative algorithm has been presented to place many poles to some desired locations. For simplicity, the case of relocating real poles only is considered.
2.6.1 Control Effort Minimization with Respect to $f$

Let Jordan canonical form for $A$ be given by:

$$\Lambda = V^{-1}AV$$

where $V$ is a modal matrix for $A$.

Then, the controller required to shift $p$ poles is given in [77] to be:

$$K = f d^i V_i^i$$

where $V_i$ is the matrix made up of the first $p$ rows of $V$, and $d^i$ is a $p$-dimensional vector with elements given by:

$$d_i = \frac{a_i}{\beta_i}$$

where

$$a_i = \frac{\prod_{j=1}^{n}(\lambda_j - \lambda_i^q)}{\prod_{j=1}^{n}(\lambda_i - \lambda_j)}$$

$$a_1 = \lambda_1 - \lambda_1^q$$

$$\beta_i = v_i^i B f$$
and $v_i$ is the $i$'th row of $V_i$ corresponding to the open-loop pole $\lambda_i$. The vector $f$ is chosen freely with the only restriction that $\beta_i \neq 0$.

However, if it is required to move one real pole from $\lambda$, to $\lambda_i$, then the vector $f$ which minimizes the cost functional is given by:

$$ f = \frac{f}{h^T f} $$

where

$$ h^T = \frac{v_i^T B}{a_1} $$

Furthermore, to place $p$ poles, we may use an iterative algorithm which gives the controller at the $i$'th stage by:

$$ K_i^{(i)} = K_{i-1}^{(i-1)} + f^{(i)} \sigma^{(i)} v_i^{(i)} \quad i = 0, 1, 2, ... $$

where $K_{i-1}^{(i-1)}$ is defined to be a zero matrix.

However, in order to optimize the norm of the resulting controller at each stage with respect to the controller structure obtained at all previous stages, we should use for $f_i$ the following expression:

$$ f_i^{(i)} = \rho \left[ B^T v_i^{(i)} - \frac{v_i^{(i)} B B^T v_i^{(i)}}{a_i^{(i)} v_i^{(i)} + v_j^{(i)} B K_{i-1}^{(i-1)} v_j^{(i)}} \right] $$

where $\rho$ is any nonzero scalar, and $v_j^{(i)}$ is the eigenvector corresponding to the pole $\lambda_j$ which must be shifted to $\lambda_j^{(i+n)}$. 

OTHER STATE-FEEDBACK CONTROLLERS WITH MINIMUM EFFORT 25
2.6.2 Control Effort Minimization with Respect to $f$ and $g$

Here, an extension to the algorithm in [77] is presented. The proposed extension increases the degrees of freedom by allowing modes to be preserved by making them either uncontrollable or unobservable.

Minimizing the cost functional with respect to both $f$ and $g$ and, considering again the problem of relocating one real pole, we get:

$$f = M[M'M]^{-1}M'B\nu_1$$  (2.6.12)  

$$g' = I_1\nu_1^t[I - V_2[V_2'V_2]^{-1}V_2']$$  (2.6.13)  

where $V_2$ is the submatrix of $V$ when it is partitioned as $V = [V_1 \mid V_2 \mid V_3]$ and $V_1$ corresponds to the modes being made uncontrollable. The matrix $M$ is a full-rank representation of the kernel of $V_1B$, and $I_1 = \frac{a_1}{\nu_1 Bf}$.

Once again, to place more than one pole, an iterative algorithm must be used.

In conclusion, the modal control techniques have been implemented to utilize the degree of freedom. However, the method has several drawbacks. First, it is assumed that the poles of the open-loop system are distinct. Second, the optimization which was carried out in the iterative approach does not give a minimized overall solution because the minimization in each stage is carried out with respect to each dyadic controller component. Third, more calculations are needed to obtain the updated eigenframes as functions of the eigenframes of the previous stage. Finally, the number of computations needed is large.
Chapter 3

SOME CONTROLLABLE CANONICAL FORMS

3.1 Introduction

Canonical forms are useful as a starting point for deriving certain other general results for multivariable systems. Furthermore, they are beneficial for initiating design considerations for several problems in MIMO systems. For example, they have been used in identification, minimal input-output realization, pole-placement, observer design, and many other problems. Therefore, canonical forms received considerable attention and generated an extensive amount of literature. However, one can find several surveys devoted to this field. A survey on canonical forms for multivariable identification is found in Denham's paper [78]. A survey of Luenberger's forms is found in the work of Sinha and Rozsa [15]. A survey on several canonical forms for time-invariant linear SISO and MIMO systems is reported by Maroulas and Barnett [79] and [80], respectively.
The purpose of this chapter is to review two types of canonical forms which have great significance in our approach for solving the proposed problem. These two canonical forms are Luenberger’s canonical forms developed in [12] and Bingulac’s modified canonical forms discussed in [11].

In the sequel, we shall assume that the system is completely controllable and must have an input matrix $B$ with full column rank. The latter assumption is a technical condition included because dependent columns of $B$ contain redundant information, which does not need to be included in the system model.

### 3.2 Luenberger’s Canonical Forms

Luenberger [12] presented two structures of canonical forms for linear multivariable systems. Then, many researchers extended these structures to obtain several forms where each one of them proved to be useful for a particular need. A survey of these forms is reported by Sinha and Rozsa [15]. In this section, a review of the second type of Luenberger’s canonical forms is given. This form is particularly useful in pole-placement problem.

Consider any completely controllable system with state-space representation given by:

$$\dot{x}(t) = A x(t) + B u(t)$$

(3.2.1)

where the input matrix $B$ has full rank $m \leq n$. 

SOME CONTROLLABLE CANONICAL FORMS
An appropriate choice of the transformation matrix is required. This transformation will make a change of coordinates from state vector $x$ to $z$ defined by $z = Tx$. Therefore, the open-loop system (3.2.1) will be transformed to the following form:

$$\dot{z} = \tilde{A} z + \tilde{B} u$$  \hspace{1cm} (3.2.2)

where

$$\tilde{A} = TAT^{-1} \quad ; \quad \tilde{B} = TB$$  \hspace{1cm} (3.2.3)

Now, before presenting the procedure to construct $T$, the following useful definition will be introduced.

**Definition 3.2.1:**

The controllability index $n_i$ is the smallest integer such that $A^n b_i$ is linearly dependent on its predecessors in the matrix $[B | AB | A^2B | ...]$.

where $b_i$ is the $i$'th column of the input matrix $B$.

Now, construct the nonsingular $n \times n$ matrix $Q$ defined by:

$$Q = [b_1 | b_2 | ... | b_m | Ab_1 | ... | Ab_m | ... | A^{n_1-1}b_1 | ... | A^{n_m-1}b_m]$$  \hspace{1cm} (3.2.4)

Then, write $Q^{-1}$ in terms of its row vectors:
The only rows which play a direct role in the canonical forms are the row vectors $e_{i_1}, i = 1, \ldots, m$. For simplicity of notation, label these row vectors by:

$$e_i = e_{i_1}; \quad i = 1, \ldots, m$$

Finally, the transformation matrix $T$ can be constructed by:

$$T = \begin{bmatrix}
    e_1 \\
    e_1 A \\
    e_1 A^2 \\
    \vdots \\
    e_1 A^{n-1} \\
    e_2 \\
    \vdots \\
    e_m A^{n_m-1}
\end{bmatrix}$$

It has been shown in [12] that the matrix $T$ is nonsingular. Also, it is straight forward to show that the transformation matrix in Eq. (3.2.7) would transform the system matrices to the following forms:
\[ \tilde{A} = \begin{bmatrix}
0 & 1 & 0 & \ldots & 0 \\
0 & 0 & 1 & \ldots & 0 \\
\vdots & & & & \\
1 & & & & \\
x & x & x & \ldots & x \\
0 & 1 & 0 & \ldots & 0 \\
0 & 0 & 1 & \ldots & 0 \\
\vdots & & & & \\
1 & & & & \\
x & & & & \\
0 & 1 & \ldots & 0 \\
\vdots & & & & \\
1 & & & & \\
x & & & & \\
\cdots & \ldots & \ldots & \ldots & \ldots \\
\end{bmatrix} \]

(3.2.8)

\[ \tilde{B} = \begin{bmatrix}
0 & 0 & \ldots & 0 \\
0 & 0 & \ldots & 0 \\
\vdots & & & \\
1 & x & \ldots & x \\
0 & 0 & \ldots & 0 \\
0 & 0 & \ldots & 0 \\
\vdots & & & \\
0 & 1 & \ldots & x \\
\vdots & & & \\
0 & 0 & \ldots & 1 \\
\end{bmatrix} \]

(3.2.9)
where the $x$'s represent possible nonzero elements.

It should be pointed out that the matrix $\tilde{A}$ may be considered as composed of companion matrices located in blocks along the diagonal. Each one of these companion matrices has a dimension of $n_x \times n_x$.

The difficulties associated with the calculation of the matrices $Q$ and $T$ have been reduced by the development of various algorithms (Jordan and Sridhar [81], Datta [82], Aplevich [83] and Daly [84]).

### 3.2.1 Non-uniqueness of Luenberger's Canonical Forms

As it is well known, the canonical forms for MIMO systems are generally not unique. Furthermore, the structure of the canonical form can be controlled to a certain extent by the designer. Luenberger's canonical forms also share some of these properties. In particular, the selection of the vectors comprising the matrix $Q$ is somewhat arbitrary. In other words, the selection process is not unique, and there is a certain amount of freedom in this process. However, for a particular form, different values of the non-zero elements will be encountered, whereas the structure of the matrices $\tilde{A}$ and $\tilde{B}$ will be preserved.

In this section, a new kind of freedom in choosing the $m$ rows used in constructing the matrix $T$ in Eq. (3.2.7) is presented. Specifically, a linear combination of the rows $e_i$'s in Eq. (3.2.6) can be used to construct the matrix $T$. The following proposition
describes the above freedom for the particular canonical form discussed in the previous section.

**Proposition 3.1**

Consider the row vectors $e_i$'s shown in Eq. (3.2.6). Each one of these row vectors can be replaced by a linear combination of the other rows as depicted by:

$$e'_i = e_i + \sum_{j=1, j\neq i}^{m} a_{ij} e_j \quad ; \quad i = 1, \ldots, m$$  \hspace{1cm} (3.2.10)

where the coefficient $a_{ij}$ takes nonzero value when only the controllability index associated with the input $i$ is less than the controllability index associated with the input $j$ (i.e., $n_i < n_j$). Then, if the row vectors $e'_i$'s are used to construct $T$ as per Eq. (3.2.7), the same canonical form discussed in the previous section will be obtained.

The following example illustrates the previous discussion.

**3.2.2 Example 3.1**

Consider the following system matrices:
Simple calculation shows that the controllability indices are:

\[ n_1 = 3 \ , \quad n_2 = 1 \ , \quad n_3 = 2 \]

Therefore, Eqs. (3.2.4)-(3.2.9) give the following results:

\[
Q = [b_1 \ b_2 \ b_3 \ Ab_1 \ Ab_3 \ A^2b_1] =  \begin{bmatrix}
1 & -1 & 2 & 4 & 5 & 9 \\
-1 & 1 & 0 & 0 & 2 & 1 \\
2 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 1 \\
1 & 0 & 0 & 1 & -1 & 3 \\
0 & 0 & 0 & 1 & 1 & 0
\end{bmatrix}
\]  (3.2.12)

\[
Q^{-1} =  \begin{bmatrix}
\theta_{11} \\
\theta_{21} \\
\theta_{31} \\
\theta_{12} \\
\theta_{32} \\
\theta_{13}
\end{bmatrix} =  \begin{bmatrix}
.034 & .034 & .655 & -.069 & -.310 & -.483 \\
-.414 & .586 & .138 & .828 & .724 & .793 \\
-.276 & -.276 & -.241 & 1.552 & .483 & .862 \\
-.172 & -.172 & -.276 & .345 & .552 & 1.414 \\
.172 & .172 & .276 & -.345 & -.552 & -.414 \\
.103 & .103 & -.034 & -.207 & .069 & -.448
\end{bmatrix}
\]  (3.2.13)

\[
e_1 = e_{13} \ , \quad e_2 = e_{21} \ , \quad e_3 = e_{32}
\]  (3.2.14)
Rewriting Eq. (3.2.10), we obtain the following relation:

\[
R = \begin{bmatrix}
1 & \alpha_{12} & \alpha_{13} \\
\alpha_{21} & 1 & \alpha_{23} \\
\alpha_{31} & \alpha_{32} & 1
\end{bmatrix}
\] (3.2.18)

According to proposition 3.1, the only nonzero coefficients in the matrix \(R\) are: 
\(\alpha_{21}, \alpha_{23}\) and \(\alpha_{31}\).

Choosing for the matrix \(R\) the following value:
the matrices $T$, $\tilde{A}$ and $\tilde{B}$ will take the following forms:

$$T = \begin{bmatrix}
.103 & .103 & -.034 & -.207 & .069 & -.448 \\
.207 & .207 & .172 & -.414 & -.345 & .345 \\
.414 & .414 & .586 & .897 & -.172 & -1.276 \\
-.121 & .879 & .207 & .241 & .586 & -.310 \\
.379 & .379 & .207 & -.759 & -.414 & -1.310 \\
.759 & .759 & .966 & -.517 & -1.931 & -.069
\end{bmatrix}$$

$\tilde{A}$ and $\tilde{B}$ preserved the same structure as in Eq. (3.2.16).

However, if proposition 3.1 is violated and the coefficient $\alpha_{18}$ takes a nonzero value, the matrix $R$ will be:

$$R = \begin{bmatrix}
1 & 0 & 0 \\
2 & 1 & .5 \\
2 & 0 & 1
\end{bmatrix}$$

the matrices $T$, $\tilde{A}$ and $\tilde{B}$ will be:
\[ T = \begin{bmatrix}
0.621 & 0.621 & 0.793 & -1.241 & -1.586 & -1.690 \\
1.241 & 1.241 & 2.034 & 0.517 & -4.069 & -1.931 \\
-0.121 & 0.879 & 0.207 & 0.241 & 0.586 & -0.310 \\
0.379 & 0.379 & 0.207 & -0.759 & -0.414 & -1.310 \\
0.759 & 0.759 & 0.966 & -0.517 & -1.931 & -0.069
\end{bmatrix} \] (3.2.23)

\[ \tilde{A} = \begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
-1.813 & -2.065 & 2.759 & 0 & -1.316 & 3.038 \\
1.076 & -0.200 & 0 & 1 & -3.624 & 1.100 \\
0 & 0 & 0 & 0 & 0 & 1 \\
5.414 & -5.759 & 2 & 0 & -1.586 & 1.241
\end{bmatrix} ; \quad \tilde{B} = \begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 3 \\
1 & 0 & 11.345 \\
0 & 1 & 0 \\
0 & 0 & 0 \\
0 & 0 & 1
\end{bmatrix} \] (3.2.24)

where the matrix \( \tilde{B} \) is no longer in the Luenberger's canonical form which is in accordance with proposition 3.1.

### 3.3 Modified Canonical Forms

Bingulac, Padilla and Padilla [11] presented modified canonical forms for linear multivariable systems. These realizations are modified versions of Luenberger's canonical forms [12]. Unlike the canonical forms in [14 - 16], which are only applicable to systems with specific structural properties, the realizations in [11] are applicable to any controllable and observable systems. In addition, they have the advantage of producing a system matrix with minimum number of parameters, which can be
determined from the structural properties of the system [17;18]. Also, the proposed forms use the controllability and observability matrices directly and do not require any reordering. As an application, these realizations have been used in the identification and the input-output minimal realization of linear multivariable systems [19].

Before proceeding further, several useful definitions will be presented.

3.3.1 Definitions

The Reduced Controllability (Observability) Matrix:

Let \( Q_c(A,B) \) and \( Q_o(A,C) \) be the controllability and observability matrices of the triple \( \{A,B,C\} \), respectively, with:

\[
\rho[Q_c(A,B)] = \rho_c \quad \text{and} \quad \rho[Q_o(A,C)] = \rho_o
\]

where \( \rho[\cdot] \) denotes the rank of the argument matrix.

Then, the reduced controllability (observability) matrix is the \( n \times \rho_c \ (\rho_o \times n) \) matrix \( Q_c(A,B) \) \( (Q_o(A,C)) \) which contains the left most \( \rho_c \) (top most \( \rho_o \)) linearly independent columns (rows) of \( Q_c(A,B) \) \( (Q_o(A,C)) \), respectively.
The Horizontal (Vertical) Canonical Form:

The horizontal canonical form is the $n \times n$ matrix, associated with an $l$-dimensional integer vector $p = (p_1, p_2, \ldots, p_l)^T$, defined by:

$$
\begin{bmatrix}
0 & \ldots & 0 & 1 \\
& & & 1 \\
& & & 1 \\
p_1 \rightarrow & x & \ldots & x & x & \ldots & x & x \\
p_i \rightarrow & x & \ldots & x & x & \ldots & x & x \\
& & & & & & & 1 \\
& & & & & & & 1 \\
& & & & & & & 1 \\
& & & & & & & 1 \\
p_l \rightarrow & x & \ldots & x & x & \ldots & x & x \\
\end{bmatrix}
$$

(3.3.1)

The matrix $A_h^p$ has the following structural properties:

1. There are only $l$ rows which may have nonzero and nonunity elements.

2. The indices of these rows are given by the integer vector $p$ with

$$p_i < p_{i+1}$$
3. In any one of the rows indexed by \( p_i, i = 1, \ldots, l \), only the first \( q_i \) elements may have values other than zero, where \( q_i \) is given by:

\[
q_i = p_i - i + 1
\]

In a dual sense, the vertical canonical form \( A_c \) may be defined.

One can easily notice that the matrices \( A_g \) and \( A_c \) are uniquely defined by the integer vector \( p \).

The matrix \( D^c \)

The \( n \times m \) matrix \( D^c \) associated with the integer vector \( p \) has the following form:

\[
D^c = \begin{bmatrix}
0 \\
p_1 \rightarrow \\
\vdots \\
p_l \rightarrow \\
0
\end{bmatrix}
\]

\[
\begin{bmatrix}
x & x & x & x & x \\
x & x & x & x & x \\
\vdots \\
0 \\
x & x & x & x & x
\end{bmatrix}
\]

Therefore, the matrix \( D^c \) has the following structural properties:
1. There are only $l$ rows with nonzero elements. The indices of these rows are: $p_i, i = 1, \ldots, l$.

2. Each row $d_{p_i}, i = 1, \ldots, l$ has at least one element with unity value, i.e.,

$$d_{p_i} = \begin{bmatrix} 0 & \ldots & 0 & 1 & x & \ldots & x \end{bmatrix}$$

The column where the unity occurs will be indexed $s_i, 1 \leq s_i \leq m$.

3. The entries $d_{p_i j} = 0$ for $j = 1, \ldots, s_{i-1}$

4. The indices $s_i, i = 1, \ldots, l$ are all distinct.

Using these definitions, intermediate canonical forms and modified canonical forms will be formulated next. However, since the pole-placement problem is the main concern of this dissertation, the controllable form only of each realization is presented.

### 3.3.2 Intermediate Canonical Forms

The intermediate controllable canonical form of the system represented by Eq. (3.2.1) can be obtained by the following state transformations:

$$R_1 = \{A_1, B_1\} = \{T_1^{-1}AT_1, T_1^{-1}B\}$$

(3.3.3)

where

$$T_1 = Q_c(A, B)_r$$

(3.3.4)
Furthermore, it can be shown that the structure of the realization $R$, can be described by:

$$A_1 = A^c_v, \quad B_1 = \begin{bmatrix} f_m \\ 0 \end{bmatrix}$$  \hspace{1cm} (3.3.5)

where $p^c$ is an integer vector obtained from the $m$-dimensional integer vector $n^c$ by the following rule:

$$p^c = f_{\text{alg}}(n^c)$$  \hspace{1cm} (3.3.6)

$n^c$ being the $m$-dimensional integer vector whose elements are the controllability indices of the open-loop system, and $f_{\text{alg}}$ being the transformation defined by the algorithm shown in [11].

### 3.3.3 Modified Controllable Canonical Forms

The modified controllable canonical form of the system represented in Eq. (3.2.1) can be obtained from the intermediate canonical form by the following state transformation:

$$R_2 = \{A_2, B_2\} = \{T_2A_1T_2^{-1}, T_2B_1\}$$  \hspace{1cm} (3.3.7)

where

$$T_2 = Q_o(A_1, C_1)_r$$  \hspace{1cm} (3.3.8)
The $m \times n$ auxiliary output matrix $C$, contains as its rows, the rows of the identity matrix $I_n$ which correspond to the indices $p_i$; $i = 1, \ldots, m$.

It should be pointed out that it can be proven that the pair $(A_i, C_i)$ is observable provided that the pair $(A, B)$ are controllable.

Furthermore, the structure of the realization $R_z$ can be described by:

$$A_2 = A^c_2; \quad B_2 = D^c_2$$  \hspace{1cm} (3.3.9)

The integer vector $\vec{p}^c$, which defines the structural properties of $R_z$, may be determined using the algorithm in [11]:

$$\vec{p}^c = f_{alg}(\vec{n}^c)$$  \hspace{1cm} (3.3.10)

where $\vec{n}^c$ is the $m$-dimensional integer vector whose elements are the controllability indices of $(A_z, B_z)$ which correspond to those of system (3.2.1), but are increasingly ordered, i.e.,

$$\vec{n}^c = \begin{bmatrix} n_1^c & \cdots & n_m^c \end{bmatrix}^T; \quad 1 \leq n_1^c \leq \cdots \leq n_m^c \leq n - m + 1$$  \hspace{1cm} (3.3.11)

However, a direct method exists which transforms the original system into the modified controllable canonical form without the need of determining the intermediate canonical form.
3.3.4 Direct Determination of the Modified Canonical Forms

The modified controllable canonical form of the system (3.2.1) can be obtained directly from the original realization by the following state transformation:

\[ R_c = \{A_c, B_c\} = \{TAT^{-1}, TB\} \quad (3.3.12) \]

where

\[ T = Q_o(A, H)_r \quad (3.3.13) \]

The \( m \times n \) equivalent output matrix \( H \) contains those rows from the inverse of \( Q_o(A, B) \), which correspond to the indices \( p^c = [p_1 \ldots p_m]^T \).

In the next chapter, a modified version of these forms will be used to solve the exact pole-placement problem for multivariable systems, so that a state-feedback matrix with minimal norm can be obtained. To achieve this objective, a direct transformation is obtained to transform the system to these modified controllable canonical forms. Furthermore, the transformation matrix in the proposed method has some degrees of freedom which can be used to decrease further the norm of the feedback matrix.
In this chapter, an approximate solution to the minimum effort pole-placement problem in multivariable systems is presented. This method produces a state-feedback matrix with a norm which is very close to the absolute minimum. However, the simplicity and the reduced amount of computation in the method make it attractive. Moreover, unlike some of the other methods, the proposed algorithm is applicable to higher order systems.

This chapter progresses as follows. In section 1, modified canonical forms are presented. The proposed canonical forms have some degrees of freedom which can be used to reduce further the norm of the state feedback matrix. Then, in section 2, a solution to the pole-placement problem is outlined. The solution has been obtained by minimizing a certain functional subject to a set of constraints. The form of the
functional reflects our objective for the desired matrix to be as close as possible to
the transformed system matrix so that a small feedback norm could be obtained,
whereas the constraint ensures that the desired matrix has the set of required
eigenvalues (i.e. \(\Lambda^d\)). In section 4, a non-canonical approach to the problem is in-
vestigated. In this approach, it was shown that to solve the pole-placement problem,
it is not necessary to transform the system matrices to a certain canonical form.
However, the only requirement is that the transformed input system matrix should
have zero rows in the first \(n - m\) rows. The transformation matrix, in this case, has
larger degrees of freedom which can be utilized to solve the formulated problem.
Moreover, a pole-placement method is proposed based on the non-canonical ap-
proach. In this method, a Lyapunov matrix equation is solved to obtain the desired
state feedback matrix. In section 4, numerous examples are given to demonstrate the
various algorithms.

4.1 Modified Canonical Forms

Again, let’s consider the linear, time-invariant, multivariable system modeled by:

\[
\dot{x}(t) = Ax(t) + Bu(t)
\]  

The first step in the solution is to transform the system matrices given in Eq. (4.1.1)
to a modified controllable canonical form. The proposed form is a modified version
of the Luenberger’s canonical form [12; 13]. The structure of the transformed system
matrices in the modified form is different than Luenberger’s canonical form as we
shall see later. Another difference is that the transformation matrix in the proposed
method has some degrees of freedom while the transformation matrix in the original form does not have any freedom. This degree of freedom will be used later to reduce further the norm of the state-feedback matrix. If \( T \) denotes the similarity transformation matrix in this method, then the transformed system matrices will be given by:

\[
\tilde{A} = TAT^{-1} ; \quad \tilde{B} = TB
\]  

(4.1.2)

However, the structure of the matrices \( \tilde{A} \) and \( \tilde{B} \) and the algorithm to find the matrix \( T \) will depend on the structural properties of the system represented by Eq. (4.1.1). The structural properties of a system can be characterized by its controllability (structural) indices which are defined in section 3.2.

In fact, there exist two realizations depending on the values of these indices. These two cases will be discussed below.

4.1.1 The First Realization

The realization in this case is the generic one, and the canonical forms here can be obtained when the first \( n \) columns of the controllability matrix are linearly independent. The condition for the first \( n \) columns of the controllability matrix to be linearly independent can be easily seen to hold if and only if the controllability indices \( n_i \) satisfy the following relation [18]:

\[
n_i = \begin{cases} 
    h + 1 & \text{for } i \in [1, m_i] \\
    h & \text{for } i \in [m_i + 1, m] 
\end{cases}
\]  

(4.1.3)
The symbol \[
\binom{a}{b}
\]
denotes the greatest integer value less than or equal to the ratio \(a/b\), i.e.,
\[
(a/b) - 1 < \binom{a}{b} \leq a/b
\]
and the integer \(m_i\) is given by:

\[
m_i = n - m \binom{n - 1}{m} \quad ; \quad 1 \leq m_i \leq m
\]

The structure of the matrices \(\tilde{A}\) and \(\tilde{B}\), in this case, can be described by:

\[
\tilde{A} = \begin{bmatrix}
O & I_{n-m} \\
A_2
\end{bmatrix} ; \quad \tilde{B} = \begin{bmatrix}
O \\
B_2
\end{bmatrix}
\]

where the matrices \(A_2\) and \(B_2\) are \(m \times n\) and \(m \times m\) arbitrary matrices respectively, \(I_{n-m}\) is the identity matrix of dimension \(n - m\) and \(O\) is the null matrix of dimension \((n - m) \times m\).

Inspecting the realizations given in [17], one can conclude that the realization (4.1.5) corresponds to the case where the minimal number of parameters needed to represent the system is largest.
The transformation matrix, in this case, can be constructed using the following algorithm.

Algorithm 4.1

Let \( t', \ldots, t^n \) be the rows of the transformation matrix \( T \). These rows can be found as follows:

\[
\begin{align*}
\begin{bmatrix}
B & AB & \cdots & A^{p-1}B
\end{bmatrix} &= 0 \\
& \vdots \\
\begin{bmatrix}
B & AB & \cdots & A^{p-1}B
\end{bmatrix} &= 0 \quad (4.1.6)
\end{align*}
\]

\[
\begin{align*}
t^{m+i} &= t^{i}A & i_1 &= 1, 2, \ldots, \min\{m, n - m\} \\
t^{2m+i} &= t^{m+i}A & i_2 &= 1, 2, \ldots, \min\{m, n - 2m\} \\
& \vdots \\
t^{(l-1)m+i} &= t^{(l-1)m+i}A & i_l &= 1, 2, \ldots, \min\{m, n - lm\} \\
\end{align*}
\]

where

\[
p = \binom{n}{m}, \quad k = n - mp, \quad 0 \leq k \leq m - 1 \quad \text{and} \quad l = \binom{n-1}{m} \quad (4.1.8)
\]

APPROXIMATE METHOD FOR MINIMUM EFFORT POLE-PLACEMENT
The proof of algorithm 4.1.1 can be shown by a straightforward algebraic manipulation, and one should also observe the structure of the matrices $\tilde{A}$ and $\tilde{B}$, and should use the relation (4.1.2). The proof of this algorithm is shown in Appendix A.

The way that algorithm 4.1 can be used is as follows:

I. Calculate the integers $p,k$ and $l$ using Eq. (4.1.8).

II. Find the row vectors $t'$ to $t^n$ by calculating the null space of the appropriate matrices given in Eq. (4.1.6). For example, the row vectors $t'$ to $t^k$ are in the null space of the matrix $[B \mid AB \mid \cdots \mid A^{p-1}B]$, whereas the row vectors $t^{k+1}$ to $t^n$ are in the null space of the matrix $[B \mid AB \mid \cdots \mid A^{p-2}B]$.

III. Calculate the row vectors $t^{n+1}$ to $t^n$ using Eq. (4.1.7).

Remark 4.1.1

For the special case when $n - m < m$, algorithm 4.1 will give a negative value for $p - 2$ in Eq. (4.1.6). In this case, that means the row vectors $t^{k+1}$ to $t^n$ are free to be chosen with the only restriction that they are linearly independent with the other vectors in $T$.

Since the row vectors $t'$ to $t^n$ lie in a null space of certain matrices, the transformation matrix $T$ has a certain degree of freedom by tuning a scalar, or a set of scalars, within these rows. For example, any one of the row vectors $t'$ to $t^n$ can be multiplied by a scalar and still the resultant transformation matrix will produce the desired canonical form. In this case, the new row vector will be:
where $\alpha_i$ is a tuned scalar.

On the other hand, all the row vectors $t_i$ to $t_m$ can be multiplied by a matrix of scalars and the new row vectors will be:

$$
\begin{bmatrix}
\tilde{t}_1 \\
\vdots \\
\tilde{t}_m
\end{bmatrix} =
\begin{bmatrix}
\alpha_{11} & \ldots & \alpha_{1m} \\
\vdots & \ddots & \vdots \\
\alpha_{m1} & \ldots & \alpha_{mm}
\end{bmatrix}
\begin{bmatrix}
t_1 \\
\vdots \\
t_m
\end{bmatrix}
$$

The former type of tuning will be used later to get the right transformation matrix which helps to reduce further the norm of the feedback matrix.

### 4.1.2 The Second Realization

The controllability indices $n_i$ of the system in Eq. (4.1.1), in this case, do not satisfy Eq. (4.1.3). In other words, the first $n$ columns of the controllability matrix are not linearly independent. However, the number of rows in $\tilde{A}$ or $\tilde{B}$, which contain nonzero and nonunity elements, is the same as in the first realization. These rows are in certain locations depending on the values of the controllability indices. Before we proceed further, we need to define a set of integers called the Kronecker indices.
Definition 4.1.1

The Kronecker indices of the system in Eq. (4.1.1) are a set of integers \( \{ \mu_1, \ldots, \mu_m \} \) which are equal to the increasingly ordered controllability indices of the system, i.e.,

\[
1 \leq \mu_1 \leq \cdots \leq \mu_m \leq n - m + 1
\]

In this realization, the structure of the matrices \( \tilde{A} \) and \( \tilde{B} \) can be described by:

\[
\begin{bmatrix}
\vdots \\
0 & \cdots & 0 & 1 \\
& & & 1 \\
& & & & 1 \\
\end{bmatrix}
\begin{bmatrix}
r_1 \\
\vdots \\
r_m \\
\end{bmatrix}
\quad \begin{bmatrix}
0 \\
0 \\
0 \\
\end{bmatrix}
\begin{bmatrix}
x & \cdots & x & x & \cdots & x \  \\
x & \cdots & x & x & \cdots & x \  \\
x & \cdots & x & x & \cdots & x \  \\
\end{bmatrix}
\begin{bmatrix}
x & x & x & x \  \\
x & x & x & x \  \\
x & x & x & x \  \\
\end{bmatrix}
\]

The basic properties defining the structure of the matrix \( \tilde{A} \) may be described as follows:

1. There are only \( m \) rows which may have nonzero and nonunity elements.

2. The locations of these rows may be defined by the following integers:
\[ s_i = s_{i-1} + (\mu_i - \mu_{i-1})(m - i + 1) + 1 \quad i = 1, \ldots, m \]  
\[ \text{with } s_0 = 0 \text{ and } \mu_0 = 1 \]

3. The remaining \( n - m \) rows are the last \( n - m \) rows of the unity matrix \( I_n \). The location of the unity in the \((s_i - 1)\)'th row is determined by the following relation:

\[ r_i = s_i - i + m \quad i = 1, \ldots, m \]

The unity in the \((s_i + 1)\)'th row is located in the \((r_i + 1)\)'th column. Furthermore, the unities in the rows between the \((s_i + 1)\)'th and the \((s_{i+1} - 1)\)'th row are arranged diagonally.

The transformation matrix, in this case, can be constructed by the following algorithm.

**Algorithm 4.2**

Let \( t', \ldots, t^n \) be the rows of the transformation matrix \( T \). Then, these rows can be found using the following relations:

\[ t^1 [B | AB | \cdots | A^{q_1 - 1}B] = 0 \quad \text{where } q_1 = \begin{pmatrix} s_1 - 1 \\ m \end{pmatrix} \]

\[ t^2 [B | AB | \cdots | A^{q_2 - 1}B] = 0 \quad \text{where } q_2 = \begin{pmatrix} s_2 - s_1 - 1 \\ m - 1 \end{pmatrix} + q_1 \]  
\[ \vdots \]

\[ t^m[B | AB | \cdots | A^{q_m - 1}B] = 0 \quad \text{where } q_m = \begin{pmatrix} s_m - s_{m-1} - 1 \\ 1 \end{pmatrix} + q_{m-1} \]
Algorithm 4.2 can be used as follows:

I. Find the Kronecker indices of the system (4.1.1).

II. Compute the integers $s_i$'s using Eq. (4.1.12).

III. Use Eq. (4.1.14) to find the integers $q_i$'s; then calculate the row vectors $t^i$ to $t^m$. For instance, the row vector $t^i$ ($i = 1, \ldots, m$) is in the null space of the matrix $[B \mid AB \mid \cdots \mid A^{q_i-1}B]$.

IV. Find the row vectors $t^{m+1}$ to $t^m$ using Eq. (4.1.15).

Remark 4.1.2

In algorithm 4.2, if $q_i = 0$, then $t^i$ may be chosen freely but with the only restriction that it is linearly independent of the other rows of the matrix $T$. Observing Eq. (4.1.12) and Eq. (4.1.14), conditions for $q_i = 0$ can be easily identified. For example, when the Kronecker index $\mu_i = 1$ that leads to $q_i = 0$. In addition, if $\mu_2 = 1$ then $q_2 = 0$. However,
this case can be considered as an additional degree of freedom in choosing the matrix $T$.

**Remark 4.1.3**

In constructing the rows $t^m$ to $t^n$ using Eq. (4.1.15), if any $i_j$ ($j = 1, \ldots, m$) does not take a set of ascending integers, then skip to the next equation. This means the $s_j$'th and the $s_{j+1}$'th row are following each other without any other rows between them, i.e., $s_{j+1} = s_j + 1$.

In the following section, these canonical forms are used to solve the minimum effort pole-placement problem.

### 4.2 Pole-Placement with Minimum Effort

The approach to solve the proposed problem starts with the transformation of the open-loop realization in Eq. (4.1.1) to the modified controllable canonical forms described in the previous section. Then, consider the well known pole-placement condition given by:

$$\tilde{A} + \tilde{B}K = A^d$$  \hspace{1cm} (4.2.1)

where the $n \times n$ matrix $A^d$ is to be chosen so that it has the same structure as the matrix $\tilde{A}$, and its eigenvalues are the desired ones, i.e., $\Lambda^d$. The reason for $A^d$ to have...
the same structure as $\tilde{A}$ is related to the solvability of Eq. (4.2.1) and will be clarified later. Thus, the characteristic polynomial of the matrix $A^d$ should satisfy the following relation:

$$w(s) = \prod_{i=1}^{n}(s - \lambda_i^d) = \sum_{k=0}^{n} c_k s^{n-k} , c_0 = 1 \quad (4.2.2)$$

where $w(s)$ is the characteristic polynomial of $A^d$, and $c_k$'s are the coefficients of the desired characteristic polynomial. Solving Eq. (4.2.1), the state feedback matrix in its transformed form, $\tilde{K}$, may be obtained using one of the following equivalent expressions:

$$\tilde{K} = \tilde{B}^+ (A^d - \tilde{A}) = (\tilde{B}^T \tilde{B})^{-1} \tilde{B}^T (A^d - \tilde{A}) = (\tilde{B})_r^{-1} (A^d - \tilde{A})_r \quad (4.2.3)$$

where "$T$" denotes matrix transposition, while the matrix $\tilde{B}^+$ is the $m \times n$ generalized (pseudo) inverse of the $n \times m$ matrix $\tilde{B}$. The matrices $(\tilde{B})$, and $(A^d - \tilde{A})$, are the $m \times m$ and $m \times n$ reduced matrices containing only the nonzero rows of the respective matrices. The generalized (pseudo) inverse can be calculated using Bingulac's algorithm [20].

Observing Eq. (4.2.3), one may conclude that if the matrix $(A^d - \tilde{A})$ has minimum norm, the state feedback matrix will have small norm, accordingly. In other words, to get a state feedback matrix with small norm, the elements of the matrix $A^d$ may be obtained by minimizing the following functional:

$$J = \sum_{i=1}^{n} \sum_{j=1}^{m} (a_{ij}^d - \tilde{a}_{ij})^2 \quad (4.2.4)$$
subject to the constraints:

$$f_k(a_{ij}^d) = c_k \quad k = 1, \ldots, n \quad i = s_1, \ldots, s_m \quad j = 1, \ldots, n$$

(4.2.5)

where $\tilde{a}_i$ is the $(i,j)$-th element of the matrix $\tilde{A}$, $a_{ij}^d$ is the $(i,j)$-th element of the matrix $A^d$, $c_k$ is the coefficient given in Eq. (4.2.2), and $f_k(\cdot)$ is a function of the unknown elements of the matrix $A^d$ depicted by the following relation:

$$\det(sl - A^d) = \sum_{k=0}^{n} f_k s^{n-k} \quad f_0 = 1$$

(4.2.6)

It should be pointed out that the constraints in Eq. (4.2.5) are imposed so that to force the eigenvalues of the matrix $A^d$ to be the desired ones.

Now, we can understand why we require that the matrix $A^d$ must have the same structure as the matrix $\tilde{A}$. Since the matrix $\tilde{B}$ has zero columns corresponding to the zero rows of the matrix $\tilde{A}$, the matrix $(A^d - \tilde{A})$ should have zero rows at the same locations. Otherwise, the matrix $\tilde{K}$ obtained from Eq. (4.2.3) will not be a solution to Eq. (4.2.1).

One last point to be considered is concerning the degrees of freedom inherited in the algorithm of constructing the transformation matrix. These degrees of freedom can be utilized in the solution which would result in a transformation matrix that helps to reduce the norm of the feedback matrix. The way to do that is as follows. Since the feedback matrix is given by:

$$K = \tilde{B}(A^d - \tilde{A})T$$

(4.2.7)
It is obvious that by tuning a scalar or a set of scalars within the matrix $T$, the value of the matrix $K$ would be affected. Furthermore, experimental works with this phenomenon indicate that minimal norm of the matrix $K$ can be achieved when either the norm of $T$ or the norm of $\tilde{B}^+$ levels off or when the free scalar takes its nominal value (i.e., 1).

Strictly speaking, our method does not necessarily give the absolute minimum because the feedback matrix, as shown in Eq. (4.2.7), is a result of multiplying three matrices; however, we are only minimizing one of them while compromising the other two. Nevertheless, this method gives a result which is very close to the minimum. Also, the proposed computations are much easier than the computations in the other methods. In addition, the method is applicable to higher order systems while the methods which give absolute minimum (i.e., Ramar's method [7]) are not.

In summary, the proposed minimum effort pole-placement algorithm can be outlined by the following steps.

**Algorithm 4.2.1**

**Step 1:** Plot the norm of the matrix $T$ and the norm of the generalized inverse of the matrix $(TB)$ versus the free scalar in $T$. Then choose a particular value of the scalar when either of the above norms levels off. Denote the transformation matrix at this value by $T^*$.

**Step 2:** Find $\tilde{A}$ and $\tilde{B}$ using $T^*$ and the relation (4.1.2).
Step 3: Choose the desired matrix $A'$ by minimizing the functional in Eq. (4.2.4) subject to the constraints in Eq. (4.2.5).

Step 4: Finally

$$K^* = B^+ (A^d - \tilde{A})^T$$

Remark 4.2.1

Since explicit expressions are given for the functional in Eq. (4.2.4) subject to the constraints in Eq. (4.2.5), any standard minimization method may be employed to find $A'$. However, the number of variables can be reduced by specifying that some of the variables must equal to the value of the corresponding elements in $\tilde{A}$ so that convex constraints may be obtained. In this case, an optimal solution to the minimization problem may be achieved. Nevertheless, a good result was obtained without the need to fix some of the variables.

The optimization problem in the worked examples is solved using GINO optimization package. This package is chosen because it is friendly and interactive.

Remark 4.2.2

The functions $f_\ell (-)$ can be easily calculated using a digital computer. An explicit equation to calculate these functions is given in [21]:

**APPROXIMATE METHOD FOR MINIMUM EFFORT POLE-PLACEMENT**
\[ f_k = \sum_{i_1 < \cdots < i_k} (-1)^k \Delta(i_1, i_2, \ldots, i_k) \] (4.2.8)

where \( \Delta(i_1, \ldots, i_k) \) is the \( k \times k \) determinant formed from rows \( i = i_1, \ldots, i_k \) and columns \( j = i_1, \ldots, i_k \). However, since \( A^d \) has a special structure with rows contain only unities and zeros, some of these determinants are zeros. The only nonzero determinant can be found using the following lemma:

**Lemma 4.2.1**

Let \( S = \{s_1, \ldots, s_m\} \) and \( I = \{i_1, \ldots, i_k\} \). The determinant \( \Delta(i_1, \ldots, i_k) \) is nonzero only if:

\[ i_j + m - l \in I \] (4.2.9)

for every \( i_j \in \{i_j \mid i_j \in I \text{ but } i_j \notin S \text{ for } j = 1, \ldots, k \} \) where \( l = \text{the number of elements in } G \subset S \) and \( G = \{g_i \mid g_i \in S \text{ and } g_i < i_j\} \).

The proof of this lemma is shown in Appendix B.

A computer program is written in BASIC for IBM-PC to calculate the functions \( f_d(\cdot) \) using lemma 4.2.1. The listing of this program is shown in Appendix C.

In the next section, a non-canonical approach to the problem is outlined.
4.3 Non-Canonical Form Approach

Unlike the method in the previous section, it is not necessary for \( \tilde{A} \) and \( \tilde{B} \) to have certain canonical forms. In fact, the only requirement is that the matrix \( \tilde{B} \) must have zero elements in the first \( n - m \) rows, while the matrix \( \tilde{A} \) can take any general form, i.e.:

\[
\tilde{A} = \begin{bmatrix}
\text{general} \\
n \times n \\
\text{matrix}
\end{bmatrix} ; \quad \tilde{B} = \begin{bmatrix}
0 & \ldots & 0 \\
\vdots & \ddots & \vdots \\
0 & \cdots & 0 \\
x & \cdots & x \\
x & \cdots & x
\end{bmatrix}
\] (4.3.1)

The transformation matrix, in this method, has considerable degrees of freedom. Specifically, the first \( n - m \) rows of the transformation matrix can be obtained from the null space of the matrix \( B \):

\[
\begin{bmatrix}
t^1 \\
\vdots \\
t^{n-m}
\end{bmatrix} = [B] [O]
\] (4.3.2)

whereas, the last \( m \) rows of the matrix \( T \) can be chosen arbitrarily provided they are linearly independent of the other \( n - m \) rows. A suitable choice of the last \( m \) rows are the rows of the matrix \( B^T \). These considerable degrees of freedom can be employed to reduce further the norm of the feedback matrix.
Now, the desired matrix $A'$ must be chosen such that its first $n - m$ rows are equal to the first $n - m$ rows of $\bar{A}$ while its last $m$ rows are chosen to minimize the functional in Eq. (4.2.4) subject to the constraints given in Eq. (4.2.5). Then, the desired feedback matrix which places the closed-loop poles at the desired locations can be found using Eq. (4.2.7).

As mentioned earlier, the degrees of freedom within the transformation matrix can be utilized to reduce further the norm of the feedback matrix. For example, several scalars can be tuned within $T$ to get a new transformation matrix $T'$:

$$T' = \begin{bmatrix}
\alpha_{11} & \cdots & \alpha_{1,(n-m)} & | & \beta_{11} & \cdots & \beta_{1m} \\
\vdots & | & \vdots & & \vdots & & \vdots \\
\alpha_{(n-m),1} & \cdots & \alpha_{(n-m),(n-m)} & | & \beta_{m1} & \cdots & \beta_{mm}
\end{bmatrix}
\begin{bmatrix}
t^1 \\
\vdots \\
t^{n-m} \\
\vdots \\
t^{n-m+1}
\end{bmatrix}$$

where $\alpha_i$'s and $\beta_i$'s are free scalars and $t$'s are the rows of the matrix $T$. Experience with this method shows that by tuning these free scalars, one can obtain a feedback matrix with small norm. It was not possible to find a systematic procedure to tune all these free scalars so that a feedback matrix with minimum norm can be obtained; however, choosing values for these scalars in arbitrary manner has proved to give good results. The main difficulty associated with this method is the cost of calculating all the determinants in Eq. (4.2.8) because lemma 4.2.1 cannot be applied in this case.
The non-canonical form approach can be used to place the system poles at desired locations. The development of this new algorithm is shown in the next subsection.

### 4.3.1 Pole-placement by Non-canonical Approach

This pole-placement method starts by transforming the system matrices to the following forms:

\[
A_c = TA \, T^{-1} \\
B_c = TB
\]  

(4.3.4)

where the matrices \(A_c\) and \(B_c\) have the following structure:

\[
A_c = \begin{bmatrix} A_{c1} \\ A_{c2} \end{bmatrix} ; \quad B_c = \begin{bmatrix} 0 \\ B_{c2} \end{bmatrix}
\]  

(4.3.5)

the sub-matrix \(A_{c1}\) has a dimension of \((n - m) \times n\), while \(A_{c2}\) and \(B_{c2}\) are \(m \times n\) and \(m \times m\), respectively.

The problem is to select a desired matrix which has the desired eigenvalues and the following structure:

\[
A_{dc} = \begin{bmatrix} A_{c1} \\ A_{dc2} \end{bmatrix}
\]  

(4.3.6)

where \(A_c\) and \(A_{dc}\) have the same first \(n - m\) rows. Then, using the following pole-placement condition, one can determine the feedback matrix:
\[ A_c - B_c K_c = A_{dc} \] (4.3.7)

However, because of the special structure of \( A_c, B_c \) and \( A_{dc} \), the state feedback matrix which places the system poles at the desired locations will be:

\[ K = B_{c2}^{-1} (A_{c2} - A_{dc2}) T \] (4.3.8)

Therefore, we need to determine the matrix \( T \) to calculate \( B_{c2} \) and \( A_{c2} \). Also, we need to choose a value for \( A_{dc2} \). These calculations will be shown next.

**Determination of \( T \)**

The matrix \( T \) can be determined using the singular value decomposition. In this method, the matrix, \( B^T \) can be decomposed into the following matrices:

\[ B^T = U \Sigma V^T \] (4.3.9)

where \( U \) and \( V \) are \( m \times n \) and \( n \times n \) unitary matrices, respectively; and \( \Sigma \) contains the singular values of \( B^T \) and has the following structure:

\[ \Sigma = \begin{bmatrix} \Sigma' & 0 \\ 0 & 0 \end{bmatrix} \] (4.3.10)

where \( \Sigma' \) is a diagonal matrix containing the singular values of \( B^T \). The matrix \( V \) can be partitioned as \( V = [V_1 \mid V_2] \) where \( V_1 \) and \( V_2 \) are \( n \times m \) and \( n \times (n - m) \) sub-matrices, respectively. Then, from Eqs. (4.3.9) and (4.3.10), one can conclude that \( V_2 \) is in the null space of \( B \). Therefore, a suitable choice for \( T \) is:
\[ T = \begin{bmatrix} V_2^T \\ V_1^T \end{bmatrix} \]  

(4.3.11)

However, \( V_1^T \) could be replaced by any \( m \times n \) arbitrary matrix provided that \( T \) is non-singular.

Using this transformation matrix, one can determine \( A_{c1} \), \( A_{ct} \) and \( B_{ct} \) using Eq. (4.3.4). The only quantity in Eq. (4.3.8) which remains undetermined is \( A_{ct2} \).

**Determination of \( A_{ct2} \)**

Usually, we start with a matrix \( A_v \) which is in a diagonal or upper (lower) triangular form having the desired eigenvalues. The objective is to find a transformation matrix \( T_d \) which can transform \( A_v \) into the form shown in Eq. (4.3.6):

\[ T_d A_v T_d^{-1} = \begin{bmatrix} A_{c1} \\ A_{dc2} \end{bmatrix} \]  

(4.3.12)

partitioning the matrices \( T_d \) and \( A_v \) into:

\[ T_d = \begin{bmatrix} T_{d1} \\ T_{d2} \end{bmatrix} \quad ; \quad A_{c1} = [A_{c11} \mid A_{c12}] \]  

(4.3.13)

where \( T_{d1} \), \( T_{d2} \), \( A_{c11} \), and \( A_{c12} \) are \((n - m) \times n\), \(m \times n\), \((n - m) \times (n - m)\) and \((n - m) \times m\) sub-matrices, respectively.
Using Eq. (4.3.13) and rewriting the upper part of Eq. (4.3.12), one obtains the following relation:

\[ T_{d1} \ A_d = [A_{c11} \ | A_{c12}] \begin{bmatrix} T_{d1} \\ T_{d2} \end{bmatrix} \] (4.3.14)

Eq. (4.3.14) can be rearranged in the following form:

\[ T_{d1} \ A_d - A_{c11} \ T_{d1} = A_{c12} \ T_{d2} \] (4.3.15)

Eq. (4.3.15) can be identified as the Lyapunov matrix equation. Therefore, choosing an arbitrary value for \( T_{d1} \), the Lyapunov equation can be solved for \( T_{d1} \). Then, using the obtained value for \( T_{d1} \), one can determine \( A_{c22} \) from Eq. (4.3.12). Finally, the state feedback matrix which places the poles at the desired locations can be obtained using Eq. (4.3.8).

In summary, the new pole-placement method can be outlined by the following steps:

**Algorithm 4.3.1: Pole-placement by State Feedback**

**Step 1:** Select a value for \( A_d \).

**Step 2:** Determine \( T \) using the singular value decomposition in Eqs. (4.3.9)-(4.3.11).

**Step 3:** Find \( B_{c2} \), \( A_{c1} \) and \( A_{c2} \) using Eqs. (4.3.4)-(4.3.5).

**Step 4:** Partition \( A_{c1} \) into \( A_{c11} \) and \( A_{c12} \) as in Eq. (4.3.13).
Step 5: Choose an arbitrary value for $T_e$.

Step 6: Solve the Lyapunov matrix equation for $T_e$.

Step 7: Find $A_{dc}$ using the obtained value for $T_e$ and Eq. (4.3.12).

Step 8: Determine $K$ using Eq. (4.3.8).

It should be noted that if the obtained $T_e$ is singular, one must reject the result and choose another arbitrary $T_e$.

Alternative Method

To avoid partitioning the matrix $V$ into $V_1$ and $V_2$ and rearrange them in the matrix $T$, one can define the transformed matrices to be in the following structure:

$$A_c = \begin{bmatrix} A_{c1} \\ A_{c2} \end{bmatrix} ; \quad B_c = \begin{bmatrix} B_{c1} \\ 0 \end{bmatrix}$$

$$A_{dc} = \begin{bmatrix} A_{dc1} \\ A_{dc2} \end{bmatrix}$$

In this case, the transformation matrix $T$ will be:

$$T = V^T$$

and the Lyapunov matrix equation will be:

$$T_{d2} A_d - A_{c22} T_{d2} = A_{c21} T_{d1}$$
which can be solved for $T_\alpha$ for any given arbitrary value of $T_\alpha$. Then, $A_{dct}$ can be determined using the obtained $T_\alpha$.

Finally, the required feedback matrix will be:

$$K = B_{ct}^{-1} (A_{ct} - A_{dct}) T$$  \hspace{1cm} (4.3.20)

In the next section, several examples will be given to demonstrate the advantage of the proposed method and to compare it with the other approaches.

### 4.4 Examples

#### 4.4.1 Example 4.4.1

Consider the sixth-order model of a gas absorber system given in [8] and [10]. The system matrices are given by:

$$A = \begin{bmatrix} e & f & 0 \\ d & e & f \\ d & e & f \\ d & e & f \\ O & d & e \\ d & e & f \end{bmatrix} \hspace{1cm} B = \begin{bmatrix} d & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & f \end{bmatrix}$$  \hspace{1cm} (4.4.1)

where $d = 0.539$, $e = -1.17$ and $f = 0.634$

The output matrix and the initial state vector are given by:
The poles of the open-loop system are given by:

\[ \Lambda = \{-0.117, -0.441, -0.91, -1.43, -1.9, -2.223\} \]

The set of the desired closed-loop eigenvalues is as follows:

\[ \Lambda^d = \{-0.5, -0.5, -0.91, -1.43, -1.9, -2.223\} \]

It is easy to see that the first 6 columns of the controllability matrix are linearly independent. Hence, algorithm 4.1 was employed to find the transformation matrix which is given by:

\[
T = \begin{bmatrix}
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0.539 & -1.170 & 0.634 & 0 \\
0 & 0 & 0.539 & -1.170 & 0.634 & 0 \\
0.291 & -1.261 & 2.052 & -1.484 & 0.402 & 0 \\
0 & 0.291 & -1.261 & 2.052 & -1.484 & 0.402 \\
\end{bmatrix}
\] (4.4.4)

Using this transformation matrix and without the need to tune the matrix, the transformed system matrices are given by:
whereas the minimization procedure yields the following desired matrix:

\[
\tilde{A} = \begin{bmatrix}
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
-0.802 & 0.651 & -3.423 & 1.484 & -3.510 & 0.634 \\
0.554 & -0.802 & 1.261 & -3.423 & 0.539 & -3.510 \\
\end{bmatrix}; \quad \tilde{B} = \begin{bmatrix}
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
0.157 & 0 \\
\end{bmatrix}
\]

Using the resultant feedback matrix, the control and the output signals of the closed-loop system are shown in Figure (1) and Figure (2), respectively. Comparing the maximal amplitude of the control signals in our method with the Preuß method [8] as worked by Schmidt [10] reveals a reduction of about 27% in the maximal control signals.

**4.4.2 Example 4.4.2**

In order to compare our method with Ramar's method [7], we shall consider the following system matrices:
Figure 1. The control signals of the closed-loop system of example 4.4.1.
Figure 2. The output signals of the closed-loop system of example 4.4.1.
The set of the desired closed-loop eigenvalues is as follows:

\[ \Lambda^d = \{-1, -2, -3\} \]

Let the desired state feedback matrix be given by:

\[ K = \begin{bmatrix} k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 \end{bmatrix} \] (4.4.8)

Following Ramar’s method [7], one has two choices:

1. Using the first row of \( K \) to achieve the desired closed-loop pole locations while using the second row to minimize the following functional:

\[
J = \frac{(3 + 8k_4 + k_4k_5 + k_4k_6)^2 + (21 + k_4 + 9k_5 + k_5^2 + k_6^2 + k_5k_6)^2 + (6 + k_5 + 9k_6 + k_5k_6 + k_6^2)^2}{(1 + k_4 + k_5 + k_6)^2}
\]

\[
\implies k_4 + k_5\]

The minimization problem and the subsequent steps result in the following state feedback matrix:

\[
K = \begin{bmatrix} -2.286683 & -0.677750 & -2.330602 \\ -0.105539 & -3.587093 & 0.204378 \end{bmatrix}
\] (4.4.10)

which has a norm of:
\[ \|K\| = \left( \sum_{i=1}^{6} k_i^2 \right)^{1/2} = 4.903 \] (4.4.11)

2. Using the second row of K for pole placement while using the first row to minimize the following functional:

\[
J = k_1^2 + k_2^2 + k_3^2 + \{(-6 - 22k_1 + 6k_2 + 6k_3 - 12k_1^2 - 12k_1k_3 - 2k_3^2
- 4k_1^2k_3 - 2k_1k_3^2)^2 + (6 - 14k_1 + 6k_2 + 42k_3 - 14k_1k_2 - 2k_1k_3 - 14k_2k_3 - 2k_1
- 2k_1k_2 - 4k_1k_2k_3 - 2k_2k_3^2)^2 + (6 + 2k_1 + 6k_2 + 26k_3 + 2k_1k_2
+ 2k_2k_3 + 2k_1k_3 + 14k_1k_3 + 14k_3^2 + 2k_3^3)^2 / (-2k_1 + 2k_3
- 2k_1k_2 - 2k_1k_3 - 2k_2k_3 - 2k_3^2) \}
\]

(4.4.12)

Then, the state feedback matrix and its norm will be:

\[
K = \begin{bmatrix}
0.357097 & 3.073809 & -1.36040 \\
-1.498064 & -3.302513 & -3.694375 \\
\end{bmatrix}
\] (4.4.13)

\[ \|K\| = 6.182 \] (4.4.14)

Therefore, the second choice results in a larger norm than the first one, and this method fails to determine which choice to be used a priori in order to obtain the minimum norm feedback matrix. Moreover, the problem will be escalated when the number of inputs increases. Also, as depicted from Eqs. (4.4.9) and (4.4.12), the functional in this method is not smooth. Therefore, for systems of order four or higher,
it is tedious and sometimes impossible to search for the minimum value of the functional with respect to all of the variables.

Using the method proposed here, since the first three columns of the controllability matrix are linearly dependent, algorithm 4.2 and the tuning process give the following transformation matrix:

\[
T = \begin{bmatrix}
0.1 & 0 & 0.1 \\
-1 & -1 & 1 \\
0 & -2 & 0
\end{bmatrix}
\] (4.4.15)

Here, the tuned scalar is multiplied by \( t' \) and a value of 0.1 is chosen for the scalar. The tuning process will be explained in detail in the next example. The transformed system matrices are given by:

\[
\tilde{A} = \begin{bmatrix}
0.5 & 0.05 & -0.075 \\
0 & 0 & 1 \\
-10 & -1 & 1.5
\end{bmatrix}, \quad \tilde{B} = \begin{bmatrix}
0.2 & 0.1 \\
0 & 0 \\
0 & -2
\end{bmatrix}
\] (4.4.16)

Then, the functional to be minimized is:

\[
J = (a_{11}' - 0.5)^2 + (a_{12}' - 0.05)^2 + (a_{13}' + 0.075)^2 + (a_{31}' + 10.0)^2
+ (a_{32}' + 1.0)^2 + (a_{33}' - 1.5)^2
\] (4.4.17)

subject to the constraints:

\[
a_{11}' + a_{33}' = -6
\]
\[
a_{11}'a_{33}' - a_{12}'a_{31}' - a_{32}' = 11
\] (4.4.18)
\[
a_{11}'a_{32}' - a_{12}'a_{31}' = 6
\]
where \( a_{ij} \) is the \((i,j)\)th element of the matrix \( A^d \).

This minimization problem results in the following desired matrix:

\[
A^d = \begin{bmatrix}
-3.499808 & 0.222108 & 0.116973 \\
0 & 0 & 1 \\
-10.006199 & -1.079358 & -2.500192
\end{bmatrix}
\]  \hspace{1cm} (4.4.19)

Then, the desired state feedback matrix and its norm will be:

\[
K = \begin{bmatrix}
-2.8408 & -0.76033 & -1.1594 \\
-0.039369 & -4.0399 & 0.039989
\end{bmatrix}
\]  \hspace{1cm} (4.4.20)

\[
\|K\| = 5.129
\]  \hspace{1cm} (4.4.21)

Comparing the result in Eq. (4.4.21) with that of Eqs. (4.4.11) and (4.4.14), one can see that the result obtained by using the proposed method is very close to that of Eq. (4.4.11) and better than Eq. (4.4.14). However, the superiority of the proposed method is most appreciated when the order of the system is larger than four. In this case, it will be very difficult to apply Ramar’s method [7].

It is to be noted that for the special case when \( n - m < m \), as in this example, it is possible to apply algorithms 4.1 and 4.2 equally. However, both methods would give the same result.
4.4.3 Example 4.4.3

This example illustrates how the degrees of freedom which are built in the transformation matrix can be utilized in our approach. The system matrices in this example are:

\[
A = \begin{bmatrix}
1 & 0 & 2 & 1 \\
-1 & 1 & 0 & 1 \\
3 & 0 & 1 & 1 \\
1 & 1 & 0 & 0
\end{bmatrix} \quad \text{;} \quad B = \begin{bmatrix}
1 & 1 \\
1 & 0 \\
0 & 1 \\
-1 & 2
\end{bmatrix}
\] (4.4.22)

The open-loop system poles are:

\[
\Lambda = \{ 1.685, 3.623, -0.724, -1.584 \}
\]

whereas, the desired set of the closed-loop poles is:

\[
\Lambda^d = \{ -1, -2, -3, -4 \}
\]

Here, algorithm 4.1 is applicable with the following results:

\[
\begin{align*}
t^1[B] &= 0 \\
t^2[B] &= 0 \\
t^3 &= t^1A \\
t^4 &= t^2A
\end{align*}
\] (4.4.23)

Multiplying the row vector \( t^i \) by a variable scalar, and plotting the norm of \( T \) and the norm of the generalized inverse of \((TB)\) versus the variable scalar, one obtains Figure (3).
Figure 3. The norm of the matrix $T$ and the norm of the generalized inverse of $(TB)$ versus the variable scalar of example 4.4.3.
In this figure, one observes that the norm of $T$ levels off for values of the scalar $\leq 0.5$, whereas the norm of the generalized inverse of $(TB)$ levels off for values $\geq 2$. Therefore, we have three possibilities that we may try. These possibilities are: any value of the scalar $\leq 0.5$, any value $\geq 2$ and the nominal value (i.e., 1). The experience with this method shows that the resultant state feedback matrix has an almost constant norm in each of these intervals. After trying these three possibilities one obtains that the values of the scalar $\leq 0.5$ give a state feedback matrix with minimum norm. For example, a value of the scalar equals to 0.1 will give the following transformation matrix:

$$T = \begin{bmatrix} -0.1 & 0.1 & 0.1 & 0 \\ -2 & 3 & 0 & 1 \\ 0.1 & 0.1 & -0.1 & 0.1 \\ -4 & 4 & -4 & 1 \end{bmatrix}$$

Then, the transformed system matrices will be given by:

$$\tilde{A} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -3.75 & 0.35 & -1.75 & -0.075 \\ 17.5 & -3.5 & -52.5 & 4.75 \end{bmatrix} ; \quad \tilde{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.1 & 0.2 \\ -1 & -6 \end{bmatrix}$$

whereas, the minimization procedure will give the following desired matrix:
Finally, the desired state feedback matrix and its norm are:

\[
K = \begin{bmatrix}
-0.31967 & -24.13 & 5.0144 & -14.108 \\
-4.8532 & 9.1887 & -5.2213 & 3.7082
\end{bmatrix}
\] (4.4.27)

\[
\|K\| = 30.9117
\] (4.4.28)

In comparison, the state feedback matrix has a norm of 49.217 at the nominal value of the scalar and a norm of 39.931 at the interval where the value of the scalar \( \geq 2 \).

4.4.4 Example 4.4.4

Consider the lateral attitude control of a drone aircraft reported by Ridgely and Banda [87]. The system matrices are given by:
The open-loop poles of the plant are:

- Spiral Mode: $-0.036$
- Dutch Roll (unstable): $0.1884 \pm j1.0511$
- Roll Convergence: $-3.2503$
- Eleven Actuator: $-20$
- Rudder Actuator: $-20$

The desired closed-loop eigenvalues are:

\[ \Lambda^d = \{-0.5 \pm j1, -1, -4, -20, -20\} \]

Using algorithm 4.1 and the tuning process with respect to the first row vector of $T$ as shown in Figures (4) and (5), the obtained transformation matrix is as follows:

\[
A = \begin{bmatrix}
-0.08527 & -0.001423 & -0.9994 & 0.04142 & 0 & 0.1862 \\
-46.86 & -2.757 & 0.3896 & 0 & -124.3 & 128.6 \\
-0.4248 & -0.06224 & -0.06714 & 0 & -8.792 & -20.46 \\
0 & 1 & 0.0523 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -20 & 0 \\
0 & 0 & 0 & 0 & 0 & -20
\end{bmatrix} \quad (4.4.29)
\]

\[
B = \begin{bmatrix}
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
20 & 0 \\
0 & 20
\end{bmatrix}, \quad C = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0
\end{bmatrix} \quad (4.4.30)
\]
Figure 4. The norm of the matrix $T$ versus the variable scalar of example 4.4.4.
Figure 5. The norm of the generalized inverse of $(TB)$ versus the variable scalar of example 4.4.4.
Using this transformation matrix, the transformed system matrices are given by:

\[
\tilde{A} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \end{bmatrix} \quad ; \quad \tilde{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (4.4.32)
\]

Then, the minimization procedure gives the following desired matrix:

\[
A^d = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \end{bmatrix} \quad (4.4.33)
\]

Finally, Eq. (4.2.7) yields the following feedback matrix:
which has a norm of:

\[ \|K\| = 0.180 \] (4.4.35)

The next example shows how the non-canonical approach for pole-placement can be used.

4.4.5 Example 4.4.5

Consider the 5-th order system with three inputs which was proposed in [57]. The system matrices are:

\[
A = \begin{bmatrix}
2 & 1.25 & -2.25 & -3.5 & 1 \\
-5 & -1 & 3 & 0 & -5 \\
-5 & 3 & -1 & 0 & -5 \\
0 & -0.75 & 0.75 & -2.5 & 0 \\
-5 & -1.25 & 2.25 & 3.5 & -4
\end{bmatrix}
; \quad B = \begin{bmatrix}
1 & -1 & 1 \\
0 & 2 & 2 \\
2 & 2 & 0 \\
-1 & 0 & 2 \\
0 & 2 & 0
\end{bmatrix}
\] (4.4.36)

The open-loop poles are:

\[ \Lambda = \{1, 2, -2.5, -3, -4\} \] (4.4.37)

whereas, the desired set of eigenvalues is:

\[ \Lambda^d = \{-1, -2, -5, -1 + j1, -1 - j1\} \] (4.4.38)
Choosing for $A_d$ the following matrix:

$$
A_d = \begin{bmatrix}
-1 & 0 & 0 & 0 & 0 \\
0 & -2 & 0 & 0 & 0 \\
0 & 0 & -5 & 0 & 0 \\
0 & 0 & 0 & -1 & 1 \\
0 & 0 & 0 & -1 & -1
\end{bmatrix}
$$

(4.4.39)

The singular value decomposition of $B^T$ yields the following transformation matrix:

$$
T = \begin{bmatrix}
.546 & -.055 & -.382 & -.219 & .710 \\
0 & -.632 & .316 & .632 & .316 \\
-.064 & .656 & .582 & .137 & .455 \\
-.175 & -.408 & .498 & -.730 & .147 \\
-.816 & 0 & -.408 & 0 & .408
\end{bmatrix}
$$

(4.4.40)

Using this transformation matrix, the matrices $B_{c_2}$, $A_{c_{11}}$ and $A_{c_{12}}$ will be:

$$
B_{c_2} = \begin{bmatrix}
.964 & 3.451 & 1.524 \\
1.551 & .650 & -2.452 \\
-1.633 & .816 & -.816
\end{bmatrix}
$$

(4.4.41)

$$
A_{c_{11}} = \begin{bmatrix}
-.572 & 1.525 \\
.682 & -.975
\end{bmatrix}
$$

(4.4.42)

$$
A_{c_{12}} = \begin{bmatrix}
-.376 & -.113 & .006 \\
-.053 & -.409 & .549
\end{bmatrix}
$$

(4.4.43)

Choosing for $T_{c_2}$ the following random matrix:
Then, solving the Lyapunov matrix equation yields:

\[
T_{a2} = \begin{bmatrix}
-0.286 & 2.171 & 1.048 & -0.260 & 0.116 \\
2.509 & 0.580 & -0.252 & 1.001 & 1.420 \\
2.156 & 2.641 & 0.867 & -0.100 & 1.388
\end{bmatrix}
\] (4.4.44)

Using \( T_{a1} \) and \( T_{a2} \), one can obtain the following value for \( A_{dc2} \):

\[
A_{dc2} = \begin{bmatrix}
4.188 & 5.503 & -2.006 & -3.842 & 3.084 \\
7.429 & 7.734 & -8.546 & -5.115 & 4.064
\end{bmatrix}
\] (4.4.46)

Finally, Eq. (4.3.8) gives the following state feedback matrix:

\[
K = \begin{bmatrix}
0.500 & -0.915 & -4.006 & 0.070 & 0.318 \\
-3.395 & 3.773 & 5.408 & -1.130 & -4.271 \\
0.276 & -1.824 & -2.185 & 0.618 & 1.398
\end{bmatrix}
\] (4.4.47)

which places the closed-loop poles of the system at the desired locations.
Chapter 5

POLE-PLACEMENT BY OUTPUT FEEDBACK

5.1 Historical Background

Since the fundamental result was presented by Wonham [22], the problem of pole assignment has received much attention and generated a considerable number of articles. The result of Wonham states that it is possible to assign an arbitrary self-conjugate set of eigenvalues to the closed-loop system by an appropriate state feedback if the system is controllable. However, in industrial processes and due to practical reasons, only some of the state variables are actually measured. This observation suggests that state feedback control would be unfeasible. The alternatives are either: a) to estimate the unavailable state variables (see Kalman and Bucy [23] and Luenberger [24]) or b) to use output feedback. However, when the control system uses communication links, its reliability reduces while its complexity and cost increase. Therefore, these reasons favor using output feedback.
The problem of pole assignment in a linear, time-invariant multivariable system via direct output feedback has received substantial contributions. The first contribution to this problem was that of Davison [25]. Davison showed that if the system is both controllable and observable, then \( p \) poles of the closed-loop system are assignable almost arbitrarily by gain output feedback. In the sequel, the symbols \( n, m \) and \( p \) denote the order of the system, the number of inputs and the number of outputs, respectively. The result in [25] was extended by Davison and Chatterjee [26] and by Sridhar and Lindorff [27] who showed that under the same conditions of [25], \( \max(m, p) \) eigenvalues are assignable almost arbitrarily by gain output feedback. Then, Kimura [28] derived a simple condition for pole assignability using an approach based on the properties of the eigenvalues of the closed-loop system. His condition says that for a controllable and observable system if \( n \leq m + p - 1 \), then almost all distinct complex numbers are assignable as closed-loop poles. However, a slightly more general result was given by Davison and Wang [29] and by Topaloglu and Seborg [30]. Their result states that for a controllable and observable system with \( \text{rank}(B) = m \) and \( \text{rank}(C) = p \), for almost all \((B, C)\) pairs, \( \min(n, m + p - 1) \) poles can be assigned arbitrarily close to a desired set of eigenvalues by using output feedback. In a follow-up paper, Kimura [31] gave a better result by employing a geometric approach. His result roughly says that an arbitrary set of complex numbers are assignable by constant gain output feedback to almost all systems (excluding some pathological cases) if 1) \( n < m + p + \nu - 1 \), 2) \( m > \mu \), and 3) \( p \geq \nu \), where \( \nu \) and \( \mu \) are the controllability and observability indices, respectively. However, the results in the above papers are not necessary conditions and they fail to show a clear connection between assignability, controllability and observability. They also do not give simple methods for the calculation of the output feedback matrix.
Using a version of the implicit function theorem, Hermann and Martin [32] prove that for almost every linear system whose first $n$ Markov matrices, $CB, CAB, \ldots, CA^{n-1}B$, are linearly independent, generic pole assignment is possible provided the output feedback matrix is allowed to be complex-valued. Willems and Hesselink [33] show that for almost all systems with $m = p = 2$ and $n = 4$ ($mp = n$) generic pole assignment with real output feedback matrix is not possible. On the other hand, Brockett and Byrnes [34] take advantage of certain classical ideas based on elimination theory to develop a formula which gives values of $m, p$ and $n$, for which almost every linear system is generically assignable. In particular, they show that if either $\min(m, p) = 1$ or $\min(m, p) = 2$ and $\max(m, p) = 2^k - 1$, then $mp = n$ is a sufficient condition for generic pole assignment. However, their development is not constructive. Using a different approach, Morse, Wolovich and Anderson [35] give a constructive proof of the fact that with $m = 3$, $p = 2$, $n = 6$, $mp = n$ is a sufficient condition for generic pole assignment. Magni [36] identified a pathological case in which pole assignability is not possible by output feedback. He showed that assignability, when $n$ is odd and $n + 1 = m + p$, is not possible if only one of the desired closed-loop eigenvalues is real and equal to an invariant zero of the system.

It is known that a direct attempt to find a feedback matrix so that the closed-loop system has prescribed eigenvalues requires the simultaneous solution of a set of non-linear equations. However, this can be avoided by adopting the dyadic feedback control approach. In this approach, one can apply the single input results to the multi-input case by simply predefining the output feedback law in the dyadic form $K = gd\tau$ where $g$ and $d$ are vectors of dimension $m$ and $p$, respectively. Then the multi-input system $\dot{x} = Ax + Bu, \ y = Cx, \ u = Ky$ is reduced to the equivalent single input system $\dot{x} = Ax + Bg\hat{u}, \ y = Cx, \ \hat{u} = d\tau y$. Davison [25], Fallside and Seraji [37],
Fallside and Patel [38], Power [39], Sridhar and Lindorff [27], Seraji [40] and Power [41] have examined the various aspects of this approach. Although the dyadic approach has considerable elegance and simplicity, the resulting closed-loop systems have poor disturbance rejection properties compared with the full-rank case (see Daniel [42]). Another drawback of the dyadic approach is that it greatly reduces the attainable set of closed-loop eigenvalues so that only \( \max(m,p) \) poles can be placed arbitrarily (Davison and Chatterjee [26]).

Another approach is to find an exact solution to the equation \( K_rC = K_s \) where \( K_s \) is the state feedback matrix which achieves the same pole-placement required by the output feedback matrix. Munro and Vardulakis [43] state that a necessary and sufficient condition to assign all the poles by constant output feedback is that at least one state feedback matrix \( K_x \) which achieves the same pole-placement and one \( g_r \)-inverse of \( C \) satisfy the consistency relationship \( K_x C^{-1} C = K_s \). In this case, the output feedback matrix is given by \( K_r = K_s C^{-1} \). Then, Seraji [44], Munro [45], Patel [46] and Vardulakis [47,48] have all examined this problem by essentially restricting the structure of the state feedback matrix \( K_x \) so that the consistency condition holds. Nevertheless, these methods have been trial and error in nature and experience has shown that they are not really workable for large order systems.

A direct approach for the pole-assignment problem by output feedback is to express the coefficients of the characteristic polynomial of the closed-loop system as functions of the elements of the feedback matrix and solve the resulting set of simultaneous polynomial equations. Along this direction, Tarokh [49] investigated the possibility that some of these equations could be linear and gave the conditions for that. Then, the results were used to determine the existence of stabilizing and pole
assigning output feedback matrices. He also gave the maximum number of the arbitrarily assignable poles. Specifically, he stated that for a "v-linear characteristic coefficient" system, the maximum number of arbitrarily assignable poles by constant output feedback is \((n - p)\) where \(p\) is the rank deficiency of the matrix 

\[ E_v = (e_v, e_v, ..., e_v) \]

and \(e_v\)'s are \(pm \times 1\) vectors formed from the columns of the first \(v\) Markov matrices. Then, Kaizuka [50] pointed out that \(n \leq \max(m, p) + 2\) is a necessary condition for Tarokh's complete assignability theorem to be satisfied. However, Kolka [51] gave a counter example for Kaizuka's theorem. He also pointed out and corrected an error in Tarokh's theorem. Then, he gave necessary and sufficient conditions for a system to be \(n\)-linear characteristic coefficient.

It is well known that the dynamics of a linear multivariable system determined not only by its eigenvalues but also by its eigenvectors. Therefore, the entire eigenstructure assignment problem, where it is required to assign closed-loop eigenvectors and closed-loop eigenvalues, received considerable attention (see Kimura [28], Moore [52], Porter and D'Azio [53]). Porter and Bradshow [54] applied the method of the entire eigenstructure assignment to the design of output feedback regulators. They showed that, in the case of self-conjugate distinct eigenvalues, the closed-loop eigenstructure assignable by an output feedback is constrained by the requirement that the eigenvectors and the reciprocal eigenvectors must lie in certain subspaces. In Fletcher [55], an algorithm using ordinary procedures of linear algebra was presented to select these left and right eigenvectors in the case \(m + p > n\). Clearly, it is because of these severe constraints on the closed-loop eigenstructure that it is frequently impossible to achieve satisfactory closed-loop behavior for the linear multivariable continuous-time system by means of static output-feedback controllers.
On the other hand, several researchers explored the numerical aspects of the pole-placement problem by output feedback. Patel [56], Ramar and Gourishankar [57] described iterative numerical methods for the calculation of the desired output feedback matrix. These methods assume the existence of a solution, and the convergence of the iterations depends heavily on the initial values chosen. Furthermore, they did not give the number of poles that can be placed arbitrarily using an output feedback. Lee [58] focused on the coefficients of the characteristic polynomial and performed a gradient search for a feedback matrix which minimizes a performance index. Kabamba and Longman [59] developed an algorithm which assigns the closed-loop poles indirectly by assigning values to the functions $T_i = \text{tr}[(A - BKC)^i] ; 1 \leq i \leq n$. The solution was made possible by evaluating the Moore-Penrose pseudo-inverse of an appropriate gradient matrix written as an $n \times mp$ array. They showed that a necessary and sufficient condition for local assignability at $K$ for the case of distinct eigenvalues is that the matrices $C(A - BKC)^iB ; 1 \leq i \leq n$ are linearly independent. The gradients used in [58] are also found by Godbout and Jordan [60] who gave assignability conditions but stopped short of developing an algorithm. These conditions were similar to that of [59], but are not limited to the case of distinct eigenvalues. In addition, they presented a test procedure which estimates the number of assignable poles in the system when constant output feedback is used. Sevaston [61] examined the pole placement problem through output feedback by investigating the effect of small changes in the feedback matrix on the closed-loop poles locations. The main outcome of his investigation is a feasible directions search procedure for finding the desired feedback matrix. Recently, Patel and Misra [62] described numerical algorithms to compute the output feedback matrix for both constant and dynamic cases. They used the fact that closed-loop eigenvalues can be assigned arbitrarily close to a desired set of eigenvalues if $m + p > n$. In this paper,
the eigenvalue assignment problem was treated as the inverse of the algebraic eigenvalue problem. Their method was based on the QR algorithm for solving the algebraic eigenvalue problem. The same technique was used to handle the case when \( m + p \leq n \) by means of dynamic output feedback controller.

Still, another method is to use a dynamic controller of the form:

\[
\dot{z} = Dz + Fy
\]

\[
u = K_z z + K_y y
\]

where \( z \) is an \( r \)-dimensional vector representing the states of the dynamic controller. It can easily be shown that an \( r \)-th order dynamic controller greatly increases the degrees of freedom for pole placement to \( (m + r) \times (p + r) \). However, introducing this dynamic controller would result in an overall system of order \( n + r \). Brasch and Pearson [63], Ahmari and Vacroux [64], Patel [65] and Sirsena and Choi [66] have all examined the many aspects of pole placement with dynamic controllers. In particular, Brasch and Pearson [63] showed that for a controllable and observable system, a compensator of order \( q = \min(v - 1, \mu - 1) \) is sufficient to achieve pole assignment, where \( v \) and \( \mu \) are the controllability and observability indices, respectively. Chen and Hsu [67] and Seraji [68] have extended the idea of dyadic compensators to the dynamic case. Antsaklis and Wolovich [69] have shown that with a compensator of order \( q \), \( \min(n + q, m + p + 2q - 1) \) closed-loop poles can be generically assigned. However, this has led to a worse bound, in many cases, than the earlier result of [63]. Then, Willems and Hesselink [33] have showed that \( q(m + p - 1) + mp > n \) is a necessary condition for the generic pole assignment by using a proper output feedback compensators of order \( q \). Munro and Novin-Hirbod [70] presented a method to design
a minimum degree full-rank output-feedback compensator for arbitrary pole-placement. The desired compensator constructed from a minimal sequence of dyads. This method requires the solution of a linear systems of algebraic equations and uses the matrix generalized inverse. Using a different approach, Djaferis [71], [72] showed that, for \( p \geq m \), \( \min(n + q, (q + 1)p + q) \) closed-loop poles can be generically assigned by employing a proper output feedback compensator of order \( q \). Using their previous tools, Djaferis and Narayana [73] proved that for strictly proper systems with McMillan degree \( n \) and controllability indices \( \mu_1 \geq \cdots \geq \mu_m > 0 \), one can arbitrarily assign \( \min((q + 1)p + q + b(m - 1), n + q) \) closed-loop poles with a proper compensator of order \( q \), where \( b = \min\left(\left\lfloor \frac{p}{m} \right\rfloor, \mu_m \right) \) and \( \left\lfloor \frac{p}{m} \right\rfloor \) stands for the largest integer smaller than or equal to \( \frac{p}{m} \). In many cases, this result leads to a lower bounds for complete pole assignment with dynamic compensators. For example, if \( p = 8, m = 2 \) and \( n = 19 \), the method in [63] would require a compensator of order 2 for a complete pole assignment, where this method requires only a compensator of order 1.

Some methods which allow the explicit characterization of freedom in the feedback matrix for pole-placement problem are developed. Sebok, Richter and Decarlo [88] presented a method for minimizing the norm of a decentralized output feedback matrix which assigns a specified set of eigenvalues. The method is based on an iterative algorithm which, when initialized at a feedback assigning desired poles, produces a new feedback having a locally minimum norm while maintaining the eigenvalues constraint. The structure of the algorithm is derived from the first order variational behaviour of the eigenvalues with respect to variations in the feedback matrix. The proposed procedure is applicable to a non-decentralized system by considering the interconnected systems as having a single component. However, the
method assumes one has already computed an initial decentralized feedback matrix which assigns the desired eigenvalues. Furthermore, it requires computation of eigenvalues, eigenvectors, pseudo-inverse and bases for a particular null space which makes it computationally expensive.

5.2 An Iterative Algorithm

In this section, we shall review the iterative algorithm for output feedback pole-placement presented by Bingulac and Mitrovic [85]. This method of pole-placement has several advantages. First, the algorithm does not require the usual condition \( n < m + p \); however, even for \( m + p < n \) it gives an approximate solution. Second, the algorithm yields a full rank output feedback matrix. Third, it is applicable to nonminimal systems. However, the set of desired closed-loop poles, in this case, must contain all uncontrollable and unobservable modes. Finally, the algorithm treats the feedback matrix as a vector which can be obtained by the least square method. The latter feature is significant because our objective is to find a minimum norm output feedback matrix for the pole-placement problem.

We shall start with the development of the algorithm, then, extend it to obtain the minimum norm feedback matrix.
5.2.1 Algorithm Development

Consider the following realization describing a linear time-invariant multivariable system:

\[ R = \{ C, A, B \} \quad (5.2.1) \]

where the dimension of the state, input and output vectors are \( n, m \) and \( p \), respectively. It is required to find an output feedback \( m \times p \) matrix, \( K \), which satisfies the following condition:

\[ \Lambda(A + BKC) = \Lambda^d \quad (5.2.2) \]

where \( \Lambda(\cdot) \) denotes the eigenvalues of the argument matrix, and \( \Lambda^d \) is the set of desired closed-loop eigenvalues.

In other words, we need the characteristic polynomial of the closed-loop system to equal to the desired characteristic polynomial associated with the given \( \Lambda^d \), i.e.,

\[ \det(sI - A_c) = f_d(s) \quad (5.2.3) \]

where \( A_c = A + BKC \) and \( f_d(s) \) is the desired characteristic polynomial. Obviously, Eq. (5.2.3) will be satisfied if the coefficients of the two polynomials are equal, i.e.,

\[ f(K) = f_d \]

where \( f(K) \) and \( f_d \) are the \( n \)-dimensional vectors containing the coefficients of the polynomials \( \det(sI - A_c) \) and \( f_d(s) \), respectively, i.e.,
Expanding the vector function $f(K + \Delta K)$ using Taylor’s expansion, we get:

$$f(K + \Delta K) = f(K) + \left( \frac{\partial f}{\partial K} \right)_K \Delta k + o(\Delta K)$$

(5.2.4)

where $\left( \frac{\partial f}{\partial K} \right)_K$ denotes the matrix of partial derivatives of $f(K)$ with respect to the elements $k_i$, and the $mp$-dimensional vector $\Delta k$ has the following form:

$$\Delta k = \begin{bmatrix} k^1 \\ \vdots \\ k^p \end{bmatrix}$$

(5.2.5)

where $k^1, \ldots, k^p$ are the columns of the matrix $\Delta K$. A method for calculating $\left( \frac{\partial f}{\partial K} \right)_K$ will be given later.

Based on the previous development, an algorithm for solving the output feedback pole-placement problem can be formulated by the following steps.

**Algorithm 5.1: Pole-placement algorithm**

**Step 1:** Set $i = 0$, $K_i = K_0$, $i_m = i_{\text{max}}$, $e = e_0$ where $K_0$, $i_{\text{max}}$ and $e_0$ are initial guess for $K$, maximum number of iterations, and a sufficiently small positive number, respectively.

**Step 2:** Evaluate $f(K_i)$ and $\left( \frac{\partial f}{\partial K} \right)_{K_i}$
Step 3: If \( |f_{ij} - f_i(K_i)| < \varepsilon |f_{ij}| \) for \( j = 1, \ldots, n \), set \( K = K_i \) and stop; else go to step 4.

Step 4: Solve \( \left( \frac{\partial f}{\partial K} \right)_{K_i} \Delta k_i = f - f(K_i) \) for \( \Delta k_i \).

Step 5: Rearrange the elements of \( \Delta k_i \) into \( \Delta K_i \), and set \( K_{i+1} = K_i + \Delta K_i \) and \( i = i + 1 \).

Step 6: If \( i \leq i_m \) go to step 2; else, stop.

In order to guarantee the convergence of the algorithm, step 5 should be replaced by a suggested stabilizing procedure as given in [85].

Next, we shall review a method for calculating \( \left( \frac{\partial f}{\partial K} \right)_{K} \).

### 5.2.2 Derivatives of Characteristic Polynomials

The derivatives of the characteristic polynomials \( \left( \frac{\partial f}{\partial K} \right)_{K} \) can be calculated using the method proposed by Bingulac [86]. Denote the partial derivative of the \( l' \)th coefficient \( f_i \) of the characteristic polynomial of the triple \( \{C, A + BK, B\} \) with respect to the element \( k_{ij} \) by:

\[
g_{ij}' = \frac{\partial f_i}{\partial k_{ij}} \quad (5.2.6)
\]

Then, \( g_{ij}' \) may be given by:
\[ g'_{ij} = -q^l_{ij}; \quad l = 1, \ldots, n; \quad i = 1, \ldots, m; \quad j = 1, \ldots, p \quad (5.2.7) \]

The quantities \( q^l_i \) are the coefficients of the following polynomial matrix:

\[
Q(s) = \begin{bmatrix}
q_{11}(s) & \cdots & q_{1p}(s) \\
\vdots & \ddots & \vdots \\
q_{m1}(s) & \cdots & q_{mp}(s)
\end{bmatrix} \quad (5.2.8)
\]

and

\[
q_{ij}(s) = \sum_{l=1}^{n} q^l_{ij} s^{l-1} \quad (5.2.9)
\]

The matrix \( Q(s) \) is the numerator of the following transfer function:

\[
W(s) = \frac{Q(s)}{f(s)} = B^T [sI - A_c^T]^{-1} C^T \quad (5.2.10)
\]

The coefficients \( q^l_i \) of the polynomial \( q_i(s) \) can be arranged in an \( n \)-dimensional vector \( q_i \) given by:

\[
q^i = [q^1_i \quad q^2_i \quad \cdots \quad q^n_i]^T \quad (5.2.11)
\]

In conclusion, the partial derivative \( n \times mp \) matrix, \( \frac{\partial f}{\partial K} \), is given by:

\[
\frac{\partial f}{\partial K} = - [q^{11} \ | \ q^{21} \ | \ \cdots \ | \ q^{m1} \ | \ q^{12} \ | \ q^{22} \ | \ \cdots \ | \ q^{1p} \ | \ \cdots \ | \ q^{mp}] \quad (5.2.12)
\]
5.3 *Extended Iterative Algorithm*

The objective of this section is to extend the previous iterative algorithm so that a minimum norm output feedback can be obtained. The idea is to apply the successive pole shifting method to the iterative algorithm. The use of the successive shifting procedure helps to reduce the norm of the output feedback because small displacement of the poles requires only small effort. Moreover, experience has shown that the total feedback matrix usually has a low norm, too.

In the proposed method, the poles of the system should be shifted successively through straight paths starting from the open loop poles and ending at the desired poles. These loci ensure that the effort needed is minimum. The desired poles at each iterative step can be found using the following relation:

\[ E = \alpha E_d + (1 - \alpha) E_o \]  \hspace{1cm} (5.3.1)

where \( E, E_d \) and \( E_o \) are \( n \times 2 \) matrices containing the desired poles at each iterative step, the desired final closed-loop poles, and the open-loop poles, respectively. The first column of these matrices contains the real parts of the eigenvalues, whereas the second column contains the imaginary parts. The increment of \( \alpha \) reflects the total number of the successive pole shifting, for example if \( \Delta \alpha = 0.1 \) that means the poles are being shifted in 10 steps. However, before using Eq. (5.3.1), the eigenvalues in the matrix \( E_d \) should be ordered according to the matrix \( E_o \), so that each real eigenvalue in \( E_d \) should be placed corresponding to the nearest open-loop real eigenvalue. Also, each complex eigenvalue in \( E_d \) should be placed corresponding to the nearest open-loop complex eigenvalue in \( E_o \). However, when a complex conjugate...
pair of eigenvalues are to be shifted to a pair of real eigenvalues or vice versa, we need to shift this complex pair successively to a pair of repeated real eigenvalues, then from these repeated eigenvalues to the final desired real pair.

The new algorithm can be outlined by the following steps:

**Algorithm 5.2: Extended Iterative Algorithm**

Step 1: Form the matrix $E_\alpha$ from the desired eigenvalues.

Step 2: Set $\alpha = 0$ and choose the iteration step size $\Delta \alpha$.

Step 3: Set $\alpha = \alpha + \Delta \alpha$

Step 4: Calculate the matrix $E$ using Eq. (5.3.1).

Step 5: Find the vector $f_\alpha$ using the obtained value for $E$.

Step 6: Go to algorithm 5.1 and find $K_r$.

Step 7: Set $K_\alpha = K_r$.

Step 8: If $\alpha < 1$ go to step 3; else, stop.

However, as mentioned earlier, if a pair of complex conjugate poles in $E_\alpha$ is to be shifted to a pair of real poles or vice versa, algorithm 5.2 should be used in two stages. At the first stage $E_\alpha$ takes an intermediate value $E_i$; then at the second stage $E_\alpha = E_i$ and $E_\alpha$ takes the final desired set of eigenvalues.
Another alternative shifting scheme is to shift the poles according to a successive change of their characteristics polynomial coefficients. In this case, the desired characteristics polynomial coefficients at each iterative step can be calculated using the following relation:

\[ f_i = \alpha f_{d} + (1 - \alpha) f_{o} \]  

(5.3.2)

where \( f_{d} \) and \( f_{o} \) are the n-dimensional vectors containing the coefficients of the open-loop and the desired closed-loop polynomials, respectively. This shifting scheme has the advantage of shifting the poles in one stage. Therefore, this scheme is preferable when a complex conjugate pair is to be shifted to a pair of real poles or vice versa. Otherwise, the choice between the two scheme is left to the designer.

Experience with the extended iterative algorithm shows that it converges to a solution if such solution exists without incorporating the stabilizing procedure. This is the case, for example, when the system has sufficient number of inputs and outputs (i.e. \( p + m - 1 \geq n \)), or even when \( mp < n \) provided the existence of an exact solution. However, when an exact solution does not exist, the method needs to incorporate the following stabilizing procedure in order to give an approximate solution. This procedure should replace step 5 of algorithm 5.1.

Algorithm 5.3: Stabilizing Procedure

Step 1: Set \( j = 0, \quad d_i = \|f_{d} - f(K_i)\|, \quad b = 0.5, \quad a_j = 2. \)

Step 2: Set \( a_{j+1} = a_j b, \quad j = j + 1. \)

Step 3: If \( j > i_m \) stop; else, go to step 4.
Step 4: Set $K' = K_i + a_i \Delta K_i$ and evaluate $f(K')$.

Step 5: Set $d_z = \| f_e - f(K') \|$.

Step 6: If $d_z > d_1$, go to step 2; else, go to step 7.

Step 7: Set $K_i = K'$ and return to step 6 of algorithm 5.1.

This stabilizing procedure ensures that the approximate solution will minimize the quantity $\| f_e - f(K) \|$. Moreover, it is preferable to use this stabilizing procedure at all the times to ensure that the feedback matrix has small norm.

The implementation of the iterative algorithm incorporates the least square method to solve the system of linear algebraic equations shown in step 4 of algorithm 5.1. This utilization of the least square method ensures that the pole-placement algorithm would give the solution with the minimum norm at each iterative step. Furthermore, experience shows that the total feedback matrix has small norm, too.

In the next section, several examples are presented to illustrate the method.
5.4 Examples

5.4.1 Example 5.1

In this example, the inner-loop lateral axis design problem for a flight control as given in [89;90] will be considered. The system matrices are given by:

\[
A = \begin{bmatrix}
-0.746 & 0.387 & -12.9 & 6.05 & 0.952 & 0 \\
0.024 & -0.174 & 0.4 & -0.416 & -1.76 & 0 \\
0.006 & -0.999 & -0.058 & -0.0012 & 0.0092 & 0.0369 \\
0 & 0 & 0 & -5 & 0 & 0 \\
0 & 0 & 0 & 0 & -10 & 0 \\
1 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} \quad B = \begin{bmatrix}
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0
\end{bmatrix} \quad (5.4.1)
\]

\[
C = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix} \quad (5.4.2)
\]

The states of the system are:
\[
x = \begin{bmatrix}
\rho_s \\
r_s \\
\beta \\
\delta_a \\
\delta_r \\
\phi
\end{bmatrix}
\]

whereas, the control are:

\[
u = \begin{bmatrix}
\delta_{rc} \\
\delta_{ac}
\end{bmatrix}
\]

The desired poles are:

\[
\begin{align*}
\lambda_1 &= -200 & \text{Actuator} \\
\lambda_2 &= -100 & \text{Actuator} \\
\lambda_3 &= -4 & \text{Roll subsidence} \\
\lambda_4 &= -1.77 + j1.77 & \text{Dutch roll} \\
\lambda_5 &= -1.77 - j1.77 & \text{Dutch roll} \\
\lambda_6 &= -.005 & \text{Spiral mode}
\end{align*}
\]

The first scheme of shifting is chosen where the matrices \(E_s\) and \(E_d\) of Eq. (5.3.1) are the following:
In this example, there is no need to choose $E$, because the pair of the complex conjugate poles are to be shifted to another pair of complex poles.

The example has been solved in two ways, first, with the stabilizing procedure and second without it. In both cases, $\Delta \alpha = 0.1$ and the iterative algorithm goes to the next step when $\|\Delta K\| = 0.001$ or $i_m = 15$. When the algorithm is not utilizing the stability procedure, the following results are obtained:


$$\|K\| = 40.40$$

whereas, when the stabilizing procedure is utilized, the following results are obtained:


$$\|K\| = 38.83$$

Both of the above results assign the exact desired eigenvalues (i.e. $\Lambda(A - BK) = \Lambda^e$). Also, we note that both methods give feedback matrices with ap-
proximately equal norms, whereas the feedback matrix reported in [89] has a norm of 43.39. The high effort needed in this example reflects the requirement to shift two poles to \(-100\) and \(-200\).

**5.4.2 Example 5.2**

Consider the 5th-order system with three inputs and three outputs which is proposed in [57]. The system matrices are:

\[
A = \begin{bmatrix}
2 & 1.25 & -2.25 & -3.5 & 1 \\
-5 & -1 & 3 & 0 & -5 \\
-5 & 3 & -1 & 0 & -5 \\
0 & -0.75 & 0.75 & -2.5 & 0 \\
-5 & -1.25 & 2.25 & 3.5 & -4
\end{bmatrix} \quad ; \quad B = \begin{bmatrix}
1 & -1 & 1 \\
0 & 2 & 2 \\
2 & 2 & 0 \\
-1 & 0 & 2 \\
0 & 2 & 0
\end{bmatrix}
\]

\[ (5.4.11) \]

\[
C = \begin{bmatrix}
0 & 1 & 1 & 0 & -1 \\
1 & 0 & 0 & 1 & 1 \\
1 & 1 & -1 & 0 & 1
\end{bmatrix}
\]

\[ (5.4.12) \]

The open-loop poles are:

\[
\Lambda = \{1, 2, -2.5, -3, -4\}
\]

\[ (5.4.13) \]

whereas, the desired set of eigenvalues is:

\[
\Lambda^d = \{-1, -2, -5, -1+j1, -1-j1\}
\]

\[ (5.4.14) \]

When the second shifting scheme as shown in Eq. (5.3.2) is used with \(\Delta \alpha = 0.1\), the obtained output feedback is:
\[
K = \begin{bmatrix}
-1.494 & .307 & -.270 \\
3.466 & .755 & -.149 \\
-1.904 & 1.023 & -.193
\end{bmatrix}
\] (5.4.15)

with a norm of:

\[
\|K\| = 4.44
\] (5.4.16)

However, the original iterative method of section 5.2 gives the following results:

\[
K = \begin{bmatrix}
1.872 & .301 & -.389 \\
-5.166 & 1.347 & .569 \\
2.056 & -1.380 & .270
\end{bmatrix}
\] (5.4.17)

with a norm of:

\[
\|K\| = 6.23
\] (5.4.18)

The advantage of the extended iterative method over the original algorithm is obvious by comparing Eqs. (5.4.16) and (5.4.18). It should be noted that the number of inputs and outputs are sufficient in this case (i.e. \( m + p - 1 \geq n \)); therefore, the chosen paths for the poles shifting are feasible and the algorithm gives a feedback matrix with minimum effort. However, when the condition \( m + p - 1 \geq n \) is not satisfied, the extended algorithm may not give the minimum effort because in a certain iterative step, the chosen path is not feasible. For example, when the output feedback is chosen to be:
The extended iterative algorithm using the second shifting scheme will give the following results:

\[
C = \begin{bmatrix}
0 & 1 & 1 & 0 & -1 \\
1 & 0 & 0 & 1 & 1 \\
\end{bmatrix}
\]  

(5.4.19)

whereas, the original iterative algorithm will give the following results:

\[
K = \begin{bmatrix}
-5.756 & 3.989 \\
12.679 & -9.091 \\
-8.711 & 5.388 \\
\end{bmatrix}
\]  

(5.4.20)

Both of these output feedback matrices place the system eigenvalues at the desired locations exactly. However, the first method encountered a relatively large effort because the shifting path was not feasible.

5.4.3 Example 5.3

Consider the following 5'th order system with two inputs and two outputs. The system matrices are:
$$A = \begin{bmatrix} -1.5 & 1 & 0 & 0 & 0 \\ 0 & -1.5 & 0 & 0 & 0 \\ 0 & 0 & -3.5 & 1 & 0 \\ 0 & 0 & 0 & -3.5 & 0 \\ 0 & 0 & 0 & 0 & -5 \end{bmatrix} ; \quad B = \begin{bmatrix} 1 \\ -0.7 \\ 1.5 \\ 1 \\ 2.5 \end{bmatrix} \quad (5.4.22)$$

$$C = \begin{bmatrix} 1 & .7 & -.9 & 1 & 2 \\ .8 & -1 & .7 & .5 & 1 \end{bmatrix} \quad (5.4.23)$$

The open-loop poles are:

$$\Lambda = \{-1.5, -1.5, -3.5, -3.5, -5\} \quad (5.4.24)$$

The desired closed-loop poles are:

$$\Lambda^d = \{-2.182 \mp j.657, -4.264, -7.038, -22.722\} \quad (5.4.25)$$

Here, the first shifting scheme of Eq. (5.3.1) is used with $\Delta \alpha = 0.1$ and the following values for $E_o$ and $E_d$:

$$E_o = \begin{bmatrix} -1.5 & 0 \\ -1.5 & 0 \\ -3.5 & 0 \\ -3.5 & 0 \\ -5 & 0 \end{bmatrix} ; \quad E_d = \begin{bmatrix} -2.182 & .657 \\ -2.182 & -0 \end{bmatrix} \quad (5.4.26)$$

The extended iterative algorithm gives the following output feedback matrix:

$$K = \begin{bmatrix} -1.311 & 1.321 \\ 3.205 & -1.046 \end{bmatrix} \quad (5.4.27)$$
with the following norm:

$$\|K\| = 3.850$$ \hspace{1cm} (5.4.28)

This feedback matrix places the system poles at the following locations:

$$\Lambda(A - BK) = \{-2.154 \pm j.616, -3.836, -7.066, -21.176\}$$ \hspace{1cm} (5.4.29)

It should be noted that although the condition $mp \geq n$ is not satisfied in this example, the algorithm renders to an approximate solution which is very close to the desired one.
Chapter 6

Summary and Conclusions

6.1 Work in Retrospect

This dissertation addresses the problem of the pole-placement with minimum effort for linear multivariable systems. Both state and output feedback designs are considered. The work starts with a review of some canonical forms for linear multivariable systems. It also shows that Luenberger’s canonical forms are usually not unique. Specifically, it demonstrates that the selection process of the $m$ row vectors needed to construct the transformation matrix for Luenberger’s canonical forms is not unique. Then, other canonical forms, which can be considered as a modified version of Luenberger’s canonical forms, are also reviewed.

The proposed procedure to solve the problem starts by obtaining a direct transformation of the system matrices into the modified canonical forms. Depending on
the structural properties of the system, two realizations have been identified and the algorithms to obtain these forms have also been derived. The first realization can be obtained when the first $n$ columns of the controllability matrix are linearly independent, whereas the second realization is feasible when these $n$ columns are linearly dependent. The structures of these realizations are completely characterized by a set of integers called the Kronecker indices. In both cases, the transformation matrix has some degrees of freedom by tuning a certain scalar, or a set of scalars, within the matrix. These degrees of freedom are utilized in the solution to reduce further the norm of the state feedback matrix.

Then, a solution to the pole-placement problem with minimum effort is obtained. The solution was made possible by using the pole-placement condition and by minimizing a certain functional subject to a set of specified constraints. The form of the functional reflects our objective that the desired matrix should be as close as possible to the transformed system matrix so that a small feedback norm could be obtained, whereas the constraint ensures that the desired matrix has the set of required eigenvalues. The freedom within the method is utilized by tuning a scalar within the transformation matrix so that a compromise, between the matrices constituting the feedback matrix, can be achieved. Strictly speaking, the proposed method does not necessarily give the absolute minimum because we are only minimizing one matrix, from the three matrices constituting the feedback matrix, while compromising the other two. Nevertheless, the method produces a state feedback matrix with a norm which is very close to the absolute minimum. However, the simplicity and the reduced amount of computation in the method make it attractive. Moreover, the method is applicable to higher order systems while the other methods which give absolute minimum are not. Several practical examples have been solved using the proposed
method, and a comparison with the other methods has been made. This comparison has shown the superiority and the validity of the proposed method. The previous results have been published in [91; 92].

On the other hand, a non-canonical form approach to the problem is investigated. In this approach, it was shown that to solve the pole-placement problem, it is not necessary to transform the system matrices to a certain canonical form. However, the only requirement is that the transformed input system matrix should have zero rows in the first \( n - m \) rows. The transformation matrix, in this case, has larger degrees of freedom, which can be utilized to solve the formulated problem. Moreover, a new pole-placement method based on the non-canonical approach is proposed. In this method, the input system matrix is transformed into the above form, while a desired diagonal matrix is transformed into a form having, as its first \( n - m \) rows, the corresponding rows of the transformed system matrix. The latter desired transformation matrix is obtained by solving the Lyapunov matrix equation. Finally, the desired state feedback matrix can be determined using the well known pole-placement condition. The results of using the non-canonical approach will be published in [93].

As another part of the dissertation, an iterative algorithm for output feedback pole-placement is reviewed. This algorithm has the advantage of treating the feedback matrix as a vector which can be obtained by the least square method. The latter feature is significant because our objective is to find a minimum norm output feedback matrix for the pole-placement problem. Then, this algorithm is extended to attain the required objective. The extension has been accomplished by applying the successive pole shifting method. The use of the successive shifting procedure helps to reduce the norm of the output feedback because small displacement of the poles re-
quires only small effort. Moreover, experience has shown that the total feedback matrix has usually a low norm, too. Two schemes for the pole shifting are proposed. The first of these schemes is to successively shift the poles through straight paths starting from the open loop poles and ending at the desired poles, whereas the second scheme shifts the poles according to a successive change of their characteristic polynomial coefficients. Experience with this extended iterative algorithm shows that it is most effective when the system has a sufficient number of inputs and outputs because the shifting paths for this case are feasible. On the other hand, when these paths are not feasible, the designer may have to choose between the original and the new method depending on their performance. However, even for the case when \( mp < n \), the proposed method gives an approximate solution which minimizes the quantity \( \|f_0 - f(K)\| \).

### 6.2 Directions for Further Research

As in any other theoretical investigation, during the course of this research, many issues have come to be of significant value that deserve further attention. The following is a list of possible topics suggested for further research:

- The freedom, which the transformation matrix has in the proposed canonical forms, has been utilized to reduce the norm of the feedback matrix in the pole-placement problem. An interesting research topic is to use these degrees of freedom in other multivariable problems such as identification, minimal input-output realization, observer design, and other problems.
• The transformation matrix, in the proposed method, has certain degrees of freedom by tuning a scalar or a set of scalars within this matrix. However, only the first type of tuning has been implemented in this dissertation. Therefore, another research topic is to find a way to tune the set of scalars so as to reduce the norm of the feedback matrix.

• It has been shown recently that a given MIMO system may be represented by several equivalent representations called overlapping or pseudo-canonical forms. The structure of a Pseudo-canonical form is determined by a set of pseudo-controllability indices. Therefore, a possible research topic is to apply the idea of pseudo-controllability to the proposed problem so as to identify the structure which gives a feedback matrix with the minimum norm.

• The non-canonical approach for pole-placement has several quantities which can be chosen freely. In particular, the matrices $T$ and $T_m$ are free to be chosen with mild restrictions only. Another research topic is to utilize these freedoms to obtain the minimum norm feedback matrix.

• In the extended iterative algorithm for output feedback, it has been shown that the method is most effective when the poles are shifted in feasible paths. Therefore, another topic is to search for these paths for the case when the number of inputs and outputs is not sufficient.
Appendix A

Proof of Algorithm 4.1

Observing the structure of $\tilde{A}$ and $\tilde{B}$ in the first realization shown in Eq. (4.1.5), and since the following relations hold:

$$TA = \tilde{A} T$$  \hspace{1cm} (A.1)

$$TB = \tilde{B}$$  \hspace{1cm} (A.2)

one can conclude that the first $n - m$ row vectors of $T$ are in the null space of $B$. However, Eq. (A.1) leads to the following relations:

$$t^1 A = t^{m+1}$$
$$t^2 A = t^{m+2}$$
$$\vdots$$
$$t^{n-m} A = t^n$$  \hspace{1cm} (A.3)
where \( t' \) is the \( l' \)th row vector of \( T \). The \((m + 1)' \)th relation of Eq. (A.3) can be written as:

\[
t^{2m+1} = t^{m+1} A
\]

(A.4)

Substituting the first relation of Eq. (A.3) in Eq. (A.4) we get:

\[
t^{2m+1} = t^1 A^2
\]

Repeating the previous process for the relations \((m + 2)\) to \((n - m)\) in Eq. (A.3), one can conclude that the row vectors \( t^{m+1} \) to \( t^n \) can be expressed in terms of \( A \) and the row vectors \( t' \) to \( t^{m} \) only. Therefore, Eq. (A.2) can be rewritten in terms of the matrices \( A \) and \( B \) and the row vectors \( t' \) to \( t^{m} \). However, with more observations one can write Eq. (A.2) as:

\[
t^1 [B | AB | ... A^{p-1} B] = 0
\]

\[
: \nonumber
\]

\[
t^k [B | AB | ... A^{p-1} B] = 0
\]

\[
(A.5)
\]

and

\[
t^{k+1} [B | AB | ... A^{p-2} B] = 0
\]

\[
: \nonumber
\]

\[
t^m [B | AB | ... A^{p-2} B] = 0
\]

(A.6)

Each relation of Eq. (A.5) represents \( p \) rows of \( \tilde{B} \), while each relation of Eq. (A.6) represents \( p - 1 \) rows. However, since the total number of zero rows in \( \tilde{B} \) are \( n - m \), both Eqs. (A.5) and (A.6) represent \( n - m \) rows, i.e.

\[
k p + (m - k)(p - 1) = n - m
\]

(A.7)
Eq. (A.7) can be reduced to the following form:

\[ mp + k = n \]  \hspace{1cm} (A.8)

From Eqs. (A.5), (A.6) and (A.8), \( k \) takes the following range of values:

\[ 0 \leq k < m - 1 \]  \hspace{1cm} (A.9)

Then, from Eqs. (A.8) and (A.9), the value of \( p \) can be found by:

\[ p = \binom{n}{m} \]  \hspace{1cm} (A.10)

where the symbol \( \binom{a}{b} \) denotes the greatest integer value less than or equal to the ratio \( a/b \).

The previous discussion proves Eqs. (4.1.6) and (4.1.8) of Algorithm 4.1. The row vectors \( t^{m+1} \) to \( t^n \) can be found using Eq. (A.3) which can be rewritten as shown in Eq. (4.1.7). The integer \( l \) in Eq. (4.1.7) reflects the number of groups of rows in Eq. (A.3). Each of these groups contains \( m \) rows except the last group which may contain less than \( m \) rows. This completes the proof.
Appendix B

Proof of Lemma 4.2.1

In this lemma, the objective is to find the nonzero determinants from all possible $k \times k$ determinants $\Delta(i_1, \ldots, i_k)$ formed from the rows $i = i_1, \ldots, i_k$ and the columns $j = i_1, \ldots, i_k$ of the matrix $A'$. If any determinant has a zero row, it will be discarded. However, $A'$ has a special structure as shown in Eq. (4.1.5) and Eq. (4.1.11). In this structure, there are $m$ rows with nonzero and nonunity elements, whereas the remaining $n - m$ rows are the last $n - m$ rows of a unity matrix $I_n$. The location of the unity for the first to the $(s_1 - 1)$'th rows is located at $m + i$, whereas the location of the unity for $(s_1 + 1)$'th to $(s_2 - 1)$'th rows is located at $m + i - 1$. In general, if we denote $S = \{s_1, \ldots, s_m\}$ and $I = \{i_1, \ldots, i_k\}$, the location of the unity will be at $i + m - l$ where $l$ is the number of elements in $S$ which is less than $i$. Therefore, to find the nonzero determinants, we have to check only the elements of $I$ which is not in $S$ to have the unity included in the determinant. This discussion concludes the proof of the lemma.
C.1 Program Listing

The following is the listing of the BASIC program which calculates the coefficients of the characteristic polynomial for the desired matrix in the modified canonical form. This program utilizes Lemma 4.2.1 to select only the nonzero determinants.
10 REM xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
20 REM xx
30 REM xx THIS PROGRAM COMPUTES THE COEFFICIENTS OF xx
40 REM xx THE CHARACTERISTICS POLYNOMIAL FOR THE xx
50 REM xx DESIRED MATRIX IN MODIFIED CANONICAL FORM xx
60 REM xx
70 REM xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
80 REM
90 REM
100 REM
110 CLS
120 INPUT "Enter the order of the system ";N
130 PRINT
140 INPUT "Enter the number of inputs ";M
150 PRINT
160 DIM C(N,130,N),CT(N,50,N),S(M),KC(N),IC(10),CONI(M),
170 INPUT "Are the first N columns of the controllability
180 IF LEFT$(T,1)="N" OR LEFT$(T,1)="n" THEN GOTO 230
190 FOR J1=1 TO M
200 S(J1)=N-M+J1
210 NEXT J1
220 GOTO 340
230 PRINT PRINT PRINT PRINT PRINT
240 INPUT "Enter the controllability indices in an
250 FOR K1=1 TO H
260 PRINT " Index";K1;"is ";
270 INPUT CONI(K1)
280 NEXT K1
290 S(0)=0
300 CONI(0)=1
310 FOR J1=1 TO M
320 S(J1)=S(J1-1)+(CONI(J1)-CONI(J1-1))X(M-J1+1)+1
330 NEXT J1
340 PRINT PRINT PRINT PRINT PRINT PRINT
350 H="#CALCULATION OF POLYNOMIAL COEFFICIENTS#
360 LN=LEN(H#)
370 LPRINT TAB((60-LN)/2+.5);H#
380 LPRINT:LPRINT:LPRINT:
390 REM
400 REM xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
410 REM
420 REM xxxxx GENERATE ALL DETERMINANTS xxxxx
430 REM
440 I1=0
450 I2=0
460 I3=0
470 I4=0
480 I5=0
490 I6=0
500 I7=0
510 FOR KK=2 TO N-1
520 J1=0
530 J2=0
540 J3=0
550 J4=0

Figure 6. The Listing of the BASIC Program.
FOR I=1 TO N-KK+1
J1=I
FOR J=1 TO N-KK+2
IF J1+J>N-KK+2 THEN GOTO 1440
J2=J1+J
IF KK<>2 THEN GOTO 700
I1=I1+1
C(2,I1,1)=J1
C(2,I1,2)=J2
IF KK=2 THEN GOTO 1430
FOR K=1 TO N-KK+3
IF J2+K>N-KK+3 THEN GOTO 1430
J3=J2+K
IF KK<>3 THEN GOTO 790
I2=I2+1
C(3,I2,1)=J1
C(3,I2,2)=J2
C(3,I2,3)=J3
IF KK=3 THEN GOTO 1420
FOR N1=1 TO N-KK+4
IF J3+N1>N-KK+4 THEN GOTO 1420
J4=J3+N1
IF KK<>4 THEN GOTO 890
I3=I3+1
C(4,I3,1)=J1
C(4,I3,2)=J2
C(4,I3,3)=J3
C(4,I3,4)=J4
IF KK=4 THEN GOTO 1410
FOR N2=1 TO N-KK+5
IF J4+N2>N-KK+5 THEN GOTO 1410
J5=J4+N2
IF KK<>5 THEN GOTO 1000
I4=I4+1
C(5,I4,1)=J1
C(5,I4,2)=J2
C(5,I4,3)=J3
C(5,I4,4)=J4
C(5,I4,5)=J5
IF KK=5 THEN GOTO 1400
FOR N3=1 TO N-KK+6
IF J5+N3>N-KK+6 THEN GOTO 1400
J6=J5+N3
IF KK<>6 THEN GOTO 1120
I5=I5+1
C(6,I5,1)=J1
C(6,I5,2)=J2
C(6,I5,3)=J3
C(6,I5,4)=J4
C(6,I5,5)=J5
C(6,I5,6)=J6
1110 IF KK=6 THEN GOTO 1390
1120 FOR N4=1 TO N-KK+7
1130 IF J6+N4>N-KK+7 THEN GOTO 1390
1140 J7=J6+N4
1150 IF KK<>7 THEN GOTO 1250
1160 I6=I6+1
1170 C(7,I6,1)=J1
1180 C(7,I6,2)=J2
1190 C(7,I6,3)=J3
1200 C(7,I6,4)=J4
1210 C(7,I6,5)=J5
1220 C(7,I6,6)=J6
1230 C(7,I6,7)=J7
1240 IF KK=7 THEN GOTO 1380
1250 FOR N5=1 TO N-KK+8
1260 IF J7+N5> N-KK+8 THEN GOTO 1380
1270 J8=J7+N5
1280 I7=I7+1
1290 C(8,I7,1)=J1
1300 C(8,I7,2)=J2
1310 C(8,I7,3)=J3
1320 C(8,I7,4)=J4
1330 C(8,I7,5)=J5
1340 C(8,I7,6)=J6
1350 C(8,I7,7)=J7
1360 C(8,I7,8)=J8
1370 NEXT N5
1380 NEXT N3
1390 NEXT N2
1400 NEXT N1
1410 NEXT K
1420 NEXT I
1430 NEXT J
1440 NEXT I
1450 NEXT KK
1460 IC(2)=I1
1470 IC(3)=I2
1480 IC(4)=I3
1490 IC(5)=I4
1500 IC(6)=I5
1510 IC(7)=I6
1520 IC(8)=I7
1530 REM
1540 REM
1550 REM
1560 REM
1570 REM
1580 REM
1590 FOR I=2 TO N-1
1600 FOR J=1 TO IC(I)
1610 FOR K=1 TO I
1620 L=0
1630 FOR KI=1 TO M
1640 IF C(I,J,K)=S(KI) THEN GOTO 1730
1650 NEXT KI
1660 NEXT K
1670 NEXT J
1680 NEXT I
1690 IF KK=7 THEN GOTO 1380
1700 FOR N5=1 TO N-KK+8
1710 IF J7+N5> N-KK+8 THEN GOTO 1380
1720 J8=J7+N5
1730 I7=I7+1
1740 C(8,I7,1)=J1
1750 C(8,I7,2)=J2
1760 C(8,I7,3)=J3
1770 C(8,I7,4)=J4
1780 C(8,I7,5)=J5
1790 C(8,I7,6)=J6
1800 C(8,I7,7)=J7
1810 C(8,I7,8)=J8
1820 NEXT N5
1830 NEXT N3
1840 NEXT N2
1850 NEXT N1
1860 NEXT K
1870 NEXT I
1880 NEXT I
1890 NEXT KK
1900 IC(2)=I1
1910 IC(3)=I2
1920 IC(4)=I3
1930 IC(5)=I4
1940 IC(6)=I5
1950 IC(7)=I6
1960 IC(8)=I7
1970 REM
1980 REM
1990 REM
2000 REM
2010 REM
2020 REM
2030 FOR I=2 TO N-1
2040 FOR J=1 TO IC(I)
2050 FOR K=1 TO I
2060 L=0
2070 FOR KI=1 TO M
2080 IF C(I,J,K)=S(KI) THEN GOTO 1730
2090 NEXT KI
2100 NEXT K
2110 NEXT J
2120 NEXT I
2130 IF KK=7 THEN GOTO 1380
2140 FOR N5=1 TO N-KK+8
2150 IF J7+N5> N-KK+8 THEN GOTO 1380
2160 J8=J7+N5
2170 I7=I7+1
2180 C(8,I7,1)=J1
2190 C(8,I7,2)=J2
2200 C(8,I7,3)=J3
2210 C(8,I7,4)=J4
2220 C(8,I7,5)=J5
2230 C(8,I7,6)=J6
2240 C(8,I7,7)=J7
2250 C(8,I7,8)=J8
2260 NEXT N5
2270 NEXT N3
2280 NEXT N2
2290 NEXT N1
2300 NEXT K
2310 NEXT I
2320 NEXT I
2330 NEXT KK
2340 IC(2)=I1
2350 IC(3)=I2
2360 IC(4)=I3
2370 IC(5)=I4
2380 IC(6)=I5
2390 IC(7)=I6
2400 IC(8)=I7
2410 REM
2420 REM
2430 REM
2440 REM
2450 REM
2460 REM
2470 REM
2480 REM
2490 REM
2500 REM
2510 REM
2520 FOR I=2 TO N-1
2530 FOR J=1 TO IC(I)
2540 FOR K=1 TO I
2550 L=0
2560 FOR KI=1 TO M
2570 IF C(I,J,K)=S(KI) THEN GOTO 1730
2580 NEXT KI
2590 NEXT K
2600 NEXT J
2610 NEXT I
1660 FOR KK=1 TO M
1670 IF C(I,J,K)>S(KK) THEN L=L+1
1680 NEXT KK
1690 FOR KJ=1 TO I
1700 IF C(I,J,K)+M-L=C(I,J,KJ) THEN GOTO 1730
1710 NEXT KJ
1720 GOTO 1780
1730 NEXT K
1740 KC(I)=KC(I)+1
1750 FOR KI=1 TO I
1760 CT(I,KC(I),KI)=C(I,J,KI)
1770 NEXT KI
1780 NEXT J
1790 NEXT I
1800 KC(N)=1
1810 FOR KI=1 TO N
1820 CT(N,KI)=K1
1830 NEXT KI
1840 REM
1850 REM ********** COMPUTE EACH DETERMINANT **********
1860 REM
1870 REM
1880 REM ********** COMPUTE EACH DETERMINANT **********
1890 REM
1900 MI=0;NI=0
1910 FOR I=1 TO N
1920 LPRINT; LPRINT
1930 LPRINT "f(";I,"b)" ;=
1940 LPRINT
1950 IF I=1 THEN GOTO 3490
1960 FOR J=1 TO KC(I)
1970 FOR K=1 TO I
1980 CP(K)=CT(I,J,K)
1990 NEXT K
2000 IF I=2 THEN GOTO 3380
2010 FOR L=1 TO 9
2020 IF I<9 THEN GOTO 2120
2030 FOR NN6=1 TO I
2040 C9(9)=CT(I,J,NN6)
2050 CP(9)=C9(9)
2060 NEXT NN6
2070 CP(9)=C9(9)
2080 CP(9)=C9(9)
2090 FOR K9=1 TO I
2100 C9(K9)=CP(K9)
2110 NEXT K9
2120 FOR L9=1 TO 8
2130 IF I>8 THEN GOTO 2250
2140 IF I<8 THEN GOTO 2340
2150 FOR NN5=1 TO I
2160 C8(9)=CT(I,J,NN5)
2170 CP(9)=C8(9)
2180 NEXT NN5
2190 CP(9)=C8(9)
2200 CP(8)=C8(8)
2210 FOR K8=1 TO I
2220 C8(K8)=CP(K8)
2230 NEXT K8
2240 GOTO 2340
2250 FOR N8=1 TO I
2260 C8(N8)=C9(N8)
2270 CP(N8)=C8(N8)
2280 NEXT N8
2290 CP(8)=C8(8)
2300 CP(L8)=C8(L8)
2310 FOR K8=1 TO I
2320 C8(K8)=CP(K8)
2330 NEXT K8
2340 FOR L7=1 TO 7
2350 IF I>7 THEN GOTO 2470
2360 IF I<7 THEN GOTO 2560
2370 FOR NN4=1 TO I
2380 C7(NN4)=CT(I,J,NN4)
2390 CP(NN4)=C7(NN4)
2400 NEXT NN4
2410 CP(7)=C7(7)
2420 CP(L7)=C7(L7)
2430 FOR K7=1 TO I
2440 C7(K7)=CP(K7)
2450 NEXT K7
2460 GOTO 2560
2470 FOR N7=1 TO I
2480 C7(N7)=C8(N7)
2490 CP(N7)=C7(N7)
2500 NEXT N7
2510 CP(7)=C7(7)
2520 CP(L7)=C7(L7)
2530 FOR K7=1 TO I
2540 C7(K7)=CP(K7)
2550 NEXT K7
2560 FOR L6=1 TO 6
2570 IF I>6 THEN GOTO 2690
2580 IF I<6 THEN GOTO 2780
2590 FOR NN3=1 TO I
2600 C6(NN3)=CT(I,J,NN3)
2610 CP(NN3)=C6(NN3)
2620 NEXT NN3
2630 CP(6)=C6(6)
2640 CP(L6)=C6(L6)
2650 FOR K6=1 TO I
2660 C6(K6)=CP(K6)
2670 NEXT K6
2680 GOTO 2780
2690 FOR N6=1 TO I
2700 C6(N6)=C7(N6)
2710 CP(N6)=C6(N6)
2720 NEXT N6
2730 CP(6)=C6(6)
2740 CP(L6)=C6(L6)
2750 FOR K6=1 TO I
2760 C6(K6)=CP(K6)
2770 NEXT K6
2780 FOR L5=1 TO 5
2790 IF I>5 THEN GOTO 2910
2800 IF I<5 THEN GOTO 3000
2810 FOR NN2=1 TO I
2820 C5(NN2)=CT(I,J,NN2)
2830 CP(NN2)=C5(NN2)
2840 NEXT NN2
2850 CP(5)=C5(L5)
2860 CP(L5)=C5(5)
2870 FOR K5=1 TO I
2880 C5(K5)=CP(K5)
2890 NEXT K5
2900 GOTO 3000
2910 FOR N5=1 TO I
2920 C5(N5)=C6(N5)
2930 CP(N5)=C5(N5)
2940 NEXT N5
2950 CP(5)=C5(L5)
2960 CP(L5)=C5(5)
2970 FOR K5=1 TO I
2980 C5(K5)=CP(K5)
2990 NEXT K5
3000 FOR L4=1 TO 4
3010 IF I<5 THEN GOTO 3180
3020 IF I>4 THEN GOTO 3100
3030 FOR NN1=1 TO I
3040 C4(NN1)=CT(I,J,NN1)
3050 CP(NN1)=C4(NN1)
3060 NEXT NN1
3070 CP(4)=C4(L4)
3080 CP(L4)=C4(4)
3090 GOTO 3180
3100 FOR N4=1 TO I
3110 C4(N4)=C5(N4)
3120 NEXT N4
3130 FOR K4=1 TO I
3140 CP(K4)=C4(K4)
3150 NEXT K4
3160 CP(4)=C4(L4)
3170 CP(L4)=C4(4)
3180 FOR L1=1 TO 6
3190 MI=MI+1
3200 NI=NI+1
3210 SWAP CP(MI),CP(NI)
3220 GOSUB 3650
3230 IF NI=3 THEN MI=0; NI=0
3240 NEXT L1
3250 IF I=3 THEN GOTO 3410
3260 NEXT L4
3270 IF I=4 THEN GOTO 3410
3280 NEXT L5
3290 IF I=5 THEN GOTO 3410
3300 NEXT L6

BASIC Program
3310 IF I=6 THEN GOTO 3410
3320 NEXT L7
3330 IF I=7 THEN GOTO 3410
3340 NEXT L8
3350 IF I=8 THEN GOTO 3410
3360 NEXT L9
3370 GOTO 3410
3380 GOSUB 3650
3390 SNAP CP(1),CP(2)
3400 GOSUB 3650
3410 NEXT J
3420 GOTO 3520
3430 REM
3440 REM
3450 REM XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
3460 REM
3470 REM XXXXXXXXXXX COMUTE f(1) XXXXXXXXXXX
3480 REM
3490 FOR K1=1 TO M
3500 LPRINT "(-1)^a;S(K1);S(K1)
3510 NEXT K1
3520 NEXT I
3530 LPRINT;LPRINT;LPRINT
3540 LPRINT" The coefficients f(1),f(2),... are : "
3550 LPRINT
3560 LPRINT" det(sI-Ad)= s^n + f(1) s^(n-1) +";
3570 LPRINT" ... + f(n-1) s + f(n) "
3580 END
3590 REM
3600 REM
3610 REM XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
3620 REM
3630 REM XXXXXXXXXXX COMPUTE THE SIGN OF EACH TERM XXXXXXXXXXX
3640 REM
3650 CPDT=1
3660 FOR J1=1 TO I-1
3670 FOR JJ=1 TO I-J1
3680 CPD=CP(J1+JJ)-CP(J1)
3690 CPDT=CPDT*CPD
3700 NEXT JJ
3710 NEXT J1
3720 TS=(((-1)^I)*SGN(CPDT)
3730 REM
3740 REM
3750 REM XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
3760 REM
3770 REM XXXXXXXXXXX COMPUTE EACH TERM XXXXXXXXXXX
3780 REM
3790 REM
3800 FOR K=1 TO I
3810 L=0
3820 FOR K1=1 TO M
3830 IF CT(I,J,K)=S(K1) THEN GOTO 3900
3840 NEXT K1
3850 FOR KK=1 TO M

BASIC Program 129
3860 IF CT(I,J,K)>S(KK) THEN L=L+1
3870 NEXT KK
3880 IF CP(K)=CT(I,J,K)+M·L THEN GOTO 3930 ELSE RETURN
3890 GOTO 3930
3900 COUNT=COUNT+1
3910 CFN(1,COUNT)=CT(I,J,K)
3920 CFN(2,COUNT)=CP(K)
3930 NEXT K
3940 LPRINT "(";TS;")":
3950 FOR JJ=1 TO COUNT
3960 LPRINT "•";CFN(1,JJ);CFN(2,JJ);
3970 NEXT JJ
3980 LPRINT
3990 RETURN
C.2 Program Output

The following is a sample of the program output for the two cases:

1. When the first $n$ columns of the controllability matrix are linearly independent.

2. When these $n$ columns are linearly dependent.
Enter the order of the system 1 6
Enter the number of inputs 1 2
Are the first N columns of the controllability matrix linearly independent (Y or N)? Y

CALCULATION OF POLYNOMIAL COEFFICIENTS

\[
\begin{align*}
f(1) : \\
&(-1)a \ 5 \ 5 \\
&(-1)a \ 6 \ 6 \\

f(2) : \\
&(-1)a \ 5 \ 3 \\
&(-1)a \ 6 \ 4 \\
&(1)a \ 5 \ 5 \ a \ 6 \ 6 \\
&(-1)a \ 5 \ 6 \ a \ 6 \ 5 \\

f(3) : \\
&(-1)a \ 5 \ 1 \\
&(-1)a \ 6 \ 2 \\
&(1)a \ 5 \ 3 \ a \ 6 \ 6 \\
&(-1)a \ 5 \ 6 \ a \ 6 \ 3 \\
&(1)a \ 5 \ 5 \ a \ 6 \ 4 \\
&(-1)a \ 5 \ 4 \ a \ 6 \ 5 \\

f(4) : \\
&(-1)a \ 5 \ 6 \ a \ 6 \ 1 \\
&(1)a \ 5 \ 1 \ a \ 6 \ 6 \\
&(-1)a \ 5 \ 5 \ a \ 6 \ 2 \\
&(-1)a \ 5 \ 2 \ a \ 6 \ 5 \\
&(-1)a \ 5 \ 4 \ a \ 6 \ 3 \\
&(1)a \ 5 \ 3 \ a \ 6 \ 4 \\
\end{align*}
\]

Figure 7. A Sample of the Program Output.
f(5):
(-1)a 5 4 a 6 1
( 1)a 5 1 a 6 4
( 1)a 5 3 a 6 2
(-1)a 5 2 a 6 3

f(6):
(-1)a 5 2 a 6 1
( 1)a 5 1 a 6 2

The coefficients f(1), f(2), ..., are:
det(sI-Ad)=s^n + f(1)s^{n-1} + ... + f(n-1)s + f(n)
Enter the order of the system? 7
Enter the number of inputs? 3
Are the first N columns of the controllability matrix linearly independent (Y or N)? N

Enter the controllability indices in an ascending order:

Index 1 is Y 1
Index 2 is Y 1
Index 3 is Y 5

CALCULATION OF POLYNOMIAL COEFFICIENTS

\[ f(1) = (-1)^1 a_1 1 \]
\[ (-1)^2 a_2 2 \]
\[ (-1)^3 a_7 7 \]

\[ f(2) = (-1)^1 a_1 1 a_2 2 a_7 7 \]
\[ (-1)^2 a_2 2 a_7 7 \]
\[ (-1)^3 a_7 7 \]

\[ f(3) = (-1)^1 a_1 1 a_2 2 a_7 7 \]
\[ (-1)^2 a_1 2 a_2 7 a_7 1 \]
\[ (-1)^3 a_1 7 a_2 2 a_7 1 \]
\[ (-1)^4 a_1 7 a_2 1 a_7 2 \]
\[ (-1)^5 a_1 1 a_2 7 a_7 2 \]
<table>
<thead>
<tr>
<th>(-1)</th>
<th>a</th>
<th>1</th>
<th>a</th>
<th>2</th>
<th>a</th>
<th>7</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>(-1)</td>
<td>a</td>
<td>1</td>
<td>6</td>
<td>a</td>
<td>7</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>(1)</td>
<td>a</td>
<td>1</td>
<td>a</td>
<td>7</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(-1)</td>
<td>a</td>
<td>2</td>
<td>6</td>
<td>a</td>
<td>7</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>(1)</td>
<td>a</td>
<td>2</td>
<td>2</td>
<td>a</td>
<td>7</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>(-1)</td>
<td>a</td>
<td>7</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\(f(4)\):

<table>
<thead>
<tr>
<th>(-1)</th>
<th>a</th>
<th>1</th>
<th>2</th>
<th>a</th>
<th>2</th>
<th>6</th>
<th>a</th>
<th>7</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>(-1)</td>
<td>a</td>
<td>1</td>
<td>6</td>
<td>a</td>
<td>2</td>
<td>2</td>
<td>a</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>(1)</td>
<td>a</td>
<td>1</td>
<td>6</td>
<td>a</td>
<td>2</td>
<td>1</td>
<td>a</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>(-1)</td>
<td>a</td>
<td>1</td>
<td>2</td>
<td>a</td>
<td>2</td>
<td>1</td>
<td>a</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>(-1)</td>
<td>a</td>
<td>1</td>
<td>a</td>
<td>2</td>
<td>2</td>
<td>a</td>
<td>7</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>(1)</td>
<td>a</td>
<td>1</td>
<td>a</td>
<td>7</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(-1)</td>
<td>a</td>
<td>2</td>
<td>5</td>
<td>a</td>
<td>7</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1)</td>
<td>a</td>
<td>2</td>
<td>2</td>
<td>a</td>
<td>7</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(-1)</td>
<td>a</td>
<td>7</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\(f(5)\):

<table>
<thead>
<tr>
<th>(-1)</th>
<th>a</th>
<th>1</th>
<th>2</th>
<th>a</th>
<th>2</th>
<th>5</th>
<th>a</th>
<th>7</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>(-1)</td>
<td>a</td>
<td>1</td>
<td>5</td>
<td>a</td>
<td>2</td>
<td>2</td>
<td>a</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>(1)</td>
<td>a</td>
<td>1</td>
<td>5</td>
<td>a</td>
<td>2</td>
<td>1</td>
<td>a</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>(-1)</td>
<td>a</td>
<td>1</td>
<td>2</td>
<td>a</td>
<td>2</td>
<td>1</td>
<td>a</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>(-1)</td>
<td>a</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>a</td>
<td>7</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1)</td>
<td>a</td>
<td>1</td>
<td>4</td>
<td>a</td>
<td>7</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(-1)</td>
<td>a</td>
<td>2</td>
<td>4</td>
<td>a</td>
<td>7</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1)</td>
<td>a</td>
<td>2</td>
<td>2</td>
<td>a</td>
<td>7</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(-1)</td>
<td>a</td>
<td>7</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\(f(6)\):

<table>
<thead>
<tr>
<th>(-1)</th>
<th>a</th>
<th>1</th>
<th>2</th>
<th>a</th>
<th>2</th>
<th>4</th>
<th>a</th>
<th>7</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>(-1)</td>
<td>a</td>
<td>1</td>
<td>4</td>
<td>a</td>
<td>2</td>
<td>2</td>
<td>a</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>(1)</td>
<td>a</td>
<td>1</td>
<td>4</td>
<td>a</td>
<td>2</td>
<td>1</td>
<td>a</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>(-1)</td>
<td>a</td>
<td>1</td>
<td>2</td>
<td>a</td>
<td>2</td>
<td>4</td>
<td>a</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>(-1)</td>
<td>a</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>a</td>
<td>7</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>(-1)</td>
<td>a</td>
<td>1</td>
<td>1</td>
<td>a</td>
<td>2</td>
<td>2</td>
<td>a</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>(1)</td>
<td>a</td>
<td>1</td>
<td>3</td>
<td>a</td>
<td>7</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1)</td>
<td>a</td>
<td>1</td>
<td>1</td>
<td>a</td>
<td>7</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(-1)</td>
<td>a</td>
<td>2</td>
<td>3</td>
<td>a</td>
<td>7</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1)</td>
<td>a</td>
<td>2</td>
<td>2</td>
<td>a</td>
<td>7</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\(f(7)\):

| \(-1\) | a | 1 | 2 | a | 2 | 3 | a | 7 | 1 |

BASIC Program
The coefficients $f(1), f(2), \ldots$ are:

$$\det(aI - A^T) = s^n + f(1)s^{n-1} + \ldots + f(n-1)s + f(n)$$
References


43. Munro, N. and Vardulakis, A.: Pole-shifting Using Output Feedback. *Int. J. 

pp.721-726.

45. Munro, N.: Further Results on Pole-shifting Using Output Feedback. *Int. J. 


47. Vardulakis, A. I.: A Sufficient Condition for n Specified Eigenvalues to be As-

48. Vardulakis, A. I.: Generalized Root Locus Assignment of All the Poles of a 


52. Moore, B. C.: On the Flexibility Offered by State Feedback in Multivariable 

pp.230-231.

54. Porter, B. and Bradshow, A.: Design of Linear Multivariable Continuous-time 

55. Fletcher, L. R.: An Intermediate Algorithm for Pole Placement by Output 
pp.1121-1136.


References 142


The two page vita has been removed from the scanned document. Page 1 of 2
The two page vita has been removed from the scanned document. Page 2 of 2