

NASA Contractor Report 3889

NASA-CR-3889 19850015489

**FOR REFERENCE**

**Design of Multivariable Feedback  
Control Systems Via Spectral  
Assignment Using Reduced-Order  
Models and Reduced-Order Observers**

Roland R. Mielke, Leonard J. Tung,  
and Preston Ivanhoe Carraway III

GRANT NSG-1650  
APRIL 1985

**LIBRARY COPY**

APR 24 1985

LANGLEY RESEARCH CENTER  
LIBRARY, NASA  
HAMPTON, VIRGINIA

**NASA**



NASA Contractor Report 3889

# Design of Multivariable Feedback Control Systems Via Spectral Assignment Using Reduced-Order Models and Reduced-Order Observers

Roland R. Mielke, Leonard J. Tung,  
and Preston Ivanhoe Carraway III

*Old Dominion University  
Norfolk, Virginia*

Prepared for  
Langley Research Center  
under Grant NSG-1650

**NASA**

National Aeronautics  
and Space Administration

Scientific and Technical  
Information Branch

1985



## TABLE OF CONTENTS

	Page
ABSTRACT.....	ii
LIST OF FIGURES.....	v
LIST OF SYMBOLS.....	viii
 Chapter	
1 INTRODUCTION.....	1
1.1 Motivation.....	1
1.2 Overview.....	3
2 SPECTRAL ASSIGNMENT PROCEDURE.....	5
2.1 System Eigenstructure and Time Response.....	5
2.2 Characterization of Freedom of Eigenvector Assignment.....	13
2.3 Eigenvector Assignment for Real Eigenvalues.....	15
2.4 Eigenvector Assignment for Complex Eigenvalues.....	17
2.5 Use of Eigenvector Freedom.....	24
2.6 Improvement of Initial Assignment by Gradient Search.....	26
3 SPECTRAL ASSIGNMENT USING REDUCED-ORDER MODELS AND REDUCED-ORDER OBSERVERS.....	29
3.1 Motivation for Using Reduced-Order Models and Reduced-Order Observers.....	29
3.2 Reduced-Order Model Formulation.....	30
3.3 Spectral Assignment Using Reduced-Order Models.....	32
3.4 Reduced-Order Observer Formulation.....	37
3.5 Effect of Reduced-Order Observer on Full System Eigenstructure.....	48
4 DESIGN PROCEDURE.....	56
4.1 Design Philosophy for Full-Order System Models.....	56
4.2 Design Procedure for Full-Order System Models.....	57
4.3 Design Philosophy for Reduced-Order System Models.....	60
4.4 Design Procedure for Reduced-Order System Models and Observers.....	63

## TABLE OF CONTENTS - Concluded

Chapter	Page
4.5 Cost Function.....	67
4.6 Cost Function Gradient.....	69
5 DESIGN EXAMPLE.....	77
5.1 Original Lateral Axis Model.....	77
5.2 Reduced-Order Model Design.....	96
5.3 Summary and Concluding Remarks.....	100
REFERENCES.....	112
APPENDICES	
APPENDIX A. SOFTWARE LISTING.....	115
APPENDIX B. DESIGN SESSION EXAMPLE.....	124

## LIST OF FIGURES

Figure	Page
2.1 Linear Time Invariant System Model.....	8
2.2 System Model with Constant State Feedback.....	14
2.3 Eigenvector Projection into Allowable Subspace.....	25
3.1 Original System and Proposed Reduced-Order Observer.....	42
3.2 Proposed Reduced-Order Observer without Differentiator....	43
3.3 Modified Reduced-Order Observer.....	46
3.4 Reduced-Order Observer.....	47
3.5 Control Law Implemented with Reduced-Order Observer.....	49
4.1 Eigenvalue/Eigenvector Assignment Design Philosophy.....	58
4.2 Spectral Assignment Computer Software Package Organization.....	59
4.3 Reduced-Order Design Philosophy.....	62
4.4 Modified Spectral Assignment Computer Software Package Organization.....	64
4.5 Mode 9.....	66
4.6 Subroutine VACT.....	68
4.7 Subroutine ROCOST.....	70
4.8 Subroutine ROGRAD.....	71
5.1 Aircraft Axis System.....	79
5.2 Yaw Rate - Open Loop Response for $\phi(0) = 1^\circ$ .....	81
5.3 Sideslip Angle - Open Loop Response for $\phi(0) = 1^\circ$ .....	82

LIST OF FIGURES - Continued

Figure	Page
5.4 Roll Rate - Open Loop Response for $\phi(0) = 1^\circ$ .....	83
5.5 Bank Angle - Open Loop Response for $\phi(0) = 1^\circ$ .....	84
5.6 Rudder Deflection - Open Loop Response for $\phi(0) = 1^\circ$ .....	85
5.7 Aileron Deflection - Open Loop Response for $\phi(0) = 1^\circ$ .....	86
5.8 Washout Filter - Open Loop Response for $\phi(0) = 1^\circ$ .....	87
5.9 Yaw Rate - First Closed Loop Response for $\phi(0) = 1^\circ$ .....	89
5.10 Sideslip Angle - First Closed Loop Response for $\phi(0) = 1^\circ$ .....	90
5.11 Roll Rate - First Closed Loop Response for $\phi(0) = 1^\circ$ .....	91
5.12 Bank Angle - First Closed Loop Response for $\phi(0) = 1^\circ$ .....	92
5.13 Rudder Deflection - First Closed Loop Response for $\phi(0) = 1^\circ$ .....	93
5.14 Aileron Deflection - First Closed Loop Response for $\phi(0) = 1^\circ$ .....	94
5.15 Washout Filter - First Closed Loop Response for $\phi(0) = 1^\circ$ .....	95
5.16 Yaw Rate - Second Closed Loop Response for $\phi(0) = 1^\circ$ .....	101
5.17 Sideslip Angle - Second Closed Loop Response for $\phi(0) = 1^\circ$ .....	102
5.18 Roll Rate - Second Closed Loop Response for $\phi(0) = 1^\circ$ .....	103
5.19 Bank Angle - Second Closed Loop Response for $\phi(0) = 1^\circ$ .....	104
5.20 Rudder Deflection - Second Closed Loop Response for $\phi(0) = 1^\circ$ .....	105
5.21 Aileron Deflection - Second Closed Loop Response for $\phi(0) = 1^\circ$ .....	106
5.22 Washout Filter - Second Closed Loop Response for $\phi(0) = 1^\circ$ .....	107
5.23 Observer State #1 - Closed Loop Response for $\phi(0) = 1^\circ$ .....	108



## LIST OF FIGURES - Concluded

Figure		Page
5.24	Observer State #2 - Closed Loop Response for $\phi(0) = 1^\circ \dots$	109
5.25	Observer State #3 - Closed Loop Response for $\phi(0) = 1^\circ \dots$	110

## LIST OF SYMBOLS

$A$	System plant matrix
$\tilde{A}$	Matrix $A$ transformed by $M$
$\tilde{A}_{ij}$	Partition of $\tilde{A}$
$A_T$	Closed loop matrix $[\tilde{A} + \tilde{B} \tilde{F}]$
$B$	Input matrix
$B$	Input matrix $B$ transformed by $V$
$B_i$	$i$ th partition of $B$
$\tilde{B}$	Input matrix $B$ transformed by $M$
$\tilde{B}_i$	$i$ th partition of $\tilde{B}$
$C$	Output matrix
$\tilde{C}$	Output matrix transformed by $M$
$\tilde{C}_i$	$i$ th partition of output matrix
$\mathbb{C}$	Set of complex scalars
$\mathbb{C}^n$	$n$ dimensional complex space
$E$	Observer system plant matrix
$F$	Full state feedback matrix
$F$	Reduced-order model state feedback matrix
$\bar{F}$	$F'$ transformed by $V^{-1}$
$F'$	$[F:0]$
$\tilde{F}$	$\bar{F}$ transformed by $M$
$\tilde{F}_i$	$i$ th partition of $\tilde{F}$
$fw$	Washout filter state

LIST OF SYMBOLS - Continued

G	Observer input matrix
GR	Gradient matrix
I	Identity matrix
$I_n$	nth order identity
J	Cost function
$K_\alpha$	Complex null space matrix
$K_\beta$	Complex null space matrix
$K_\gamma$	Complex null space matrix
$K_\lambda$	Complex null space matrix
$K_{\lambda i}$	Real null space matrix
L	Reduced-order observer feedback matrix
M	Transformation matrix = $[\zeta^1 \ \zeta^2]$
$M_\lambda$	Complex null space matrix
$M_{\lambda i}$	Real null space matrix
$N_\lambda$	Complex null space matrix
$N_{\lambda i}$	Real null space matrix
$P_\lambda$	Complex null space matrix
$P_{N_\lambda}$	Real projection matrix
$P_{K_\lambda}$	Complex projection matrix
p	Roll rate (radians/second)
R	Reduced order observer matrix
R	Set of real scalars
$R^n$	n dimensional real space
$S_\lambda$	Complex null space matrix
$S_{\lambda i}$	Real null space matrix for ith eigenvalue
T	Matrix $[-L:I]$

LIST OF SYMBOLS - Continued

$U$	Left eigenvector matrix
$u$	Input vector
$u_i$	$i$ th left eigenvector
$V$	Right eigenvector matrix
$V_{ij}$	Partition of $V$
$v$	Right eigenvector
$v_i$	$i$ th right eigenvector
$v_{ij}$	Element of $V$
$\hat{v}_i$	$i$ th eigenvector of reduced-order model
$\hat{v}_{ij}$	$i$ th component of $v_j$
$\bar{v}_i$	Eigenvector assigned in reduced-order model
$\bar{v}_{ij}$	$i$ th partition of $\bar{V}_j$
$v_D$	Desired partial eigenvector assignment
$v_{Dij}$	$i$ th component of $j$ th $V_D$
$v_T$	Eigenvector of total system
$v_{Ti}$	$i$ th partition of $V_T$
$W$	Null space matrix consisting of all $w$
$w$	Reduced-order observer state vector
$w_i$	Null space vector
$r$	Yaw rate (radians/second)
$X$	Designator matrix (real)
$X_i$	$i$ th column of designator matrix
$x$	State vector
$x_0$	State vector at $t = 0$
$X_{ij}$	Element of $X$
$x_{c1}$	Complex designator vector
$x_{c2}$	Complex designator vector

## LIST OF SYMBOLS - Concluded

$x_T$	Complex designator vector
$y$	Output vector
$z$	State vector transformed by $V$
$z_i$	$i$ th partition of $Z$
$\tilde{z}$	Reduced-order model state vector after transformation
$\tilde{z}_i$	$i$ th partition of $\tilde{Z}$
$\alpha$	Complex null space matrix
$\beta$	Sideslip angle (radians)
$\delta_a$	Aileron deflection (radians)
$\delta_r$	Rudder deflection (radians)
$\gamma$	Complex null space matrix
$\theta$	Proposed reduced-order observer state vector
$\Lambda$	Diagonal eigenvalue matrix
$\Lambda_i$	$i$ th partition of $\Lambda$
$\lambda$	Eigenvalue
$\lambda_i$	$i$ th eigenvalue
$\phi$	Bank angle (radians)
$\tau$	Convolution variable
$\Omega$	Reduced-order observer input matrix
$( )^T$	Transpose of quantity
$( )^{-1}$	Inverse of quantity
$( )^*$	Complex conjugate of quantity
$( )^\perp$	Orthogonal complement of quantity
$( )^{-L}$	Left inverse of quantity
$( )_{\text{Re}}$	Real component of quantity
$( )_{\text{Im}}$	Imaginary component of quantity



## CHAPTER 1

### INTRODUCTION

The use of reduced-order models [1] and reduced-order observers [2] in the design of feedback controllers has been studied by several researchers. In addition, the development of eigenvalue/eigenvector assignment techniques has received much attention in recent years. In this work, a reduced-order model is used with eigenvalue/eigenvector assignment techniques to design a constant state feedback controller for the original full-order system. The eigenvalues and eigenvectors contained in the reduced-order model are reassigned in the full-order system while those eigenvalues and associated eigenvectors not included in the reduced-order model remain unchanged in the full-order system. The constant state feedback matrix is implemented using output feedback with a reduced-order observer. It is shown that the eigenvalues and eigenvectors of the closed loop full-order system remain unchanged when the reduced-order observer is implemented.

#### 1.1 Motivation

During the past fifteen years significant advances have been made toward developing viable synthesis techniques for multivariable feedback control systems. Notable among these techniques is the eigenvalue/

eigenvector assignment procedure. Early studies in this area focused on an algorithmic formulation of the spectral assignment by Srinathkumar [3], while later studies included a geometric formulation of the same problem by Moore [4], Kimura [5], and Davison and Wang [6]. Based on these theories, design procedures have been developed for approximating desired mode mixing [7], reducing eigensystem sensitivity to variations in plant parameters [8], reducing the effects of actuator noise on system performance [9] and modifying the resultant feedback gain matrix to specified gain constraints [10]. Recently, these procedures have been incorporated in a spectral assignment computer aided design package [11]. A deficiency in all work concerning eigenvalue/eigenvector assignment procedures is an absence of application of these techniques to real world design problems. A primary factor contributing to this problem is the lack of understanding of how to use reduced-order models and reduced-order observers with spectral assignment procedures.

Models representing the behavior of physical systems often consist of a very large number of coupled, linear differential equations. Such models are difficult to use when designing control systems due to excessive requirements for computer time and memory, and to the numerical analysis problems inherently present when dealing with large systems of equations. It is, therefore, desirable to develop a design procedure which utilizes reduced-order system models. Simplification of large order dynamic systems has received the attention of many researchers in recent years. The major difficulty with this work is that only open-loop system behavior is approximated. Of concern when using reduced-



order models with eigenvalue/eigenvector assignment procedures is the fact that while the reduced-order model may approximate open-loop system behavior, the modeling error may be so great or of such a nature that actual closed-loop system performance is not acceptable. Also of concern is the closed-loop behavior of those modes of the original system which are not included in the reduced-order model.

Full state feedback is implemented by the use of a dynamic observer system when there are fewer outputs than states. Since some states are usually available for measurement at the output, a reduced-order observer is desirable in order to minimize the complexity of the control system. Of concern is the effect of a reduced-order observer on the system eigenvalues and eigenvectors.

## 1.2 Overview

In this section an overview of the thesis is given. A background of spectral assignment theory is discussed in Chapter 2. A subsequent design procedure implemented by Marefat in a computer aided design package is presented next. This information provides a necessary foundation to support the material in the remaining chapters. In Chapter 3, a new technique is developed that uses a reduced-order model of a known larger system and spectral assignment procedures to reassign selected eigenvalues of the system. This is accomplished without affecting the eigenvalues and eigenvectors not included in the reduced-order model. Secondly, a technique is developed that uses Luenberger's [2] reduced-order observer and spectral assignment procedures to implement a constant full

state feedback design using dynamic output feedback. The assigned eigenvalues and eigenvectors of the original system are retained using this technique. In Chapter 4, a design philosophy and then a corresponding design procedure are developed for the new synthesis techniques presented in Chapter 3. A software package is developed to facilitate the design of dynamic output feedback control systems using this new philosophy and procedure. The package is included as a new mode to an existing spectral assignment computer aided design program. The use of spectral assignment with reduced-order models and reduced-order observers in an actual design problem is demonstrated in Chapter 5. Results are compared to those obtained by an alternate design procedure. A program listing and an example of a computer aided design session are included as appendices.

## CHAPTER 2

### SPECTRAL ASSIGNMENT PROCEDURE

In this chapter a background of spectral assignment theory is presented to support the development in Chapter 3. Definitions of eigenvalues and eigenvectors are given. Then the effect of eigenvalues and eigenvectors on the time response of a system is presented. Lastly, a characterization of the freedom available in selecting eigenvectors for a given eigenvalue assignment using constant state feedback is presented.

#### 2.1 System Eigenstructure and Time Response

The eigenvalues of an  $n$ th order real matrix  $A$  are the zeros of the polynomial  $\det [\lambda I - A]$ . The eigenvalues,  $\lambda_i \in \mathbb{C}$ , form a self-conjugate set. That is, for each complex eigenvalue  $\lambda_i$  there exists a complex conjugate eigenvalue  $\lambda_{i+1} = \lambda_i^*$ . For each eigenvalue  $\lambda_i$ , there is a right eigenvector,  $v_i \in \mathbb{C}^n$ , that satisfies the equation,

$$Av_i = v_i \lambda_i, \quad (2.1)$$

for  $i = 1, \dots, n$ . If the eigenvalues of  $A$  form a distinct set, then the associated eigenvectors are linearly independent [11]. Equation

(2.1) is written for all  $\lambda_i$  and  $v_i$  as

$$AV = V\Lambda, \quad (2.2)$$

where  $V = [V_1 \dots V_n]$  and  $\Lambda = \text{diag.}(\lambda_1, \dots, \lambda_n)$ . Since the columns of  $V$  are linearly independent,  $V$  is invertible. Therefore,

$$A = V\Lambda V^{-1}. \quad (2.3)$$

Similarly, for each eigenvalue  $\lambda_i$  there is a left eigenvector,  $u_i \in \mathbb{C}^n$ , that satisfies the equation

$$u_i^T A = \lambda_i u_i^T \quad (2.4)$$

for  $i = 1, \dots, n$ . The left eigenvector equation is written for all  $\lambda_i$  and  $u_i$  as

$$U^T A = \Lambda U^T \quad (2.5)$$

where  $U = [u_1 \dots u_n]$ . For distinct eigenvalues, the left eigenvectors are also linearly independent [12]. Hence, premultiplying equation (2.5) by  $(U^T)^{-1}$  yields

$$A = (U^T)^{-1} \Lambda U^T. \quad (2.6)$$

Substituting for  $A$  from equation (2.3) into equation (2.6) yields

$$V\Lambda V^{-1} = (U^T)^{-1} \Lambda U^T. \quad (2.7)$$

Premultiplying by  $U^T$  and postmultiplying by  $V$  yields

$$U^T V \Lambda = \Lambda U^T V. \quad (2.8)$$

Since  $\Lambda$  is a diagonal matrix of distinct eigenvalues, equation (2.8) can only be satisfied if  $U^T V$  is a diagonal matrix. For convenience the eigenvectors are usually normalized so that  $U^T V = I$  or

$$U^T = V^{-1}. \quad (2.9)$$

The effect of eigenvalues and eigenvectors on system time response is now presented. Consider the linear time invariant system in Figure 2.1 represented by the system state equations

$$\dot{x} = Ax + Bu \quad (2.10)$$

and

$$y = Cx, \quad (2.11)$$

where  $A$ ,  $B$ , and  $C$  are the plant, input, and output matrices respectively and  $x \in \mathbb{R}^n$ ,  $u \in \mathbb{R}^m$ , and  $y \in \mathbb{R}^p$ . The system time response is determined by solving the differential equation (2.10). Let a change of coordinates be defined by

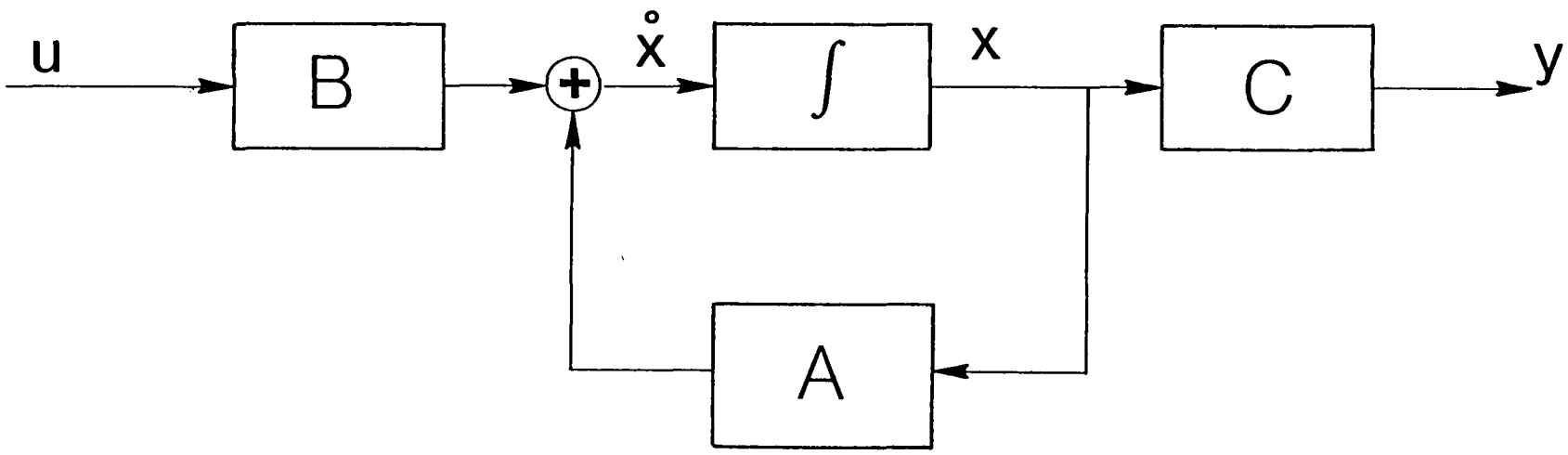


Figure 2.1. Linear Time Invariant System Model

$$x = Vz. \quad (2.12)$$

The transformed system is

$$\dot{z} = V^{-1}AVz + V^{-1}Bu \quad (2.13)$$

$$y = CVz. \quad (2.14)$$

Substituting from equation (2.3) into equation (2.13) yields

$$\dot{z} = \Lambda z + V^{-1}Bu. \quad (2.15)$$

The solution of equation (2.15) is given by [13]

$$z(t) = e^{\Lambda t} z_0 + \int_0^t e^{\Lambda(t-\tau)} U^T B u(\tau) d\tau \quad (2.16)$$

where  $z_0$  is the initial value of  $z(t)$  at  $t = 0$ . Substituting from equation (2.12) yields the time response

$$x(t) = V e^{\Lambda t} U^T x_0 + V \int_0^t e^{\Lambda(t-\tau)} U^T B u(\tau) d\tau. \quad (2.17)$$

The first term of equation (2.17) is called the zero input response and the second term is called the zero state response.

Expanding the zero input response yields

$$\begin{aligned}
 x(t) &= [v_1, \dots, v_n] \begin{bmatrix} e^{\lambda_1 t} & & & 0 \\ & \cdot & & \vdots \\ \vdots & & \cdot & \\ 0 & \dots & & e^{\lambda_n t} \end{bmatrix} \begin{bmatrix} u_1^T \\ \vdots \\ u_n^T \end{bmatrix} x_0 \\
 &= \begin{bmatrix} v_{11} e^{\lambda_1 t} (u_1^T x_0) + \dots + v_{1n} e^{\lambda_n t} (u_n^T x_0) \\ \vdots \\ v_{n1} e^{\lambda_1 t} (u_1^T x_0) + \dots + v_{nn} e^{\lambda_n t} (u_n^T x_0) \end{bmatrix} \quad (2.18)
 \end{aligned}$$

From equation (2.18) the  $i$ th component of the statevector is determined to be

$$x_i(t) = \sum_{j=1}^n v_{ij} e^{\lambda_j t} (u_j^T x_0). \quad (2.19)$$

Expanding equation (2.19) yields

$$\begin{bmatrix} x_1(t) \\ \vdots \\ x_n(t) \end{bmatrix} = \begin{bmatrix} v_{11} \\ \vdots \\ v_{n1} \end{bmatrix} e^{\lambda_1 t} (u_1^T x_0) + \dots + \begin{bmatrix} v_{1n} \\ \vdots \\ v_{nn} \end{bmatrix} e^{\lambda_n t} (u_n^T x_0). \quad (2.20)$$

The zero state response is expanded next. Let the input vector  $u(t)$  be a vector of unit step functions,  $u_0$ , for computational ease.



This yields

$$\begin{aligned} x(t) &= V \int_0^t e^{\Lambda(t-\tau)} U^T_B u_0 d\tau \\ &= (V e^{\Lambda t}) \int_0^t e^{-\Lambda\tau} d\tau (U^T_B u_0). \end{aligned} \quad (2.21)$$

Since  $\Lambda$  is a diagonal matrix, the integral term is written as

$$\begin{aligned} \int_0^t e^{-\Lambda\tau} d\tau &= \begin{bmatrix} \int_0^t e^{-\lambda_1\tau} d\tau & & \dots & 0 \\ \vdots & & & \vdots \\ 0 & \dots & \int_0^t e^{-\lambda_n\tau} d\tau & \end{bmatrix} \\ &= \begin{bmatrix} -\frac{1}{\lambda_1} (e^{-\lambda_1 t} - 1) & & \dots & 0 \\ \vdots & & & \vdots \\ 0 & \dots & -\frac{1}{\lambda_n} (e^{-\lambda_n t} - 1) & \end{bmatrix}. \end{aligned} \quad (2.22)$$

Premultiplying equation (2.22) by the diagonal matrix  $e^{\Lambda t}$  yields

$$e^{\Lambda t} \int_0^t e^{-\Lambda\tau} d\tau = -\frac{1}{\lambda_1} \begin{bmatrix} (1 - e^{-\lambda_1 t}) & & \dots & 0 \\ \vdots & & & \vdots \\ 0 & \dots & -\frac{1}{\lambda_n} (1 - e^{-\lambda_n t}) & \end{bmatrix}. \quad (2.23)$$

The vector  $K$  is defined to be

$$K = U^T B u_0 . \quad (2.24)$$

Substituting equations (2.23) and (2.24) into (2.21) gives

$$x(t) = [v_1 \dots v_n] \begin{bmatrix} -\frac{1}{\lambda_1} (1 - e^{\lambda_1 t}) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & -\frac{1}{\lambda_n} (1 - e^{\lambda_n t}) \end{bmatrix} K$$

$$= \sum_{i=1}^n v_i \left( \frac{-K_i}{\lambda_i} \right) (1 - e^{\lambda_i t}) \quad (2.25)$$

where  $K_i$  denotes the  $i$ th element of  $K$ . Expanding equation (2.25) yields

$$\begin{bmatrix} x_1(t) \\ \vdots \\ x_n(t) \end{bmatrix} = \begin{bmatrix} v_{11} \\ \vdots \\ v_{n1} \end{bmatrix} \left( \frac{-K_1}{\lambda_1} \right) (1 - e^{\lambda_1 t}) + \dots + \begin{bmatrix} v_{1n} \\ \vdots \\ v_{nn} \end{bmatrix} \left( \frac{-K_n}{\lambda_n} \right) (1 - e^{\lambda_n t}) . \quad (2.26)$$

The terms  $e^{\lambda_i t}$  are called the modes of the system. Equations (2.20) and (2.26) show that the eigenvalues of the system determine the rates of decay of the modes while the eigenvectors determine the contribution of each mode to the various states. Thus, the time response of a system

can be controlled by proper selection of system eigenvalues and eigenvectors.

## 2.2 Characterization of Freedom in Eigenvector Assignment

Given the linear, time invariant controllable system with constant state feedback in Figure 2.2, the system state equations are written

$$\dot{x} = (A + BF) x + Bv \quad (2.27)$$

$$y = Cx. \quad (2.28)$$

Given that constant state feedback is used, Wonham [14] states that an  $m \times n$  matrix  $F$  can be found to assign an arbitrary self conjugate set of eigenvalues if the system is controllable. Moore [4] characterizes the freedom available to assign eigenvectors for an arbitrary self-conjugate set of eigenvalues. He gives necessary and sufficient conditions to find a unique real matrix  $F$  that satisfies the eigenvector equation

$$(A + BF) v_i = v_i \lambda_i \quad (2.29)$$

For  $i = 1, \dots, n$  when  $B$  has full column rank. Associate with each eigenvalue  $\lambda_i$  an  $n \times (n+m)$  matrix  $S_{\lambda_i}$  where

$$S_{\lambda_i} = [\lambda_i I - A : B] \quad (2.30)$$

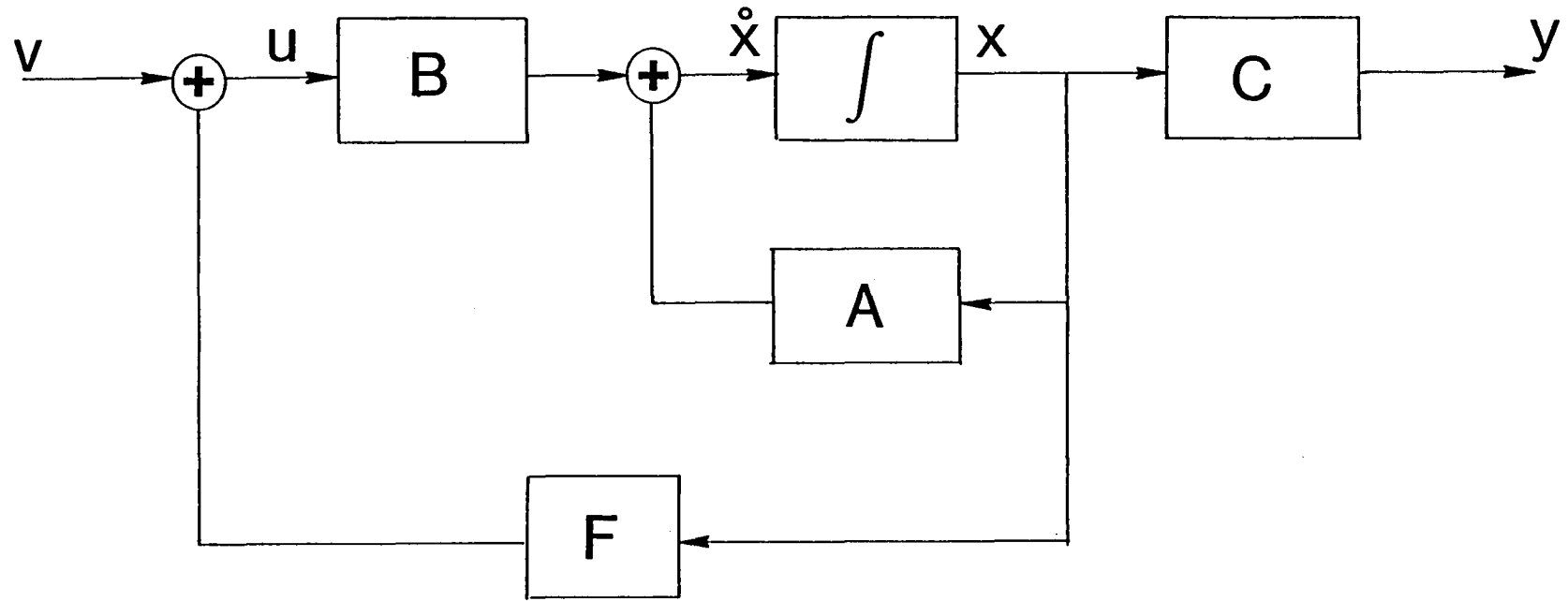


Figure 2.2. System Model with Constant State Feedback

and a compatibly partitioned  $(n+m) \times n$  matrix

$$K_{\lambda_i} = \begin{bmatrix} N_{\lambda_i} \\ \hline M_{\lambda_i} \end{bmatrix} \quad (2.31)$$

whose columns constitute a basis for the null space of  $S_{\lambda_i}$ . Then the necessary and sufficient conditions to find a unique real matrix  $F$  that satisfies equation (2.29) are:

- 1) Vectors  $v_i \in \mathbb{C}^n$  are linearly independent,
- 2)  $v_i = v_j^*$  whenever  $\lambda_i = \lambda_j^*$ , and
- 3)  $v_i \in \text{span}(N_{\lambda_i})$ .

Thus it is possible to assign an arbitrary selfconjugate set of eigenvalues and a set of eigenvectors from within the span of  $N_{\lambda_i}$ . The null space  $N_{\lambda_i}$  is determined by the selection of an eigenvalue  $\lambda_i$ . The subspace,  $N_{\lambda_i}$  identifies the freedom available to assign eigenvector  $v_i$ .

### 2.3 Eigenvector Assignment for Real Eigenvalues

It is first assumed that  $\lambda_i \in \mathbb{R}$  so that  $v_i \in \mathbb{R}^n$  for  $i = 1, \dots, n$ . Equation (2.29) is rewritten as

$$(\lambda_i I - A) v_i - (BF) v_i = 0. \quad (2.32)$$

Since  $K_{\lambda_i}$  is a basis for the null space of  $S_{\lambda_i}$ , then any vector  $K_i$  that postmultiplies  $K_{\lambda_i}$  gives a resulting vector that lies in the null space of  $S_{\lambda_i}$ . Therefore,

$$[\lambda_i I - A \ : \ B] \begin{bmatrix} N_{\lambda_i} \\ M_{\lambda_i} \end{bmatrix} K_i = 0. \quad (2.33)$$

Expanding equation (2.33) yields

$$[\lambda_i I - A] N_{\lambda_i} K_i + [B] M_{\lambda_i} K_i = 0. \quad (2.34)$$

Since  $v_i \in \text{span}(N_{\lambda_i})$ , then  $K_i$  determines where in the allowable subspace  $v_i$  exists. Hence,

$$v_i = N_{\lambda_i} K_i. \quad (2.35)$$

It follows from equations (2.32) and (2.35) that

$$Fv_i = -M_{\lambda_i} K_i. \quad (2.36)$$

By defining  $w_i$  as

$$w_i = -M_{\lambda_i} K_i, \quad (2.37)$$

equation (2.36) is rewritten in matrix form for all  $i$  as

$$F[v_1, \dots, v_n] = [w_1, \dots, w_n] \quad (2.38)$$

or

$$FV=W. \quad (2.39)$$

Since the eigenvectors are linearly independent, then

$$F = W V^{-1}. \quad (2.40)$$

#### 2.4 Eigenvector Assignment for Complex Eigenvalues

It is next assumed that  $\lambda_i \in \mathbb{C}$  for  $i = 1, 2$  and  $\lambda_i \in \mathbb{R}$  for  $i = 3, \dots, n$ . Then the first closed loop right eigenvector must satisfy the equation

$$[A + BF] (v_{RE} + jv_{IM}) = (v_{RE} + jv_{IM}) (\lambda_{RE} + j\lambda_{IM}) \quad (2.41)$$

where the subscript one is suppressed for simplicity. Equating real and imaginary parts yields

$$[A + BF] v_{RE} = v_{RE} \lambda_{RE} - v_{IM} \lambda_{IM} \quad (2.42)$$

and

$$[A + BF] v_{IM} = v_{IM} \lambda_{RE} + v_{RE} \lambda_{IM} \quad (2.43)$$

The two equations are written in matrix form as

$$[\lambda_{RE} I - A : \lambda_{IM} I : B] \begin{bmatrix} v_{RE} \\ -v_{IM} \\ -Fv_{RE} \end{bmatrix} = 0 \quad (2.44)$$

and

$$[\lambda_{RE} I - A : \lambda_{IM} I : B] \begin{bmatrix} v_{IM} \\ -v_{RE} \\ -Fv_{IM} \end{bmatrix} = 0. \quad (2.45)$$

For the case of complex eigenvalues, the  $n \times (2n+m)$  matrix  $S_\lambda$  is defined as

$$S_\lambda = [\lambda_{RE} I - A : \lambda_{IM} I : B] \quad (2.46)$$

and a compatibly partitioned  $(2n+m) \times n$  matrix  $K_\lambda$  is defined by

$$K_\lambda = \begin{bmatrix} N_\lambda \\ \hline P_\lambda \\ \hline M_\lambda \end{bmatrix} \quad (2.47)$$

where the columns of  $K_\lambda$  constitute a basis for the null space of  $S_\lambda$ .  
Hence,



$$\begin{bmatrix} \lambda_{RE} I - A \\ \lambda_{IM} I - B \end{bmatrix} \begin{bmatrix} N_\lambda \\ P_\lambda \\ M_\lambda \end{bmatrix} = 0. \quad (2.48)$$

From equations (2.44), (2.45) and (2.48) it is apparent that the vectors in (2.44) and (2.45) are contained in the null space defined by  $K_\lambda$ .

Therefore

$$\begin{bmatrix} v_{RE} \\ -v_{IM} \\ -Fv_{RE} \end{bmatrix} \in \text{SPAN} \begin{bmatrix} N_\lambda \\ P_\lambda \\ M_\lambda \end{bmatrix} \quad (2.49)$$

and

$$\begin{bmatrix} v_{IM} \\ v_{RE} \\ -Fv_{IM} \end{bmatrix} \in \text{SPAN} \begin{bmatrix} N_\lambda \\ P_\lambda \\ M_\lambda \end{bmatrix}. \quad (2.50)$$

From equations (2.49) and (2.50) it is apparent that the allowable subspace for  $v_{RE}$  and  $v_{IM}$  is described by

$$\begin{bmatrix} v_{RE} \\ v_{IM} \end{bmatrix} \in \text{SPAN} \begin{bmatrix} N_\lambda \\ -P_\lambda \end{bmatrix} \quad (2.51)$$

and

$$\begin{bmatrix} V_{RE} \\ V_{IM} \end{bmatrix} \in \text{SPAN} \begin{bmatrix} P_\lambda \\ N_\lambda \end{bmatrix} \quad (2.52)$$

Combining equations (2.51) and (2.52) yields

$$\begin{bmatrix} V_{RE} \\ V_{IM} \end{bmatrix} \in \text{SPAN} \begin{bmatrix} N_\lambda \\ -P_\lambda \end{bmatrix} \cap \text{SPAN} \begin{bmatrix} P_\lambda \\ N_\lambda \end{bmatrix} . \quad (2.53)$$

The following characterization of the freedom available to assign complex eigenvectors is not developed here but is proved by Marefat [15]. Matrixes  $\alpha$  and  $\beta$  are defined by

$$\alpha = \begin{bmatrix} N_\lambda \\ -P_\lambda \end{bmatrix}^T \quad (2.54)$$

and

$$\beta = \begin{bmatrix} P_\lambda \\ N_\lambda \end{bmatrix}^T . \quad (2.55)$$

$K_\alpha$  and  $K_\beta$  are defined to be matrices whose columns constitute bases for the null spaces of  $\alpha$  and  $\beta$ , respectively. Matrix  $\gamma$  is defined by

$$\gamma = [K_\alpha \text{ : } K_\beta]^T \quad (2.56)$$

and  $K_\gamma$  is defined to be a matrix whose columns constitute a basis for the null space of  $\gamma$ . A basis for  $[\text{SPAN}(\alpha) \cap \text{SPAN}(\beta)]^\perp$  is  $[(\text{SPAN}(\alpha))^\perp + (\text{SPAN}(\beta))^\perp]^\perp$  where "+" denotes set direct summation and " $\perp$ " denotes orthogonal complementation. Also, a basis for  $\gamma^\perp$  is a basis for  $\gamma^T$ . Hence

$$\begin{bmatrix} v_{RE} \\ v_{IM} \end{bmatrix} \in \text{SPAN}(K_\gamma). \quad (2.57)$$

A specific vector within the null space of  $\gamma$  is defined by postmultiplying  $K_\gamma$  by a vector  $x_T$ . Thus

$$\begin{bmatrix} v_{RE} \\ v_{IM} \end{bmatrix} = K_\gamma x_T. \quad (2.58)$$

Using equations (2.51) and (2.52),  $x_{c1}$  and  $x_{c2}$  are defined by

$$\begin{bmatrix} v_{RE} \\ v_{IM} \end{bmatrix} = \begin{bmatrix} N_\lambda \\ -P_\lambda \end{bmatrix} x_{c1} \quad (2.59)$$

and

$$\begin{bmatrix} v_{RE} \\ v_{IM} \end{bmatrix} = \begin{bmatrix} P_\lambda \\ N_\lambda \end{bmatrix} x_{c2}. \quad (2.60)$$

The left inverses of  $\begin{bmatrix} N_\lambda \\ -P_\lambda \end{bmatrix}$  and  $\begin{bmatrix} P_\lambda \\ N_\lambda \end{bmatrix}$  exist since the columns of  $K_\lambda$  are linearly independent. Therefore

$$x_{c1} = \begin{bmatrix} N_\lambda \\ -P_\lambda \end{bmatrix}^{-L} \begin{bmatrix} v_{RE} \\ v_{IM} \end{bmatrix} \quad (2.61)$$

and

$$x_{c2} = \begin{bmatrix} P_\lambda \\ N_\lambda \end{bmatrix}^{-L} \begin{bmatrix} v_{RE} \\ v_{IM} \end{bmatrix} \quad (2.62)$$

From equations (2.48) and (2.49) it is apparent that the vectors  $[-Fv_{RE}]$  and  $[-Fv_{IM}]$  lie in the space defined by the columns of  $M_\lambda$ . Hence

$$-Fv_{RE} = M_\lambda x_{c1} \quad (2.63)$$

and

$$-Fv_{IM} = M_\lambda x_{c2} \quad (2.64)$$

Since  $\lambda_1$  and  $\lambda_2 \in \mathbb{C}$  and  $\lambda_i \in \mathbb{R}$  for  $i = 3, \dots, n$ , then  $\lambda_2 = \lambda_1^*$  because the eigenvalues form a self conjugate set. Furthermore, the second condition of spectral assignment requires that  $v_2 = v_1^*$ . Thus the specification of one complex eigenvalue and eigenvector contains all the essential information of the complex conjugate pair. It is also important to note that if  $v_2 = v_2^*$  and the pair  $v_1, v_2$  are linearly independent, then  $v_{RE}$  and  $v_{IM}$  are also linearly independent. In order to calculate the feedback matrix  $F$ , the following

definitions are given:

$$w_1 = M_{\lambda} x_{c1}, \quad (2.65)$$

$$w_2 = -M_{\lambda} x_{c2}, \quad (2.66)$$

$$v_1 = v_{RE}, \quad (2.67)$$

and

$$v_2 = v_{IM}. \quad (2.68)$$

Recalling that for the case of real eigenvalues

$$w_i = -M_{\lambda_i} x_i \quad (2.69)$$

and

$$v_i = v_i, \quad (2.70)$$

equation (2.38) is rewritten so that

$$F[v_{RE}, v_{IM}, v_3, \dots, v_n] = [-M_{\lambda} x_{c1}, -M_{\lambda} x_{c2}, -M_{\lambda_3} x_3, \dots, -M_{\lambda_n} x_n]. \quad (2.71)$$

Substituting equations (2.65) through (2.70) into (2.71) yields

$$F[v_1, \dots, v_n] = [w_1, \dots, w_n] \quad (2.72)$$

or

$$FV = W. \quad (2.73)$$

As in the case for real eigenvalues,

$$F = WV^{-1}. \quad (2.74)$$

This development is easily extended to more than one pair of complex conjugate eigenvalues.

### 2.5 Use of Eigenvector Freedom

It is shown in the previous three sections that eigenvectors for the selected eigenvalues must reside in an allowable subspace that is determined by the plant matrix  $A$ , the input matrix  $B$ , and the selected eigenvalues  $\lambda_i$ . Normally the eigenvector assignment that is most desirable for a given set of eigenvalues is not achievable because it does not lie within the allowable eigenvector space. In this case it is desirable to select the allowable eigenvector that is closest to the desired eigenvector. This is accomplished by projecting the desired vector into the allowable space so that the error between the desired and the assigned vector is minimized in a least squares sense as illustrated in Figure 2.3.

The desired vector is projected onto the allowable space by the projection operator

$$P_{N_\lambda} = N_\lambda (N_\lambda^T N_\lambda)^{-1} N_\lambda^T \quad (2.75)$$

for real eigenvalues and

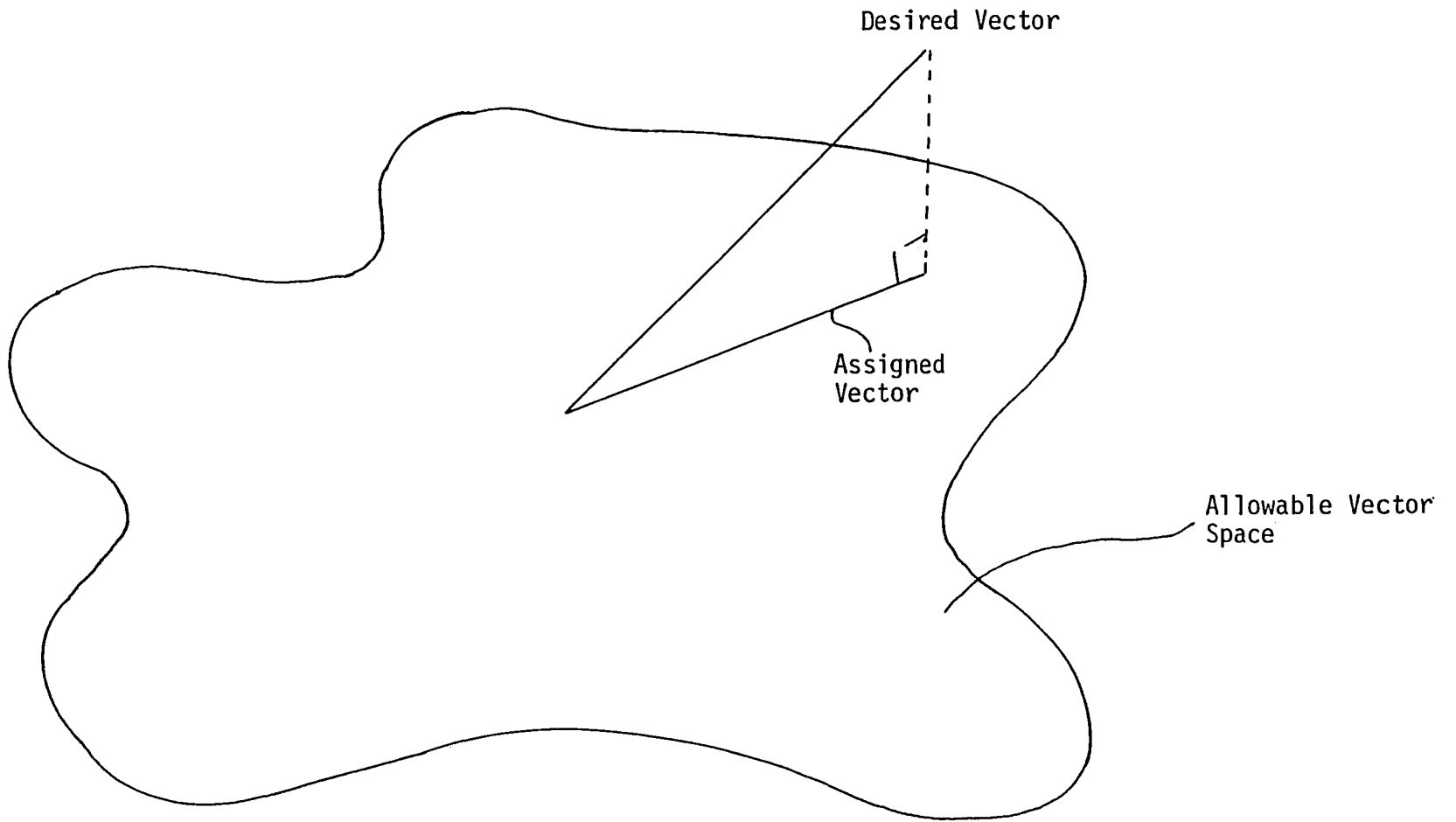


Figure 2.3. Eigenvector Projection into Allowable Subspace

$$P_{K_Y} = K_Y (K_Y^T K_Y)^{-1} K_Y^T \quad (2.76)$$

for complex eigenvalues [15]. Indicating the desired vector by the subscript "D" and the assigned vector by the subscript "A", the projection is accomplished by the equations

$$v_A = P_{N_\lambda} v_D \quad (2.77)$$

and

$$\begin{bmatrix} v_{RE} \\ v_{IM} \end{bmatrix}_A = P_{K_Y} \begin{bmatrix} v_{RE} \\ v_{IM} \end{bmatrix}_D \quad (2.78)$$

for the real and complex assignments, respectively.

## 2.6 Improvement of Initial Assignment by Gradient Search

The selection of eigenvalues and eigenvectors for a system is normally motivated by the desire to shape the time response as discussed in Section 2.1. However, once the desired time response is approximated, there are often other aspects of the assignment that are unacceptable. An example is an assignment which requires extremely high feedback gains which are expensive to implement and very sensitive to noise. Another example is extreme eigensystem sensitivity to small plant parameter variations or modeling errors. The freedom available to select the eigenvectors often provides a means to drastically improve these secondary design objectives while only slightly modifying the initial eigen-



vector assignment and thus the time response. This improvement is accomplished by modifying the eigenvectors within an area local to the original assignment. The vectors are modified in such a manner as to reduce the undesirable aspect of the assignment most rapidly.

A cost function  $J$  is defined so that a reduction in the value of  $J$  corresponds to reduction of the undesirable aspect of an initial eigenvector assignment. A gradient matrix is computed in terms of  $J$  to determine how the eigenvector assignment is most efficiently changed. Recalling that the eigenvectors are determined by the equation

$$v_i = N_{\lambda_i} X_i \quad (2.79)$$

for the case of real eigenvalues and

$$\begin{bmatrix} v_{iRE} \\ v_{iIM} \end{bmatrix} = K_Y \begin{bmatrix} X_i \\ X_{i+1} \end{bmatrix} \quad (2.80)$$

for the case of complex eigenvalues, it is apparent that small variations in  $X_i$  will cause correspondingly small variations in the eigenvector assignment. A matrix  $X$  is defined as

$$X = [X_1, \dots, X_n]. \quad (2.81)$$

Since this matrix designates which eigenvectors are assigned, it is called the designator matrix. A gradient matrix  $[GR]$  with elements  $[GR]_{ij}$  is defined to be

$$[\text{GR}]_{ij} = \frac{\frac{\partial J}{\partial X_{ij}}}{\left\| \frac{\partial J}{\partial X_{ij}} \right\|} . \quad (2.82)$$

The designator matrix  $X$  is then varied according to the rule

$$X_{ij}(q+1) = X_{ij}(q) - d[\text{GR}]_{ij} \quad (2.83)$$

where  $d$  denotes the step size of  $X_{ij}$  during each iteration. The gradient search is continued until a satisfactory compromise between the reduction in the value of the cost function and the modification of the time response is achieved.

## CHAPTER 3

### SPECTRAL ASSIGNMENT USING REDUCED-ORDER MODELS AND REDUCED-ORDER OBSERVERS

In this chapter, the use of reduced-order models and reduced-order observers in the design of feedback controllers is investigated. A reduced-order model of a known system is formulated. It is then used to design a constant full state feedback matrix for the original full-order system. It is shown that the eigenvalues and eigenvectors reassigned in the reduced-order model are reassigned in the full-order system while those not included in the reduced-order model remain unchanged. The constant state feedback matrix is then implemented by output feedback using a Luenberger [2] reduced-order observer. It is shown that the eigenvalue and eigenvector assignments in the full-order system remain unchanged when a reduced-order observer is used.

#### 3.1 Motivation for Using Reduced-Order Models and Reduced-Order Observers

Models representing the behavior of physical systems often consist of a very large number of coupled linear differential equations. Such models are difficult to use when designing control systems due to excessive requirements for computer time and memory, and to the numerical analysis problems inherently present when dealing with large systems of

equations. It is, therefore, desirable to develop a design procedure which utilizes reduced-order system models.

The spectral assignment synthesis methods described in Chapter 2 use full state feedback. However, full state feedback is not feasible for most systems because there are often fewer outputs than system states. Full state feedback is implemented by the use of a dynamic observer for these systems. The use of a full system observer is unnecessary since some states usually are available for measurement and therefore need not be estimated. A reduced-order observer is therefore desirable in order to minimize the complexity of the control system.

This chapter develops a reduced-order model and reduced-order observer. Control system design for the full-order system is accomplished using the reduced-order observer. Reduced-order models and observers have been used for several years. However, it is shown here that the eigenvalues and eigenvectors assigned using the reduced-order model are retained in the closed loop full-order system while the eigenvalues and eigenvectors not included in the reduced-order model remain unchanged in the closed loop full-order system.

### 3.2 Reduced-Order Model Formulation

A reduced-order model that is used in the design of a constant state feedback controller for the full-order system model is formulated in this section. The reduced-order model contains the eigenvalues that are to be reassigned in the full-order model. Let the original system model be described by the state equations

$$\dot{x} = Ax + Bu \quad (3.1)$$

and

$$y = Cx \quad (3.2)$$

where  $x \in \mathbb{R}^n$ ,  $u \in \mathbb{R}^m$ , and  $y \in \mathbb{R}^p$ . It is assumed that the eigenvalues of  $A$  are distinct and are denoted by  $(\lambda_1, \dots, \lambda_n)$ . The corresponding modal matrix for  $A$  is denoted by  $V = [v_1, \dots, v_n]$  where  $v_i$  denotes the eigenvector corresponding to  $\lambda_i$ . The system model is transformed by defining a new state variable

$$z = V^{-1}x. \quad (3.3)$$

Equation (3.1) is transformed to give

$$\dot{z} = \Lambda z + \hat{B}u \quad (3.4)$$

where  $\Lambda = V^{-1}AV = \text{diag}(\lambda_1, \dots, \lambda_n)$ , and  $\hat{B} = V^{-1}B$ . The system is now partitioned to separate the eigenvalues to be reassigned in the reduced-order model from those that will remain unchanged in the full-order system model. Thus,

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} \Lambda_1 & \vdots & 0 \\ \text{---} & \text{---} & \text{---} \\ 0 & \vdots & \Lambda_2 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u \quad (3.5)$$

where  $z_1 \in \mathbb{R}^k$  and  $z_2 \in \mathbb{R}^{n-k}$ . The eigenvalues included in the reduced-order model must be contained in  $\Lambda_1$  and those which are not included must be contained in  $\Lambda_2$ . The reduced-order model is thus described by the state equation

$$\dot{z}_1 = \Lambda_1 z_1 + \hat{B}_1 U. \quad (3.6)$$

The reduced-order model is then used in conjunction with the spectral assignment procedure to assign eigenvalues and partially assign eigenvectors in the full-order system model. However, the relationship between the eigenvalue and eigenvector assignments in the reduced-order and full-order models must be investigated first.

### 3.3 Spectral Assignment Using Reduced-Order Models

The reduced-order model is used to design a constant state feedback matrix for the full-order system. The relationship between the eigenvalues and eigenvectors of the closed-loop reduced-order model and the closed loop full-order system must be understood in order to accomplish this. The relationship between reduced-order and full-order system eigenvalues is determined first. Let  $F$  denote a constant state feedback matrix computed for the reduced-order model. The control law is then written as

$$u = F z_1. \quad (3.7)$$

The reduced-order model closed loop equation is therefore

$$\dot{z}_1 = (\Lambda_1 + \hat{B}_1 \hat{F}) z_1. \quad (3.8)$$

F is now implemented about the full-order system by assuming full state availability in the full order model and transforming  $\hat{F}$  back to the original coordinate system. Equation (3.7) is rewritten as

$$u = [\hat{F}:0] \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = F'z. \quad (3.9)$$

Substituting for  $z$  from equation (3.3) yields

$$u = F'V^{-1}x = \bar{F}x. \quad (3.10)$$

Hence, the closed-loop full order system is written as

$$\begin{aligned} \dot{x} &= Ax + B \bar{F}x \\ &= [A + B \bar{F}]x. \end{aligned} \quad (3.11)$$

The eigenvalues of the full order system are the eigenvalues of  $[A + B\bar{F}]$ . This matrix is rewritten as

$$\begin{aligned} [A + B\bar{F}] &= [V\Lambda V^{-1} + V\hat{B}F'V^{-1}] \\ &= V[\Lambda + \hat{B}F']V^{-1}. \end{aligned} \quad (3.12)$$

Since  $[\Lambda + B\bar{F}]$  and  $[\Lambda + BF']$  are related by a similarity transformation, they have the same eigenvalues. Matrix  $[\Lambda + \hat{B}F']$  is expanded as

$$\begin{aligned} [\Lambda + \hat{B}F'] &= \begin{bmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{bmatrix} + \begin{bmatrix} \hat{B}_1 \hat{F} & 0 \\ \hat{B}_2 \hat{F} & 0 \end{bmatrix} \\ &= \begin{bmatrix} \Lambda_1 + \hat{B}_1 \hat{F} & 0 \\ \hat{B}_2 \hat{F} & \Lambda_2 \end{bmatrix}. \end{aligned} \quad (3.13)$$

The eigenvalues of this matrix are obviously the eigenvalues of  $[\Lambda_1 + \hat{B}_1 \hat{F}]$  and  $[\Lambda_2]$ . Thus it is possible to reassign the  $k$  eigenvalues included in the reduced-order model without modifying the  $(n-k)$  original system eigenvalues which were not included in the reduced-order model.

The relationship between eigenvectors of the reduced-order and full order system models is determined next. The eigenvector equation for  $[\Lambda + BF']$  is written as

$$[\lambda_i I - \Lambda - \hat{B}F'] v_i = \begin{bmatrix} \lambda_i I - \Lambda_1 - \hat{B}_1 \hat{F} & 0 \\ -\hat{B}_2 \hat{F} & \lambda_i I - \Lambda_2 \end{bmatrix} \begin{bmatrix} \hat{v}_{1i} \\ \hat{v}_{2i} \end{bmatrix} = 0 \quad (3.14)$$

which yields the two equations

$$[\lambda_i I - \Lambda_1 - \hat{B}_1 \hat{F}] \hat{v}_{1i} = 0 \quad (3.15)$$

and

$$[-\hat{B}_2 \hat{F}] \hat{v}_{1i} + [\lambda_i I - \Lambda_2] \hat{v}_{2i} = 0. \quad (3.16)$$



If  $\lambda_i$  is an eigenvalue of  $[\Lambda_1 + \hat{B}_1 \hat{F}]$ , but not an eigenvalue of  $[\Lambda_2]$ , then from equation (3.15) it follows that  $\hat{v}_{1i}$  is an eigenvector of  $[\Lambda_1 + \hat{B}_1 \hat{F}]$ . Since  $\lambda_i$  is not an eigenvalue of  $[\Lambda_2]$ ,  $[\lambda_i I - \Lambda_2]$  is nonsingular. Thus, from equation (3.16)

$$\hat{v}_{2i} = [\lambda_i I - \Lambda_2]^{-1} \hat{B}_2 \hat{F} \hat{v}_{1i}. \quad (3.17)$$

Therefore,  $\hat{v}_i$  is written as

$$\hat{v}_i = \begin{bmatrix} \text{-----} I \text{-----} \\ [\lambda_i I - \Lambda_2]^{-1} \hat{B}_2 \hat{F} \end{bmatrix} \hat{v}_{1i}. \quad (3.18)$$

Equation (3.18) illustrates that the first  $k$  elements of  $\hat{v}_i$  can be assigned using the reduced-order model while the remaining  $(k-p)$  elements are linear combinations of  $\hat{v}_{1i}$ .

On the other hand, if  $\lambda_i$  is an eigenvalue of  $\Lambda_2$  and not an eigenvalue of  $[\Lambda_1 + \hat{B}_1 \hat{F}]$ , the matrix  $[\lambda_i I - \Lambda_1 - \hat{B}_1 \hat{F}]$  is nonsingular. Therefore, equation (3.15) is satisfied only if

$$\hat{v}_{1i} = 0. \quad (3.19)$$

From equation (3.16) it follows that  $\hat{v}_{2i}$  is an eigenvector of  $\Lambda_2$  for eigenvalue  $\lambda_i$ . Since  $\Lambda_2$  is a diagonal matrix,  $\hat{v}_{2i}$  is written as

$$\hat{v}_{2i} = [0, \dots, 0, k_i, 0, \dots, 0]^T \quad (3.20)$$

where  $k_i$  is a nonzero constant. Therefore  $\hat{v}_i$  is written as

$$\hat{v}_i = \begin{bmatrix} 0 \\ \hat{v}_{2i} \end{bmatrix}. \quad (3.21)$$

Let  $\bar{v}_i$  be the eigenvector of  $[A + B\bar{F}]$  corresponding to eigenvalue  $\lambda_i$ . The eigenvectors  $\bar{v}_i$  and  $v_i$  are related by the transformation

$$\bar{v}_i = V v_i. \quad (3.22)$$

Expanding equation (3.22) for eigenvalues of  $[\Lambda_1 + B_1F]$  yields

$$\begin{aligned} \begin{bmatrix} \bar{v}_{1i} \\ \bar{v}_{2i} \end{bmatrix} &= \begin{bmatrix} V_{11} & \vdots & V_{12} \\ \hline V_{21} & \vdots & V_{22} \end{bmatrix} \begin{bmatrix} I \\ [\lambda_i I - \Lambda_2]^{-1} \hat{B}_2 \hat{F} \end{bmatrix} \hat{v}_{1i} \\ &= \begin{bmatrix} V_{11} + V_{12} [\lambda_i I - \Lambda_2]^{-1} \hat{B}_2 \hat{F} \\ V_{21} \quad V_{22} \quad [\lambda_i I - \Lambda_2]^{-1} \hat{B}_2 \hat{F} \end{bmatrix} \hat{v}_{1i}. \end{aligned} \quad (3.23)$$

Often when the reduced-order model contains only the dominant modes of the system,  $V_{11} = V_{11} + V_{12} [\lambda_i I - \Lambda_2]^{-1} \hat{B}_2 \hat{F}$ . Then the top  $k$  components of the first  $k$  eigenvectors are assigned by choosing  $v_{1i}$  as

$$\hat{v}_{1i} = V_{11}^{-1} \bar{v}_{1i}. \quad (3.24)$$

When the above approximation does not apply, an initial assignment of  $\hat{v}_{1i}$  is made using equation (3.24),  $\hat{F}$  is calculated, and the error between the desired top  $k$  components of the first  $k$  eigenvectors and the actual assignment is calculated. A gradient search procedure is then used to reduce this error.

For  $\lambda_i$  which are eigenvalues of  $\Lambda_2$ , equation (3.22) expands to

$$\bar{v}_i = V\hat{v}_i = \begin{bmatrix} V_{12} \\ V_{22} \end{bmatrix} \hat{v}_{2i}. \quad (3.25)$$

Substituting for  $\hat{v}_{2i}$  from equation (3.20) yields

$$\bar{v}_i = k_i \begin{bmatrix} V_{12} \\ V_{22} \end{bmatrix}. \quad (3.26)$$

Therefore, the last  $(n-k)$  eigenvectors in the closed loop full order system are the original open-loop eigenvectors. Thus eigenvectors corresponding to the eigenvalues retained in the full order system model remain the same in the final closed loop system.

### 3.4 Reduced-Order Observer Formulation

In order to implement the feedback matrix  $\bar{F}$  calculated in equation (3.10), full state availability is required. However, if the number of outputs  $p$  is less than the number of states  $n$ , then  $(n-p)$  states must be estimated. This section parallels Luenberger's [2] development of a reduced-order observer system to allow the implementa-

tion of a full state feedback matrix in such a system. The following section shows that not only are the eigenvalues of the observer system retained in the closed loop system, but that the original eigenvector assignment is not affected by use of the observer.

The open loop full-order system model described by equations (3.1) and (3.2) is

$$\dot{x} = Ax + Bu \quad (3.26)$$

and

$$y = Cx. \quad (3.27)$$

If the feedback matrix  $\bar{F}$  is implemented about the system model, then

$$u = \bar{F}x. \quad (3.28)$$

It is assumed without any loss in generality that the first  $p$  columns of  $C$  are linearly independent. A transformation matrix  $M$  is defined to be

$$M = \begin{bmatrix} C_1 & C_2 \\ 0 & I_{n-p} \end{bmatrix} \quad (3.29)$$

where  $I_{n-p}$  is the  $(n-p)^{\text{th}}$  order identity matrix. A new state variable is defined by

$$\tilde{z} = Mx. \quad (3.30)$$

Equations (3.26), (3.27), and (3.28) are transformed and written as

$$\dot{\tilde{z}} = \tilde{A} \tilde{z} + \tilde{B}u, \quad (3.31)$$

$$y = \tilde{C} \tilde{z}, \quad (3.32)$$

and

$$u = \tilde{F} \tilde{z} \quad (3.33)$$

where

$$\tilde{A} = MAM^{-1}, \quad (3.34)$$

$$\tilde{B} = MB, \quad (3.35)$$

$$\tilde{C} = CM^{-1} = [I_p \ : \ 0], \quad (3.36)$$

and

$$\tilde{F} = \bar{F}M^{-1}. \quad (3.37)$$

Equations (3.31) and (3.32) are expanded as

$$\begin{bmatrix} \dot{\tilde{z}}_1 \\ \dot{\tilde{z}}_2 \end{bmatrix} = \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}_{21} & \tilde{A}_{22} \end{bmatrix} \begin{bmatrix} \tilde{z}_1 \\ \tilde{z}_2 \end{bmatrix} + \begin{bmatrix} \tilde{B}_1 \\ \tilde{B}_2 \end{bmatrix} u \quad (3.38)$$

$$y = [\tilde{C}_1 \quad \tilde{C}_2] \begin{bmatrix} \tilde{z}_1 \\ \tilde{z}_2 \end{bmatrix} = \tilde{z}_1. \quad (3.39)$$

Therefore the  $p$  components of the output vector are the first  $p$  states of the transformed system denoted by  $\tilde{z}_1$ . The reduced-order observer must then estimate the remaining  $(n-p)$  states denoted by  $\tilde{z}_2$ . Equation (3.38) is expanded to be

$$\dot{\tilde{z}}_1 = \tilde{A}_{11} \tilde{z}_1 + \tilde{A}_{12} \tilde{z}_2 + \tilde{B}_1 u \quad (3.40)$$

and

$$\dot{\tilde{z}}_2 = \tilde{A}_{21} \tilde{z}_1 + \tilde{A}_{22} \tilde{z}_2 + \tilde{B}_2 u. \quad (3.41)$$

Since  $\tilde{z}_1$  is available as the output vector  $y$ , it can be differentiated to generate  $\dot{\tilde{z}}_1$ . Hence, equation (3.40) is solved for  $\tilde{A}_{12} \tilde{z}_2$

which is used as an input to a reduced-order observer to approximate  $\tilde{z}_2$ . The proposed observer is shown in Figure 3.1. It is desired that

$$\theta = \tilde{z}_2 \quad (3.42)$$

in order to implement the feedback matrix  $\tilde{F}$ . The observer state equation is written as

$$\dot{\theta} = E\theta + L\dot{\tilde{z}}_1 - L\tilde{A}_{11}\tilde{z}_1 + [\Omega - L\tilde{B}_1]u. \quad (3.43)$$

The need to differentiate  $\tilde{z}_1$  is avoided by redrawing the observer as shown in Figure 3.2. If  $w$  is defined by

$$w = \theta - L\tilde{z}_1, \quad (3.44)$$

then

$$\dot{w} = L\dot{\tilde{z}} - \dot{\theta} = 0. \quad (3.45)$$

Using equation (3.42) to substitute for  $\dot{\theta}$  in equation (3.45) yields

$$\dot{w} + L\dot{\tilde{z}}_1 - \dot{\tilde{z}}_2 = 0. \quad (3.46)$$

Calculating each term of the above equation results in the three equa-

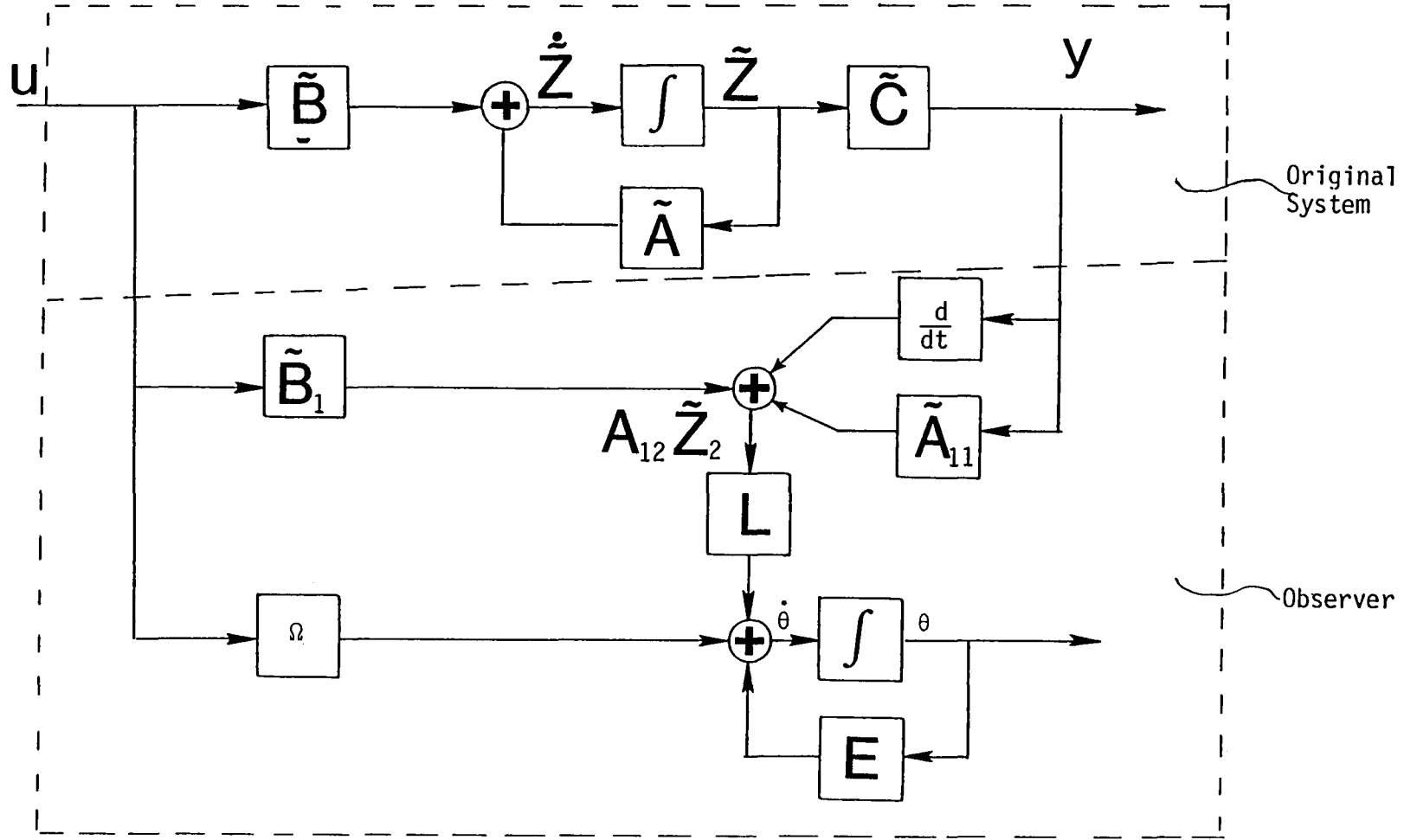


Figure 3.1. Original System and Proposed Reduced-Order Observer



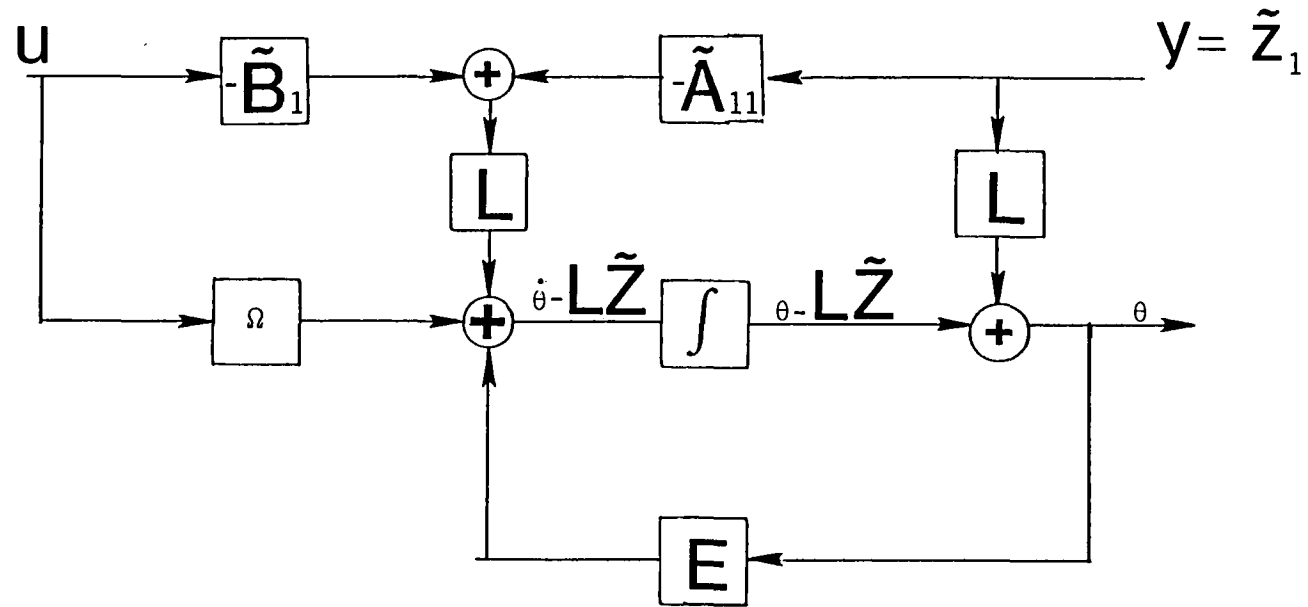


Figure 3.2. Proposed Reduced-Order Observer without Differentiator

tions,

$$\dot{w} = E [w + L\tilde{z}_1] + [\Omega - L\tilde{B}_1]u - L\tilde{A}_{11}\tilde{z}_1, \quad (3.47)$$

$$L\dot{\tilde{z}}_1 = L\tilde{A}_{11}\tilde{z}_1 + L\tilde{A}_{12}\tilde{z}_2 + L\tilde{B}_1 u, \quad (3.48)$$

and

$$-\dot{\tilde{z}} = -\tilde{A}_{21}\tilde{z}_1 - \tilde{A}_{22}\tilde{z}_2 - \tilde{B}_2 u. \quad (3.49)$$

Substituting equations (3.47), (3.48), and (3.49) into (3.46) gives

$$Ew + [EL - \tilde{A}_{21}] \tilde{z}_1 + [L\tilde{A}_{12} - \tilde{A}_{22}] \tilde{z}_2 + [\Omega - \tilde{B}_2] u = 0. \quad (3.50)$$

Using equations (3.42) and (3.44) to substitute for  $w$  yields

$$[\tilde{A}_{21}] \tilde{z}_1 + [E - \tilde{A}_{22} + L\tilde{A}_{12}] \tilde{z}_2 + [\Omega - \tilde{B}_2] u = 0. \quad (3.51)$$

Since the input and state vectors are not generally zero, the multiplying matrices must all be equal to zero for the equation to be true.

Solving for the last two terms gives

$$E = \tilde{A}_{22} - L\tilde{A}_{12} \quad (3.52)$$

and

$$\Omega = \tilde{B}_2. \quad (3.53)$$

Matrix  $\tilde{A}_{21}$  is generally nonzero also. This indicates that the proposed reduced-order observer is not adequate. Thus the observer is modified by adding  $\tilde{A}_{21} \tilde{z}_1$  to  $\dot{w}$  and grouping terms as shown in Figure 3.3. To remove the summer located after the integrator, it is noted that

$$EL \tilde{z}_1 = [(\tilde{A}_{22} - L\tilde{A}_{12})L] \tilde{z}_1. \quad (3.54)$$

Using equation (3.54), the reduced-order observer is drawn as Figure 3.4. The eigenvalues and eigenvectors of  $E = [\tilde{A}_{22} - L\tilde{A}_{12}]$  are determined by proper selection of  $L$  since the eigenvalues of  $[\tilde{A}_{22} - L\tilde{A}_{12}]$  are also the eigenvalues of  $[\tilde{A}_{22}^T - A_{12}^T L^T]$ . Chapter 2 describes a procedure for selecting a proper  $L^T$  to achieve a desired eigenvalue and eigenvector assignment. A guideline for selecting reduced-order observer eigenvalue locations is discussed in Chapter 4.

The reduced-order observer is now used to implement the feedback matrix  $\tilde{F}$ . Expanding the control law given by equation (3.33) gives

$$u = [\tilde{F}_1 : \tilde{F}_2] \begin{bmatrix} \tilde{z}_1 \\ \tilde{z}_2 \end{bmatrix} = F_1 z_1 + F_2 z_2. \quad (3.55)$$

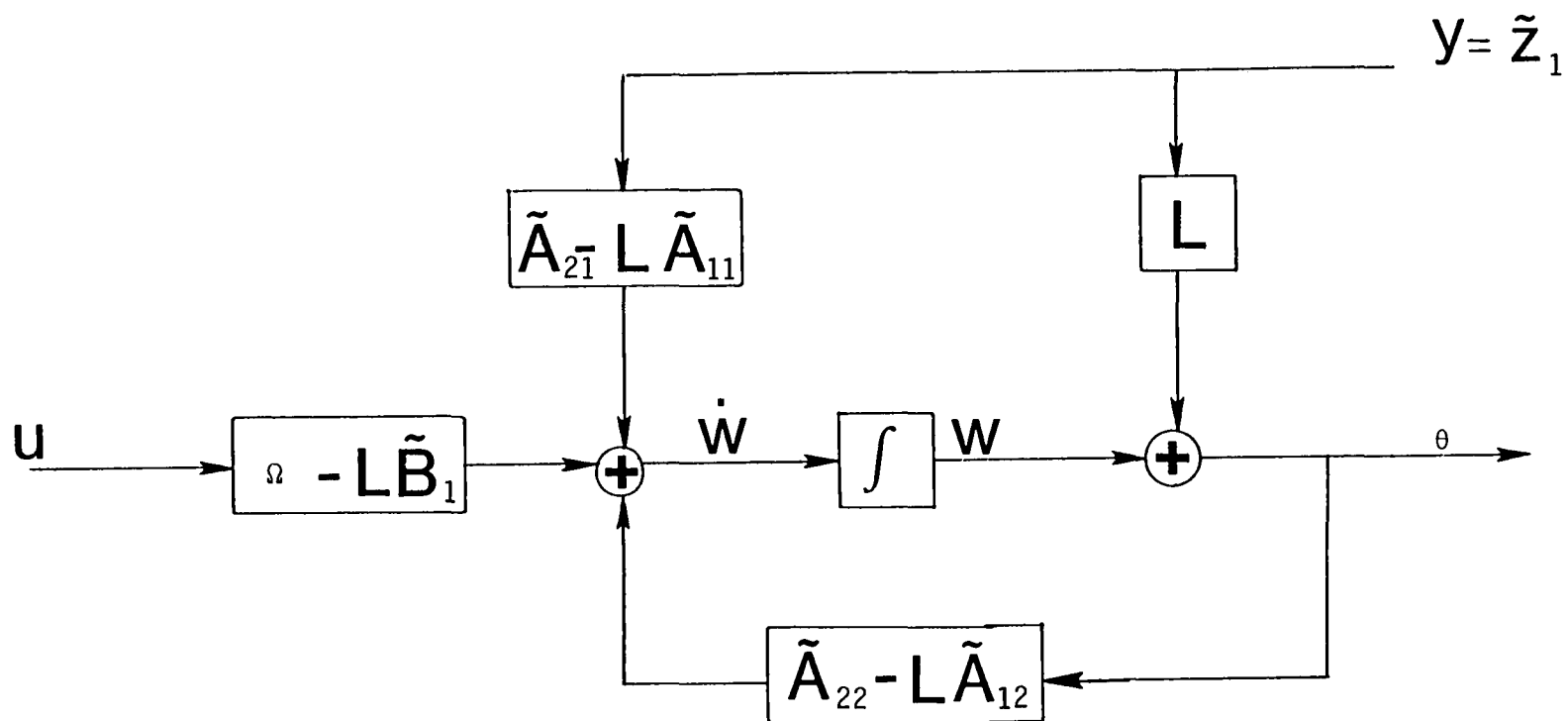


Figure 3.3. Modified Reduced-Order Observer

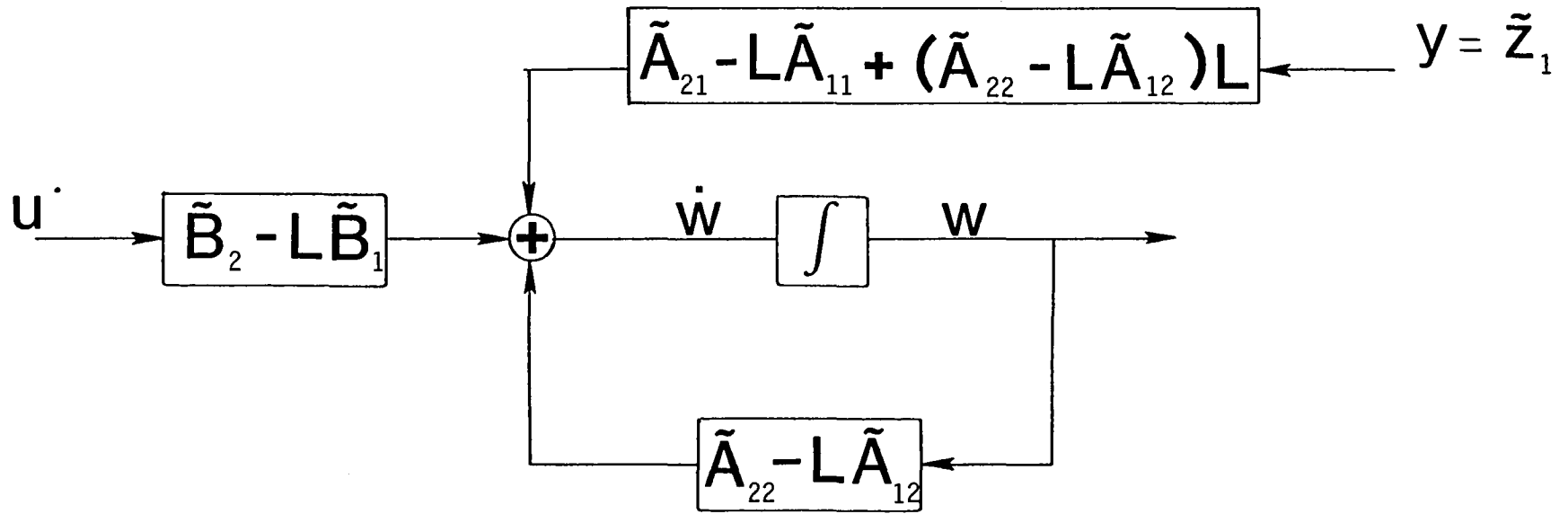


Figure 3.4. Reduced-Order Observer

Using equations (3.42) and (3.44), a substitution is made for  $\tilde{z}_2$  resulting in

$$u = [\tilde{F}_1 + \tilde{F}_2 L] \tilde{z}_1 + [\tilde{F}_2] w. \quad (3.56)$$

The control law in equation (3.56) is implemented in Figure 3.5.

### 3.5 Effect of Reduced-Order Observer on Full System Eigenstructure

The effect of a reduced-order observer on the eigenvalue and eigenvector assignment in the closed loop full-order system is developed in this section. Luenberger [2] has proven that the eigenvalues of the original system assignment and the observer assignment remain unchanged in the closed loop full-order system. It is shown here that the eigenvector assignment also remains unchanged. The following definitions for matrices  $G$ ,  $R$ , and  $T$  are given to reduce the algebraic complexity of this development. Let

$$G = [(\tilde{A}_{21} - L\tilde{A}_{11}) + (\tilde{A}_{22} - L\tilde{A}_{12})L], \quad (3.57)$$

$$R = [\tilde{F}_1 + \tilde{F}_2 L] \quad (3.58)$$

and

$$T = [-L : I]. \quad (3.59)$$

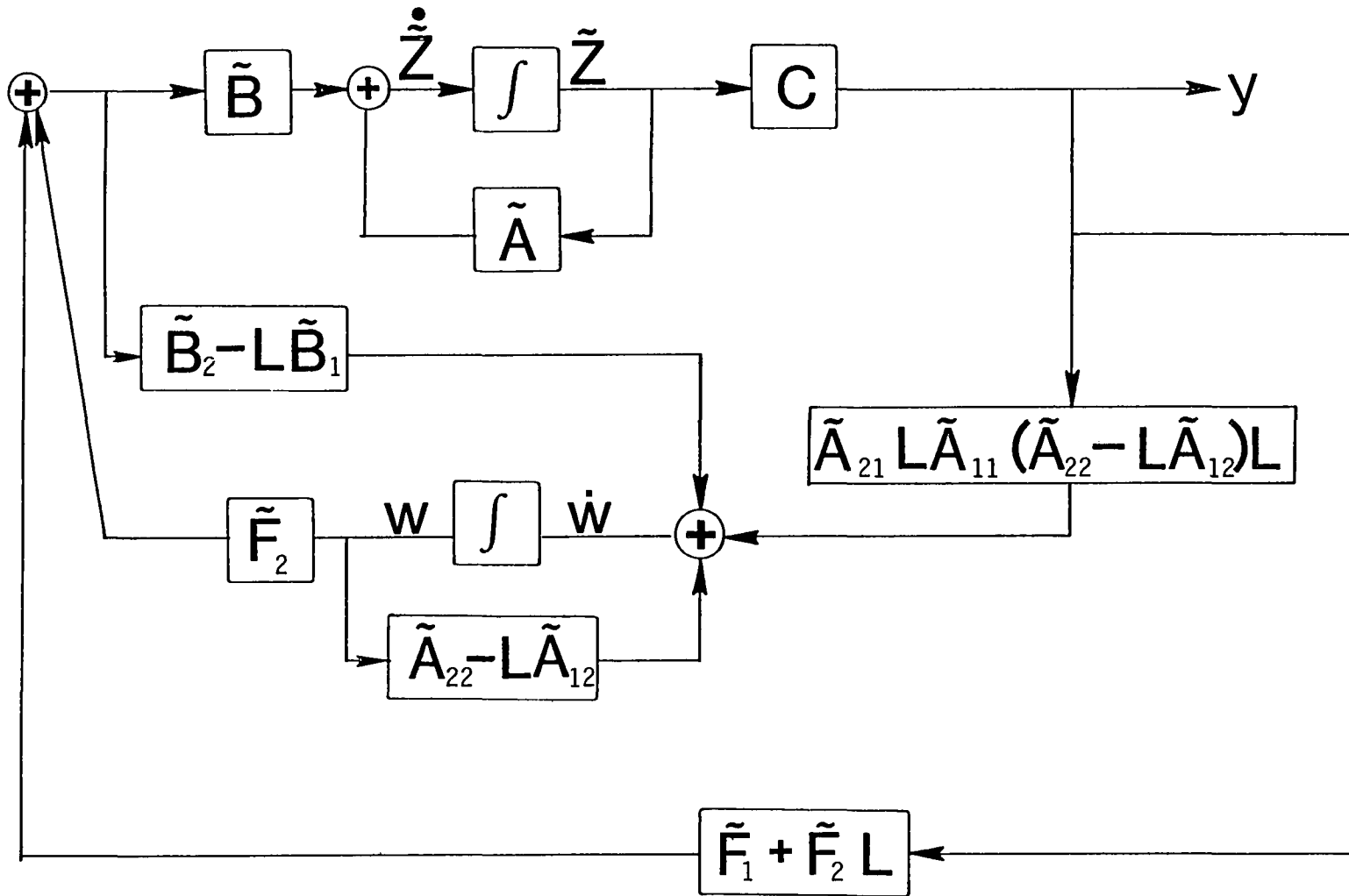


Figure 3.5. Control Law Implemented with Reduced-Order Observer

The eigenvalues of the system are determined first. Using equations (3.57), (3.58), and (3.59), the total system state equation is written as

$$\begin{bmatrix} \dot{\tilde{z}} \\ \tilde{z} \\ \dot{w} \\ w \end{bmatrix} = \begin{bmatrix} \tilde{A} + \tilde{B}\tilde{R}\tilde{C} & \tilde{B}\tilde{F}_2 \\ \tilde{T}\tilde{B}\tilde{R}\tilde{C} + \tilde{G}\tilde{C} & E + \tilde{T}\tilde{B}\tilde{F}_2 \end{bmatrix} \begin{bmatrix} \tilde{z} \\ w \end{bmatrix}. \quad (3.60)$$

Equation (3.60) is now transformed to an upper triangular form so that the system eigenvalues are apparent. The transformation matrix  $P$  is defined by

$$P = \begin{bmatrix} I & 0 \\ -T & I \end{bmatrix}, \quad (3.61)$$

and a new state variable is defined to be

$$v = w - T\tilde{z} \quad (3.62)$$

so that

$$\begin{bmatrix} \dot{\tilde{z}} \\ \tilde{z} \\ \dot{v} \\ v \end{bmatrix} = \begin{bmatrix} I & 0 \\ -T & I \end{bmatrix} \begin{bmatrix} \tilde{A} + \tilde{B}\tilde{R}\tilde{C} & \tilde{B}\tilde{F}_2 \\ \tilde{T}\tilde{B}\tilde{R}\tilde{C} + \tilde{G}\tilde{C} & E + \tilde{T}\tilde{B}\tilde{F}_2 \end{bmatrix} \begin{bmatrix} T & 0 \\ T & I \end{bmatrix} \begin{bmatrix} \tilde{z} \\ v \end{bmatrix}. \quad (3.63)$$

Equation (3.63) is simplified to give



$$\begin{bmatrix} \dot{\tilde{z}} \\ \dot{\tilde{v}} \end{bmatrix} = \begin{bmatrix} \tilde{A} + \tilde{B} (R\tilde{C} + \tilde{F} T) & \tilde{B}\tilde{F}_2 \\ -TA + G\tilde{C} + ET & E \end{bmatrix} \begin{bmatrix} \tilde{z} \\ \tilde{v} \end{bmatrix}. \quad (3.64)$$

The submatrix  $[-TA + G\tilde{C} + ET]$  is equal to zero. This is shown by evaluating each term within the matrix. The individual terms are given by

$$-T\tilde{A} = [L\tilde{A}_{11} - \tilde{A}_{21} \quad ; \quad L\tilde{A}_{12} - \tilde{A}_{22}], \quad (3.65)$$

$$G\tilde{C} = G[I:0] = [\tilde{A}_{21} - L\tilde{A}_{11} + (\tilde{A}_{22} - L\tilde{A}_{12})L \quad ; \quad 0], \quad (3.66)$$

and

$$ET = [-(\tilde{A}_{22} - L\tilde{A}_{12})L \quad ; \quad -L\tilde{A}_{12} + \tilde{A}_{22}]. \quad (3.67)$$

Hence,

$$-T\tilde{A} + G\tilde{C} + ET = 0. \quad (3.68)$$

The expression  $(R\tilde{C} + \tilde{F}_2T)$  is equivalent to  $\tilde{F}$ . This is shown by expanding  $(R\tilde{C} + \tilde{F}_2T)$  as

$$\begin{aligned} R\tilde{C} + \tilde{F}_2T &= [\tilde{F}_1 + \tilde{F}_2L] [I \quad ; \quad 0] + \tilde{F}_2 [-L \quad ; \quad I] \\ &= [\tilde{F}_1 \quad ; \quad \tilde{F}_2] = \tilde{F}. \end{aligned} \quad (3.69)$$

Therefore the substitution of equations (3.68) and (3.69) into (3.63) gives

$$\begin{bmatrix} \dot{\tilde{z}} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} \tilde{A} + \tilde{B}\tilde{F} & \tilde{B}\tilde{F}_2 \\ 0 & E \end{bmatrix} \begin{bmatrix} \tilde{z} \\ v \end{bmatrix}. \quad (3.70)$$

Thus the eigenvalues of the system are those assigned to  $[\tilde{A} + \tilde{B}\tilde{F}]$  and  $[E]$ . In other words, the use of the reduced-order observer has no effect on the original system eigenvalue assignment.

The eigenvectors of the system are determined next. Transforming the state equation using equation (3.30) results in

$$\begin{aligned} \begin{bmatrix} \dot{x} \\ \dot{v} \end{bmatrix} &= \begin{bmatrix} M^{-1} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \tilde{A} + \tilde{B}\tilde{F} & \tilde{B}\tilde{F}_2 \\ 0 & E \end{bmatrix} \begin{bmatrix} M & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix} \\ &= \begin{bmatrix} A + B\tilde{F} & B\tilde{F}_2 \\ 0 & E \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix}. \end{aligned} \quad (3.71)$$

Let the closed loop system matrix in equation (3.70) be denoted by  $A_T$ . An eigenvector of  $A_T$  corresponding to  $\lambda$  is denoted by  $v_T$ . Eigenvector  $v_T$  is compatibly partitioned so that

$$v_T = \begin{bmatrix} v_{T1} \\ v_{T2} \end{bmatrix} \quad (3.72)$$

The eigenvector equation for  $A_T$  is

$$\begin{bmatrix} \lambda I - A_T & -B\tilde{F}_2 \\ 0 & \lambda I - E \end{bmatrix} \begin{bmatrix} v_{T1} \\ v_{T2} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \quad (3.73)$$

Equation (3.73) is expanded giving

$$[\lambda I - A_T] v_{T1} - [B\tilde{F}_2] v_{T2} = 0 \quad (3.74)$$

and

$$[\lambda I - E] v_{T2} = 0. \quad (3.75)$$

It is assumed that the eigenvalues for  $[A_T]$  and  $[E]$  are distinct since their locations are arbitrarily assigned as discussed in Chapter 2.

Suppose  $\lambda$  is an eigenvalue of  $[A_T]$  but not  $[E]$ . Then  $[\lambda I - E]^{-1}$  exists. Premultiplying equation (3.75) by  $[\lambda I - E]^{-1}$  yields the result

$$v_{T2} = 0. \quad (3.76)$$

Hence equation (3.74) is simplified to

$$[\lambda I - A_T] v_{T1} = 0. \quad (3.77)$$

Therefore  $v_{T1}$  is an eigenvector of  $[A_T]$ .

Now suppose  $\lambda$  is an eigenvalue of  $[E]$  but not  $[A_T]$ . Then  $[\lambda I - A_T]^{-1}$  exists. Equation (3.75) implies that  $v_{T2}$  must be an eigenvector of  $[E]$ . Premultiplying equation (3.74) by  $[\lambda I - A_T]^{-1}$  and rearranging terms gives

$$v_{T1} = [\lambda I - A_T]^{-1} [B\tilde{F}_2] v_{T2}. \quad (3.78)$$

Hence for  $\lambda$  that are eigenvalues of  $[A_T]$ ,

$$v_T = \begin{bmatrix} v_{T1} \\ 0 \end{bmatrix} \quad (3.79)$$

where  $v_{T1}$  is an eigenvector of  $[A_T]$ . For  $\lambda$  that are eigenvalues of  $[E]$ ,

$$v_T = \begin{bmatrix} (\lambda I - A_T) & B \tilde{F}_2 \\ & I \end{bmatrix} v_{T2} \quad (3.80)$$

where  $v_{T2}$  is an eigenvector of  $[E]$ . Thus it is shown that a reduced-order model can be used to design a constant state feedback controller for a full-order system. The eigenvalues and eigenvectors assigned using the reduced-order model are retained in the full system while the

eigenvalues and eigenvectors not included in the reduced-order model remain unchanged in the full-order system. It is also shown that a reduced-order observer can be used to implement a full state feedback design without affecting the eigenvalues and eigenvectors of that design.

## CHAPTER 4

### DESIGN PROCEDURE

The methods described in Chapters 2 and 3 are the basis for developing a design philosophy, and then a corresponding design procedure, for constant state feedback controller design. The design procedure presented in this chapter is most useful when a designer is able to characterize the desired system in terms of the closed loop eigenvalues and eigenvectors as well as the time response. This chapter reviews an existing spectral assignment design philosophy. A corresponding design procedure and computer aided design package [11] are discussed next. Then an extension of the design philosophy is presented followed by a corresponding design procedure. This design procedure is included as a supplement to the computer aided design package. Lastly, the significant portions of the additional computer aided design software are described in detail. The modified design procedure uses reduced-order models and reduced-order observers with spectral assignment methods to reassign selected eigenvalues and eigenvectors in the full-order system model.

#### 4.1 Design Philosophy for Full-Order System Models

The constant state feedback design philosophy for full-order system

models is illustrated in Figure 4.1. The objectives faced by a system designer are often many and sometimes conflicting in nature. However, the location of eigenvalues and eigenvectors, and the system time response are generally the prime consideration. After these objectives are satisfactorily achieved, secondary design objectives are considered. These secondary objectives include feedback gain reduction, minimization of closed-loop system sensitivity to modeling errors or parameter variations, and noise suppression. The spectral assignment design procedure achieves a satisfactory control design by selecting an appropriate set of eigenvalues and approximating a desired set of corresponding eigenvectors. The eigenvalues determine the rates of decay of the various system modes while the eigenvectors determine the relative contribution of each mode to the different system states and outputs. After a satisfactory time response is achieved with an initial eigenvalue and eigenvector assignment, the secondary design objectives are considered. The freedom available to select the eigenvectors often provides a means to drastically improve these secondary design objectives while only slightly modifying the initial eigenvector assignment and thus the time response. This improvement is accomplished by modifying the eigenvectors within an area local to the original assignment. The direction and magnitude of the eigenvector modification is determined by a gradient search procedure as discussed in Section 2.6.

#### 4.2 Design Procedure for Full-Order System Models

A computer aided design package written by Marefat [11] currently exists and is illustrated in Figure 4.2. The software package consists

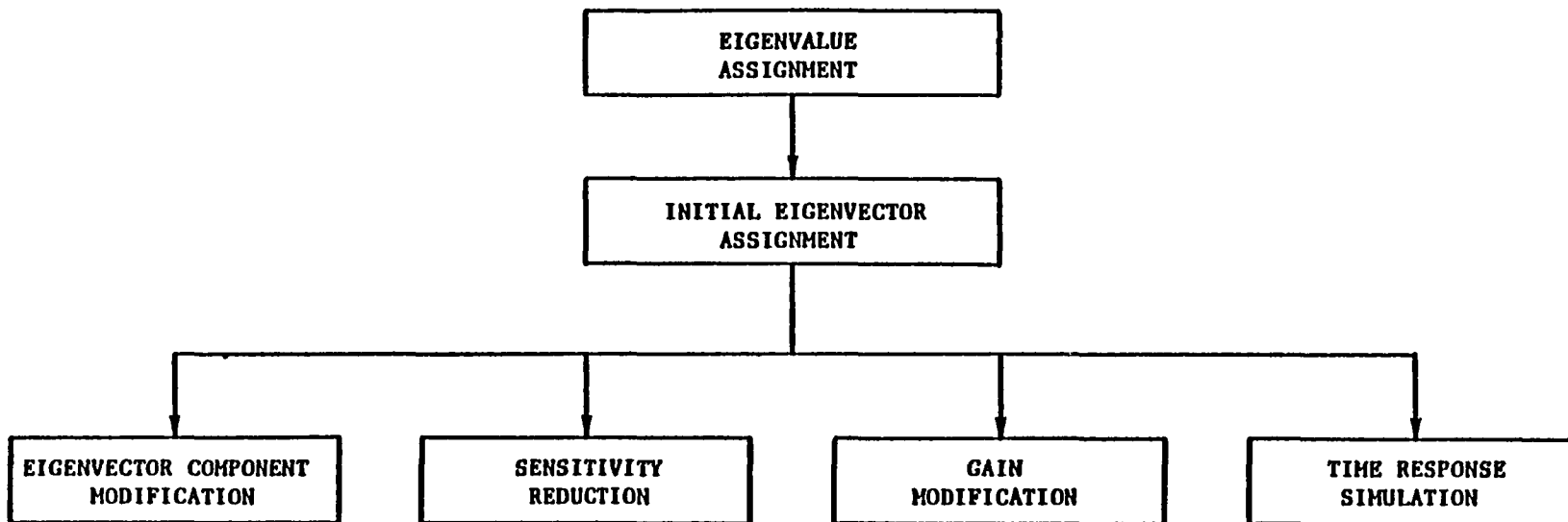


Figure 4.1. Eigenvalue/Eigenvector Assignment Design Philosophy



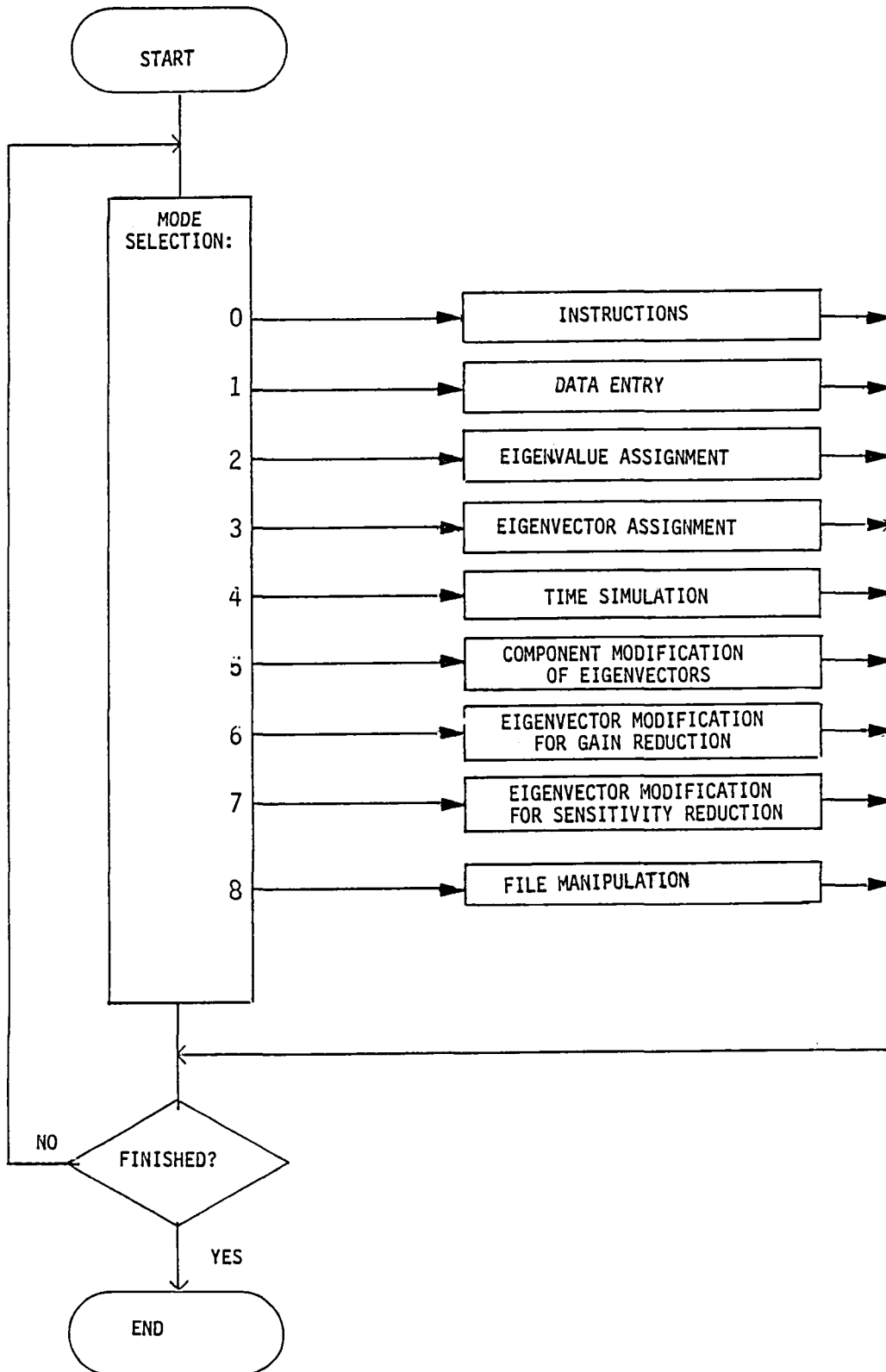


Figure 4.2. Spectral Assignment Computer Software Package Organization

of several special purpose subroutines that are accessed by the main control program. The subroutines may be entered in any order to implement specific design objectives according to the design philosophy in Figure 4.1. The system description (i.e., A, B, C) is entered in Mode 1. An arbitrary set of eigenvalues is assigned in Mode 2, which then formulates the allowable subspaces for the eigenvectors. The desired eigenvectors are approximated in Mode 3 by projecting them into the allowable eigenvector subspaces. Mode 4 allows the system designer to observe the time response for various initial conditions and system inputs. The initial design is then improved by alternating between Modes 4, 5, 6, and 7 until a compromise between primary and secondary design objectives is achieved. Modes 5, 6, and 7 modify eigenvector components, reduce feedback gain, and reduce system eigen-sensitivity, respectively, using gradient search procedures.

#### 4.3 Design Philosophy for Reduced-Order System Models and Observers

The design procedure discussed in the preceding section is useful for systems where full state feedback is feasible. Another feature of the procedure is that it assigns all eigenvalue and eigenvector locations. A system designer is often satisfied with several open loop eigenvalue and eigenvector locations in a large system. The reassignment of the remaining eigenvalues and eigenvectors is better accomplished using a reduced-order model that contains only those eigenvalues, due to reduced requirements for computer time and memory. Also, large systems typically have fewer independent outputs than states. A full state

feedback design is implemented in this case with an observer system. The observer estimates the system states in order to implement the feedback control law. Since some of the states can be obtained from the outputs, only the remaining states need to be estimated with an observer. A reduced-order observer is desirable in this case. It is designed using less computer resources than a full-order observer. Also, less hardware is required for implementation of a reduced-order observer.

A design philosophy that uses reduced-order models and reduced-order observers is illustrated in Figure 4.3. In order to design a control system using spectral assignment with reduced-order models and reduced-order observers, a designer must have knowledge of a desired set of system eigenvalues and eigenvectors. The original open loop eigenvalues and eigenvectors are compared with the desired eigenvalues and eigenvectors. A decision is made as to which of the eigenvalues and associated eigenvectors are satisfactory and which need to be reassigned. The spectral assignment design procedure is used to assign the desired eigenvalues and approximate the desired partial eigenvector assignment using the reduced-order model. Error between the initial eigenvector assignment and the desired eigenvector assignment is then reduced by a gradient search. Next, the resultant reduced-order feedback matrix is transformed to the full-order system.

If all of the states are simultaneously available for measurement, then the full state feedback matrix is implemented. However, if some states are not available, then a reduced-order observer is designed. The eigenvalues of the observer are assigned to be slightly more

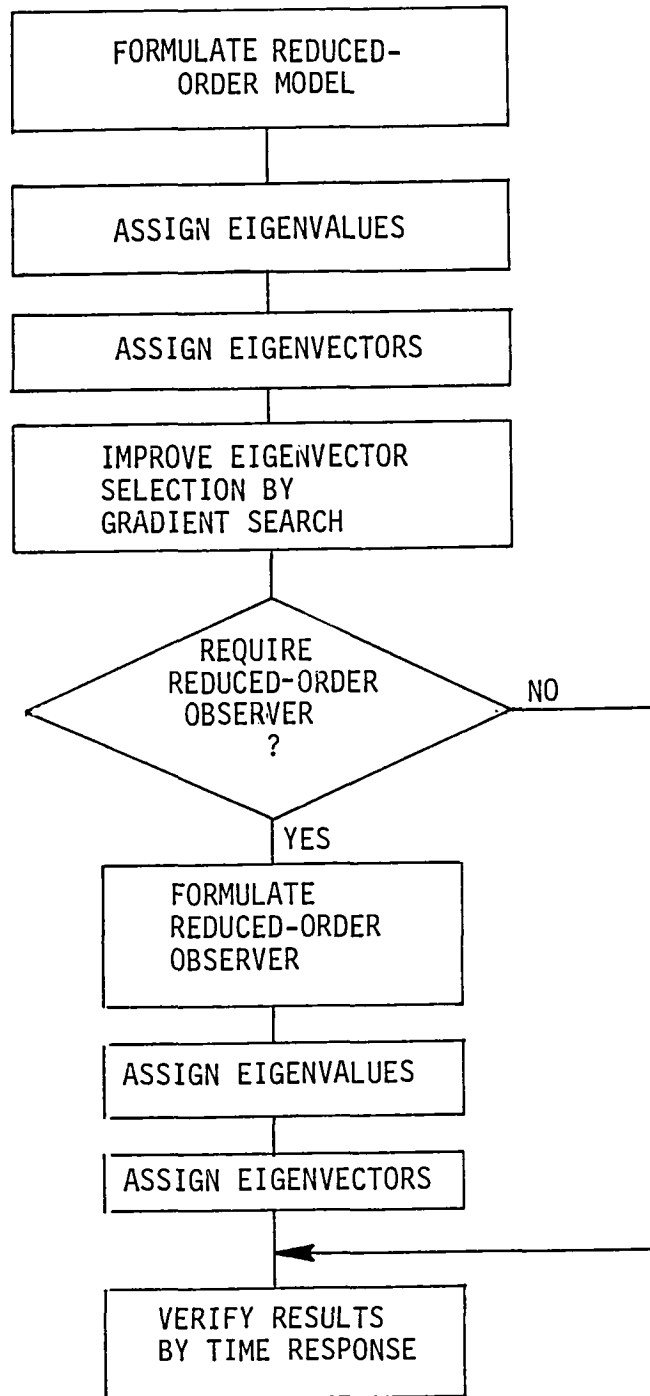


Figure 4.3. Reduced-Order Design Philosophy

negative than the dominant eigenvalues of the closed loop system design. This is done to ensure that the observer can respond quickly enough to follow the states being estimated. Theoretically if the observer eigenvalues are assigned to be very large negative numbers then the observer will provide a better estimate of the states. However, this is not done in practice because the observer then acts like a differentiator and is very susceptible to noise.

This philosophy and the synthesis methods described in Chapter 3 are used to develop an extended design procedure that exactly reassigns an arbitrary subset of the original system eigenvalues which are included in a reduced-order model. A partial eigenvector assignment is then approximated for these eigenvalues. This control is implemented with a reduced-order observer if there are fewer system outputs than states. A contribution of this thesis is that the reduced-order design and implementation are accomplished with the knowledge that the eigenvalues and eigenvectors not included in the reduced-order model remain unchanged.

#### 4.4 Design Procedure for Reduced-Order System Models and Observers

The computer aided design package written by Marefat has been modified as illustrated in Figure 4.4. An additional mode (Mode 9) has been added to incorporate the use of reduced-order models and reduced-order observers in system design. The full-order system description is entered in Mode 1. If a reduced-order model is to be used in the control system design, Mode 9 is entered. Otherwise the design procedure con-

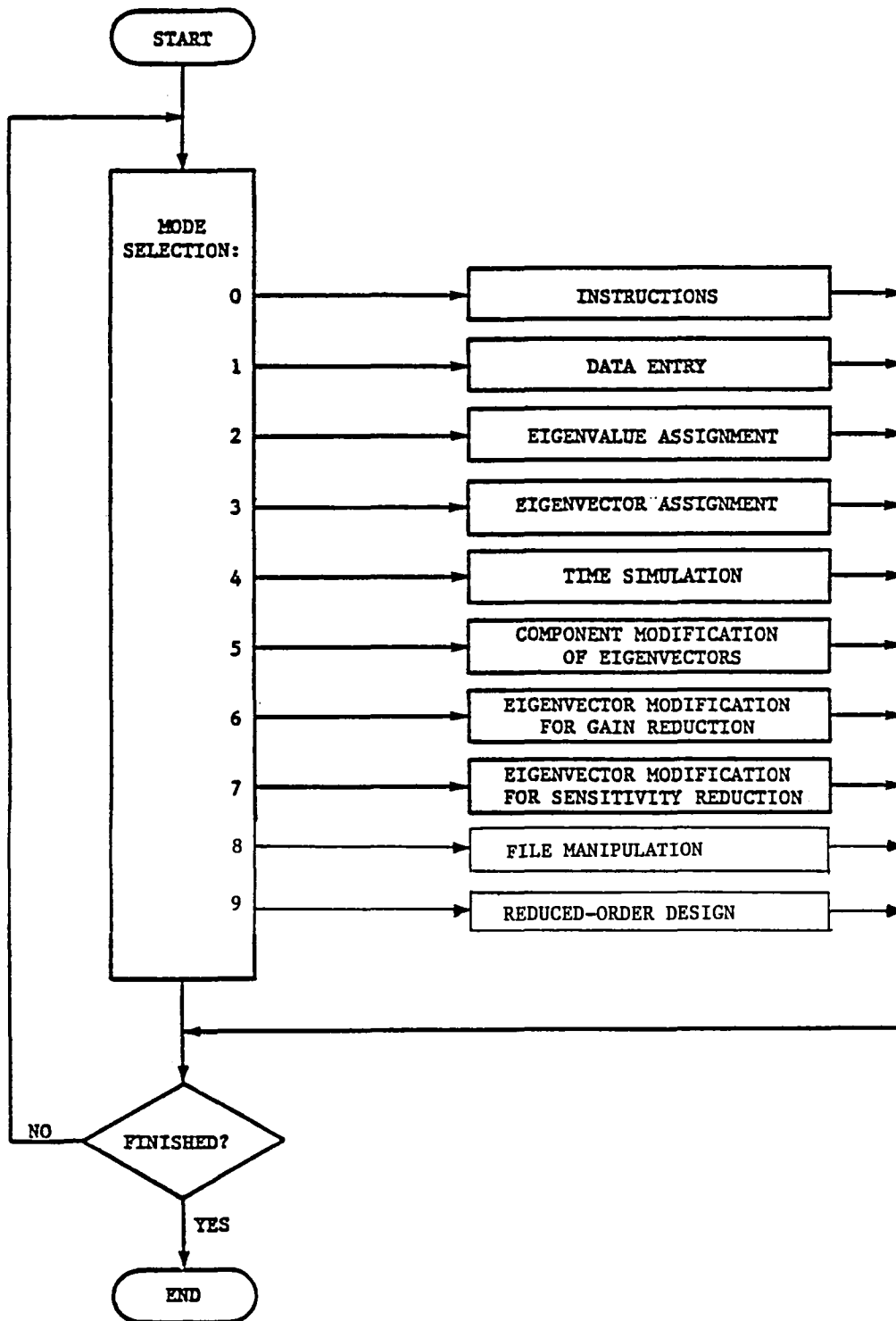


Figure 4.4. Modified Spectral Assignment Computer Software Package Organization.

tinues as described in Section 4.2.

A flowchart for Mode 9 is shown in Figure 4.5. Several of the existing program subroutines including Modes 2, 3, 4, and 6 are called from Mode 9. Three new subroutines are also called from within Mode 9. These subroutines are described in Sections 4.5 and 4.6.

After Mode 9 is entered the reduced-order model is formulated. Mode 2 is then called automatically and the reduced-order model eigenvalues are assigned. The designer is now prompted to enter the desired partial eigenvector assignment for the full-order system. An initial eigenvector assignment is calculated from the reduced-order model using equation (3.24) and the assignment is approximated using Mode 3. A gradient search is then initiated in order to decrease the error between the desired and actual partial eigenvector assignment. Upon completion of the gradient search, the reduced-order model feedback matrix is calculated and transformed to the full-order system coordinates. A reduced-order observer is formulated next. Eigenvalues and eigenvectors are assigned to the observer using Modes 2 and 3. Finally a time response is calculated and displayed for the combined system using Mode 4. If the designer is not satisfied with the time response Mode 9 is re-entered.

Two portions of the above design procedure required an extensive programming effort. Calculation of the cost function used in the gradient search is described in Section 4.5 and the gradient matrix calculation is described in Section 4.6.

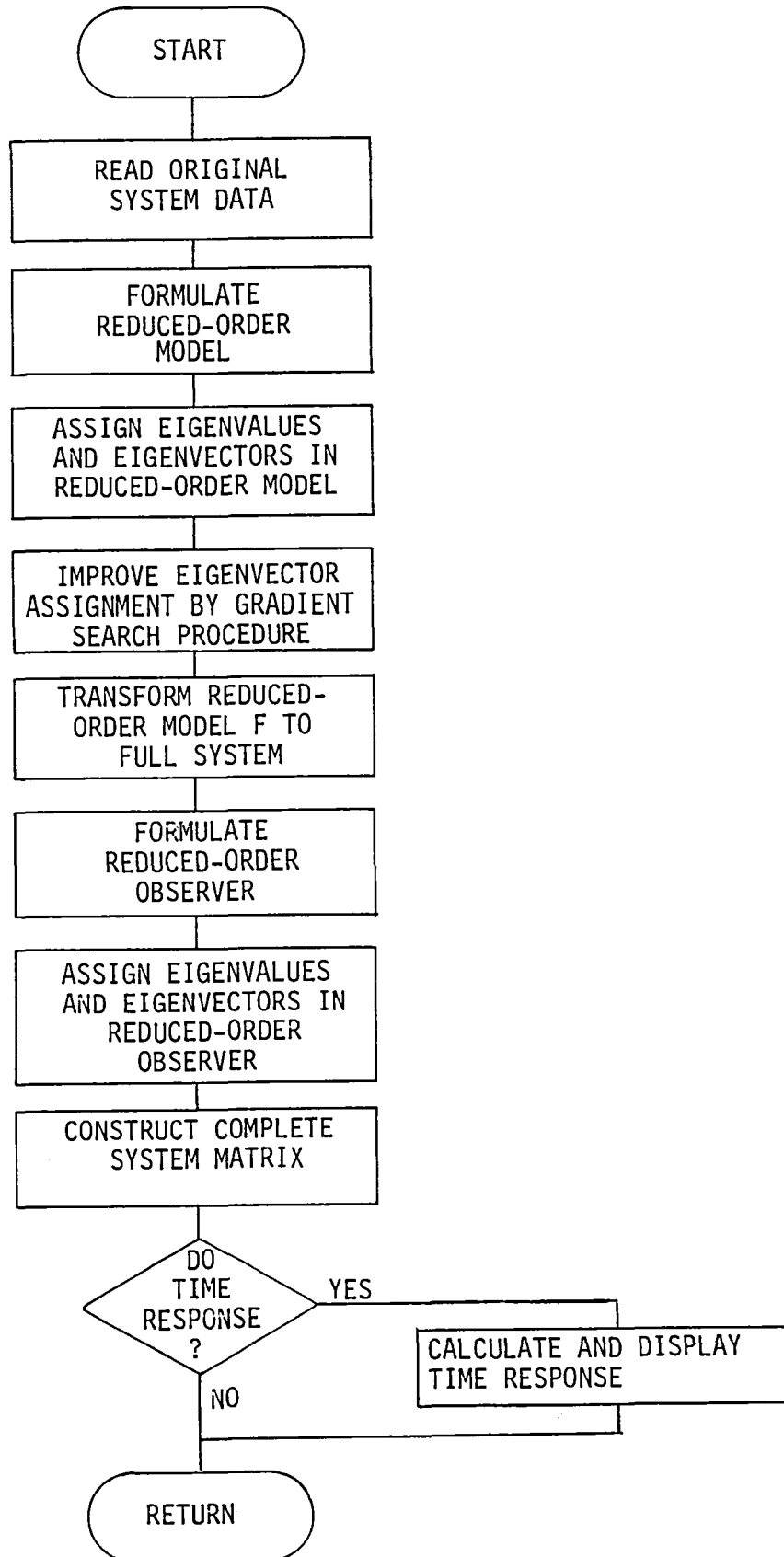


Figure 4.5. Mode 9



#### 4.5 Cost Function

The cost function  $J$  that is used in the gradient search routine is a measure of the error between the actual and desired partial eigenvector assignments in the full system model. Calculation of the cost function is accomplished in two parts. The actual partial eigenvector assignment is computed first using a subroutine called VACT. The actual partial eigenvectors are then used to compute the value of  $J$  in a subroutine called ROCOST.

A flowchart illustrating VACT is given in Figure 4.6. The actual partial eigenvector assignment is denoted by  $\bar{v}_{1i}$  and the eigenvector assigned in the reduced-order model is denoted by  $\hat{v}_{1i}$ . This is consistent with the notation used in Chapter 3. The partial eigenvector assignment  $\bar{v}_{1i}$  of the full order closed loop system that is obtained by assigning  $\hat{v}_{1i}$  in the reduced-order model can only be determined after all reduced-order model eigenvalues and eigenvectors are assigned and the feedback matrix  $F$  is computed. The top half of equation (3.23) is given by

$$v_{1i} = [V_{11} + V_{12} [\lambda_i I - \Lambda_2]^{-1} \hat{B}_2 \hat{F}] \hat{v}_{1i}. \quad (4.1)$$

For a real eigenvalue the subroutine computes  $\bar{v}_{1i}$  using equation (4.1). If the eigenvalue  $\lambda_i$  is complex the calculation becomes slightly more involved. Separating equation (4.1) into real and imaginary parts yields

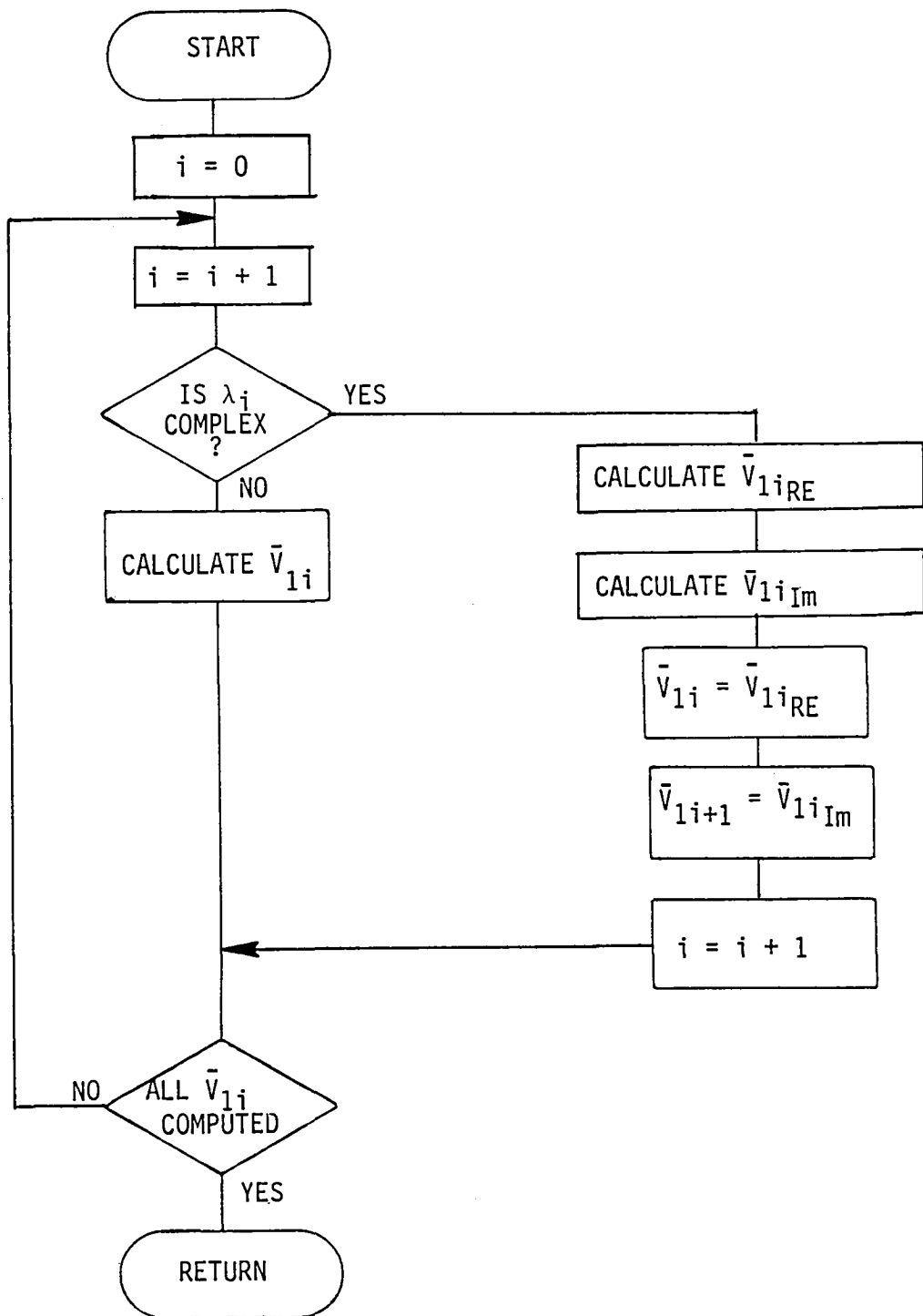


Figure 4.6. Subroutine VACT.

$$\bar{v}_{1i_{RE}} = [V_{11} + V_{12}(\lambda_{i_{RE}} I - \Lambda_2)^{-1} \hat{B}_2 \hat{F}] \hat{v}_{1i_{RE}} - [V_{12}(\lambda_{i_{IM}} I)^{-1} \hat{B}_2 \hat{F}] \hat{v}_{1i_{IM}} \quad (4.2)$$

and

$$\bar{v}_{1i_{IM}} = [V_{11} + V_{12}(\lambda_{i_{RE}} I - \Lambda_2)^{-1} \hat{B}_2 \hat{F}] \hat{v}_{1i_{IM}} + [V_{12}(\lambda_{i_{IM}} I)^{-1} \hat{B}_2 \hat{F}] \hat{v}_{1i_{RE}}. \quad (4.3)$$

Equations (4.2) and (4.3) are used to compute partial eigenvector assignments for complex eigenvalues. The actual eigenvector assignments are then used in subroutine ROCOST to compute the value of  $J$ .

A flowchart illustrating ROCOST is given in Figure 4.7. If the desired partial eigenvector assignment is denoted by  $v_D$  and the actual partial eigenvector assignment is denoted by  $\bar{v}$ , then the cost function is calculated by

$$J + \sum_{ij} (v_{ij} - v_{D_{ij}})^2 \alpha_{ij} \quad (4.4)$$

where  $\alpha_{ij}$  are arbitrary weighting constants. The weighting constants determine the relative penalty between the eigenvector component errors. For example, if one eigenvector component has a much larger weighting constant than the others, then an error in that component receives a much greater penalty than other component errors.

#### 4.6 Cost Function Gradient

The cost function gradient matrix is computed in subroutine ROGRAD. A flowchart illustrating ROGRAD is given in Figure 4.8. It is seen from

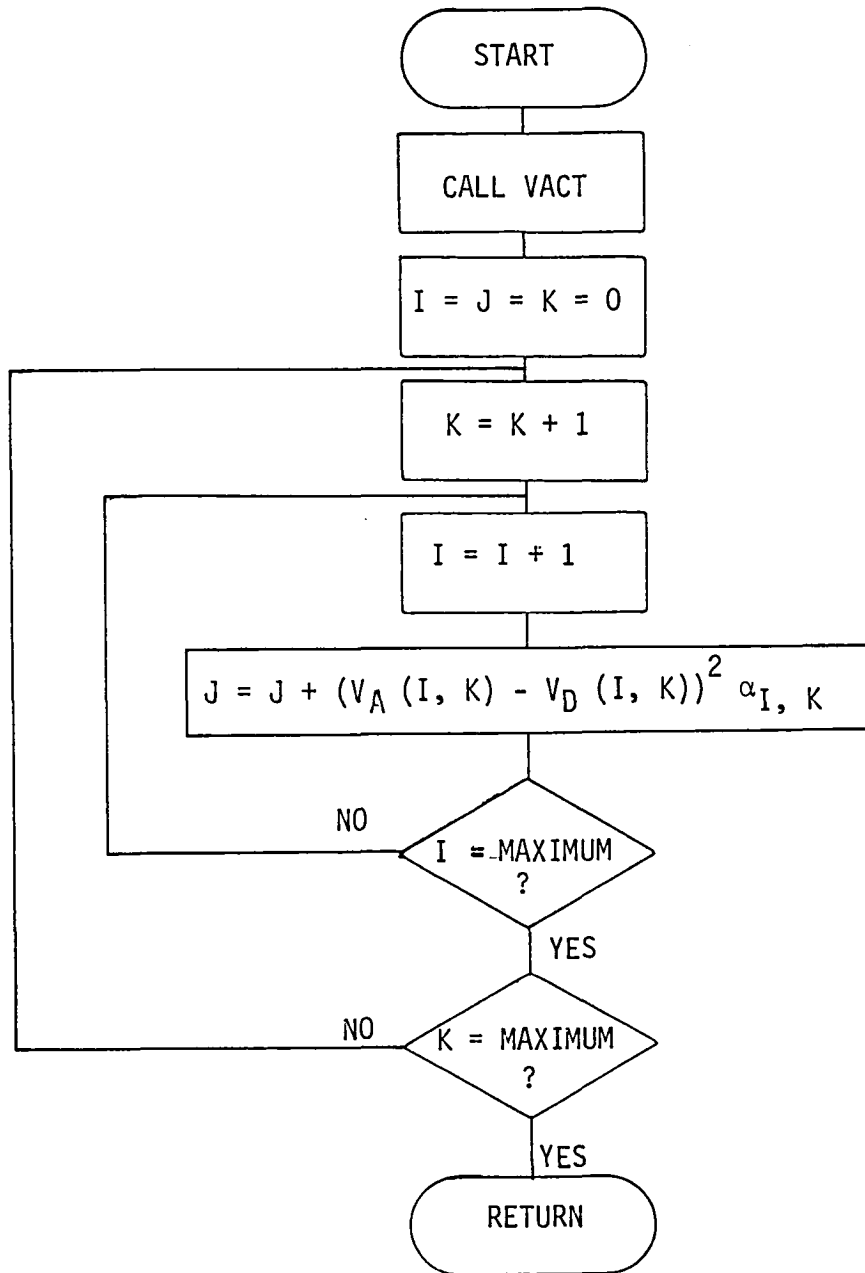


Figure 4.7. Subroutine ROCOST

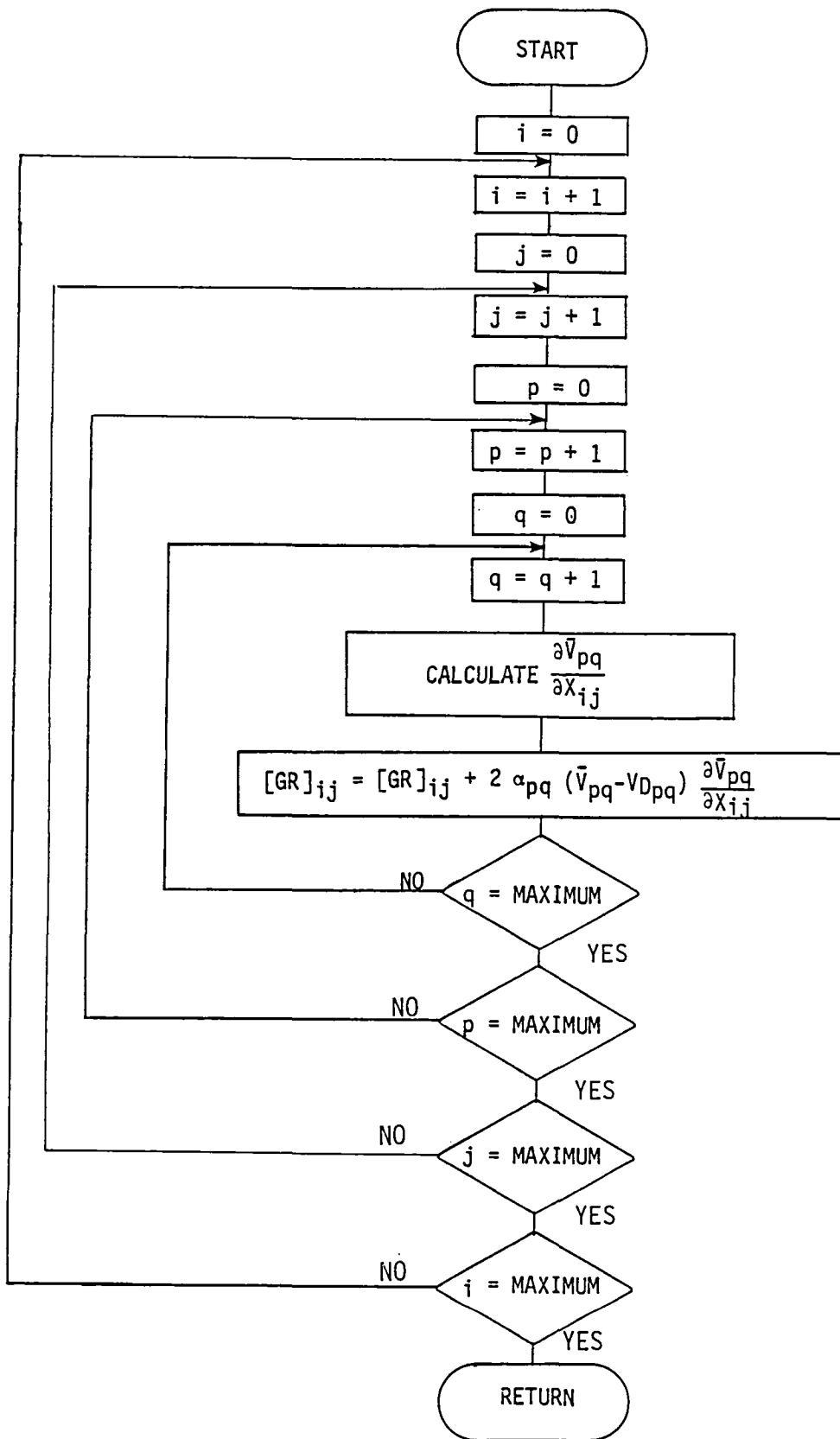


Figure 4.8. Subroutine ROGRAD

equation (4.4) that the cost function  $J$  is a function of the partial eigenvector assignment  $\bar{v}$ . Hence, it is also a function of the designator matrix  $X$  which is discussed in Section 2.6. By computing a gradient of the cost function  $J$  with respect to the elements of the designator matrix  $X_{ij}$ , it can be determined how to vary the designator matrix in order to reduce the cost function and therefore the error between  $v_D$  and  $\bar{v}$ . Recalling equation (2.80), the gradient matrix is defined to be

$$[GR]_{ij} = \frac{\frac{\partial J}{\partial X_{ij}}}{\left\| \frac{\partial J}{\partial X_{ij}} \right\|}. \quad (4.5)$$

Solving for  $\partial J / \partial X_{ij}$  gives

$$\frac{\partial J}{\partial X_{ij}} = \sum_{pq} 2 \alpha_{pq} (\bar{v}_{pq} - v_{D_{pq}}) \frac{\partial (v_{pq} - v_{D_{pq}})}{\partial X_{ij}}. \quad (4.6)$$

Since  $v_D$  is a constant valued matrix,

$$\frac{\partial J}{\partial X_{ij}} = \sum_{pq} 2 \alpha_{pq} (\bar{v}_{pq} - v_{D_{pq}}) \frac{\partial \bar{v}_{pq}}{\partial X_{ij}}. \quad (4.7)$$

To evaluate  $\partial \bar{v}_{pq} / \partial \lambda_{ij}$ ,  $q$  is substituted for  $i$  and the subscript 1 is dropped in equation (4.1) to yield

$$\bar{v}_q = [V_{11} + V_{12}(\lambda_q I - \Lambda_2)^{-1} \hat{B}_2 \hat{F}] \hat{v}_q. \quad (4.8)$$

Since  $\bar{v}_{pq}$  is the  $p^{\text{th}}$  element of vector  $\bar{v}_q$ , then  $\partial \bar{v}_{pq} / \partial \lambda_{ij}$  is the  $p^{\text{th}}$  element of  $\partial \bar{v}_q / \partial \lambda_{ij}$ . Solving for  $\partial \bar{v}_q / \partial \lambda_{ij}$ , then gives

$$\begin{aligned} \frac{\partial \bar{v}_q}{\partial \lambda_{ij}} &= \frac{\partial [V_{11} \hat{v}_q]}{\partial \lambda_{ij}} + V_{12}(\lambda_q I - \Lambda_2)^{-1} \hat{B}_2 \frac{\partial [\hat{F} \hat{v}_q]}{\partial \lambda_{ij}} \\ &= V_{11} \frac{\partial \hat{v}_q}{\partial \lambda_{ij}} + [V_{12}(\lambda_q I - \Lambda_2)^{-1} \hat{B}_2] \left[ \hat{F} \frac{\partial \hat{v}_q}{\partial \lambda_{ij}} + \frac{\partial \hat{F}}{\partial \lambda_{ij}} \hat{v}_q \right]. \end{aligned} \quad (4.9)$$

In order to evaluate  $\partial \hat{F} / \partial \lambda_{ij}$ , equation (2.73) is modified to be

$$\hat{F} = W \hat{V}^{-1} \quad (4.10)$$

and the element of  $F$  in the  $p^{\text{th}}$  row and  $q^{\text{th}}$  column is denoted by  $f_{pq}$  so that

$$f_{pq} = R_p [W] [\hat{V}^{-1}] C_q. \quad (4.11)$$

$R_p$  is a row vector with a one in the  $p^{\text{th}}$  column and zero elsewhere and  $C_q$  is a column vector with a one in the  $q^{\text{th}}$  row and zero elsewhere, so that

$$\frac{\partial f_{pq}}{\partial X_{ij}} = R_p \left[ \frac{\partial W}{\partial X_{ij}} \hat{V}^{-1} + W \frac{\partial \hat{V}^{-1}}{\partial X_{ij}} \right] C_q. \quad (4.12)$$

Solving for  $\partial W / \partial X_{ij}$  yields

$$\frac{\partial W}{\partial X_{ij}} = \frac{\partial}{\partial X_{ij}} [w_1, \dots, w_k]. \quad (4.13)$$

From equation (2.68),

$$w_j = -M_{\lambda_j} X_j. \quad (4.14)$$

Therefore

$$\frac{\partial W}{\partial X_{ij}} = [-M_{\lambda_1}] \frac{\partial X_1}{\partial X_{ij}}, \dots, [-M_{\lambda_K}] \frac{\partial X_K}{\partial X_{ij}}. \quad (4.15)$$

Noting that only the  $j^{\text{th}}$  column of  $X$  is dependent on  $X_{ij}$ ,

$$\frac{\partial W}{\partial X_{ij}} = [0, \dots, 0, [-M_{\lambda_j}] \partial X_j / \partial X_{ij}, 0, \dots, 0]. \quad (4.16)$$



Furthermore, only the  $i$ th row of  $X_j$  is dependent on  $X_{ij}$ . It is easily seen that  $\partial X_j / \partial X_{ij}$  is a column vector with a one in the  $i^{\text{th}}$  row and zero elsewhere. Hence, equation (4.16) is written

$$\frac{\partial W}{\partial X_{ij}} = [0, \dots, 0, [-M_{\lambda j}]_i, 0, \dots, 0] \quad (4.17)$$

where  $[-M_{\lambda j}]_i$  denotes the  $i^{\text{th}}$  column of  $[-M_{\lambda j}]$ . It is noted that

$$\frac{\partial \hat{V}^{-1}}{\partial X_{ij}} = \hat{V}^{-1} \frac{\partial \hat{V}^*}{\partial X_{ij}} \hat{V}^{-1} \quad (4.18)$$

and

$$\frac{\partial V}{\partial X_{ij}} = \frac{\partial}{\partial X_{ij}} [\hat{V}_1, \dots, \hat{V}_k]. \quad (4.19)$$

Since only the  $j^{\text{th}}$  eigenvector is a function of  $X_{ij}$ , it follows that

$$\frac{\partial \hat{V}}{\partial X_{ij}} = [0, \dots, 0, \frac{\partial \hat{V}_i}{\partial X_{ij}}, 0, \dots, 0]. \quad (4.20)$$

Substituting from equation (2.77) gives

$$\begin{aligned} \frac{\partial \hat{V}}{\partial X_{ij}} &= [0, \dots, 0, [N_{\lambda j}] \partial X_j / \partial X_{ij}, 0, \dots, 0] \\ &= [0, \dots, 0, [N_{\lambda j}]_i, 0, \dots, 0], \end{aligned} \quad (4.21)$$

where  $[N_{\lambda_j}]_i$  denotes the  $i^{\text{th}}$  column of  $[N_{\lambda_i}]$ . To evaluate  $\partial \hat{v}_q / \partial X_{ij}$ , equation (4.21) is postmultiplied by  $C_q$ . Similarly, to evaluate  $\partial \hat{v}_{pq} / \partial X_i$ , equation (4.9) is premultiplied by  $R_p$ .

Hence, it is shown that the partial derivatives are computed by selecting appropriate rows and columns from the  $[N_{\lambda}]$  and  $[M_{\lambda}]$  matrices. This reduces the calculation of the gradient matrix  $[GR]$  to a bookkeeping operation easily implemented in a computer program. It is not necessary to numerically approximate a derivative quantity. Subroutine ROGRAD computes the cost function gradient matrix using this procedure.

## CHAPTER 5

### DESIGN EXAMPLE

The design procedure described in Chapter 4 is illustrated in this chapter by an actual design problem. A controller is designed for the lateral axis model of an L-1011 aircraft using a reduced-order model and a reduced-order observer. The resulting design is then compared to an output feedback controller designed by Andrey et al. [15]. It is shown that the design procedure presented in this thesis is a viable tool for constant feedback controller design.

#### 5.1 Original Lateral Axis Model

The lateral axis model of an L-1011 aircraft is used as the original full-order system model. The state vector  $x$  is given by:

$x_1 = r = \text{Yaw rate (Radians/second)}$

$x_2 = \beta = \text{Sideslip angle (radians)}$

$x_3 = p = \text{Roll rate (radians/second)}$

$x_4 = \phi = \text{Bank angle (radians)}$

$x_5 = \delta_r = \text{Rudder deflection (radians)}$

$x_6 = \delta_a = \text{Aileron deflection (radians)}$

$x_7 = f_w = \text{Washout filter state.}$

Rudder and aileron deflections (states 5 and 5) produce changes in the yaw rate, sideslip angle, roll rate, and bank angle (states 1-4). The coordinate system is illustrated in Figure 5.1. Under certain conditions yaw rate is equal to the derivative of the sideslip angle with respect to time while roll rate is equal to the derivative of the bank angle with respect to time. The washout filter is a high pass filter for the yaw rate.

The A, B, and C system matrices are given by

$$A = \begin{bmatrix} -0.154 & 1.54 & -0.0042 & 0 & -0.744 & -0.032 & 0 \\ -0.996 & -0.117 & -0.000295 & 0.0386 & 0.02 & 0 & 0 \\ 0.249 & -5.2 & -1.0 & 0 & 0.337 & -1.12 & 0 \\ 0 & 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -20.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -25.0 & 0 \\ 0.5 & 0 & 0 & 0 & 0 & 0 & -0.5 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 20 & 0 \\ 0 & 25 \\ 0 & 0 \end{bmatrix},$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

The system input,  $u$ , consists of components

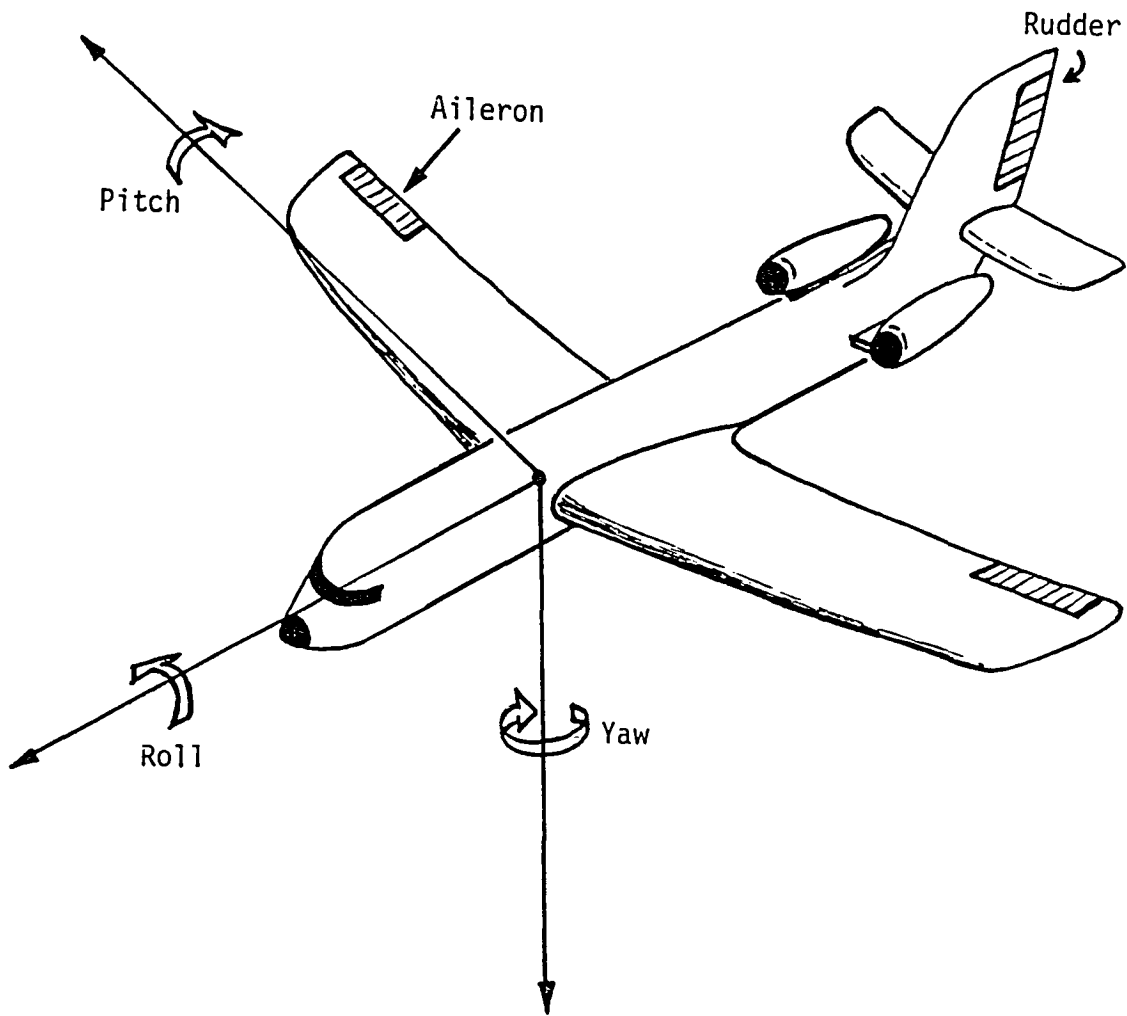


Figure 5.1. Aircraft Axis System

$$u_1 = \delta_{rc} = \text{Rudder command (radians)}$$

and

$$u_2 = \delta_{ac} = \text{Aileron command (radians).}$$

The open loop eigenvalues of this system are:

$\lambda_{1,2}$	=	$-0.08819 \pm j 1.269$	- Dutch roll mode
$\lambda_3$	=	$-1.085$	- Roll subsistence mode
$\lambda_4$	=	$-0.00965$	- Spiral mode
$\lambda_5$	=	$-20.0$	- Rudder mode
$\lambda_6$	=	$-25.0$	- Aileron mode
$\lambda_7$	=	$-0.5$	- Washout filter mode.

The open loop system time response is shown in Figures 5.2-5.8 for zero input and an initial condition of  $\phi(0) = 1$  degree. After ten seconds the system states are still oscillating and the bank angle  $\phi$  has not yet reached zero degrees.

It is known that a desirable eigenvalue assignment for the system is

$$\lambda_{1,2} = -1.5 \pm j 1.5$$

and

$$\lambda_{3,4} = -2.0 \pm j 1.0.$$

When the roll subsistence mode  $\lambda_3$  and the spiral mode  $\lambda_4$  are a com-

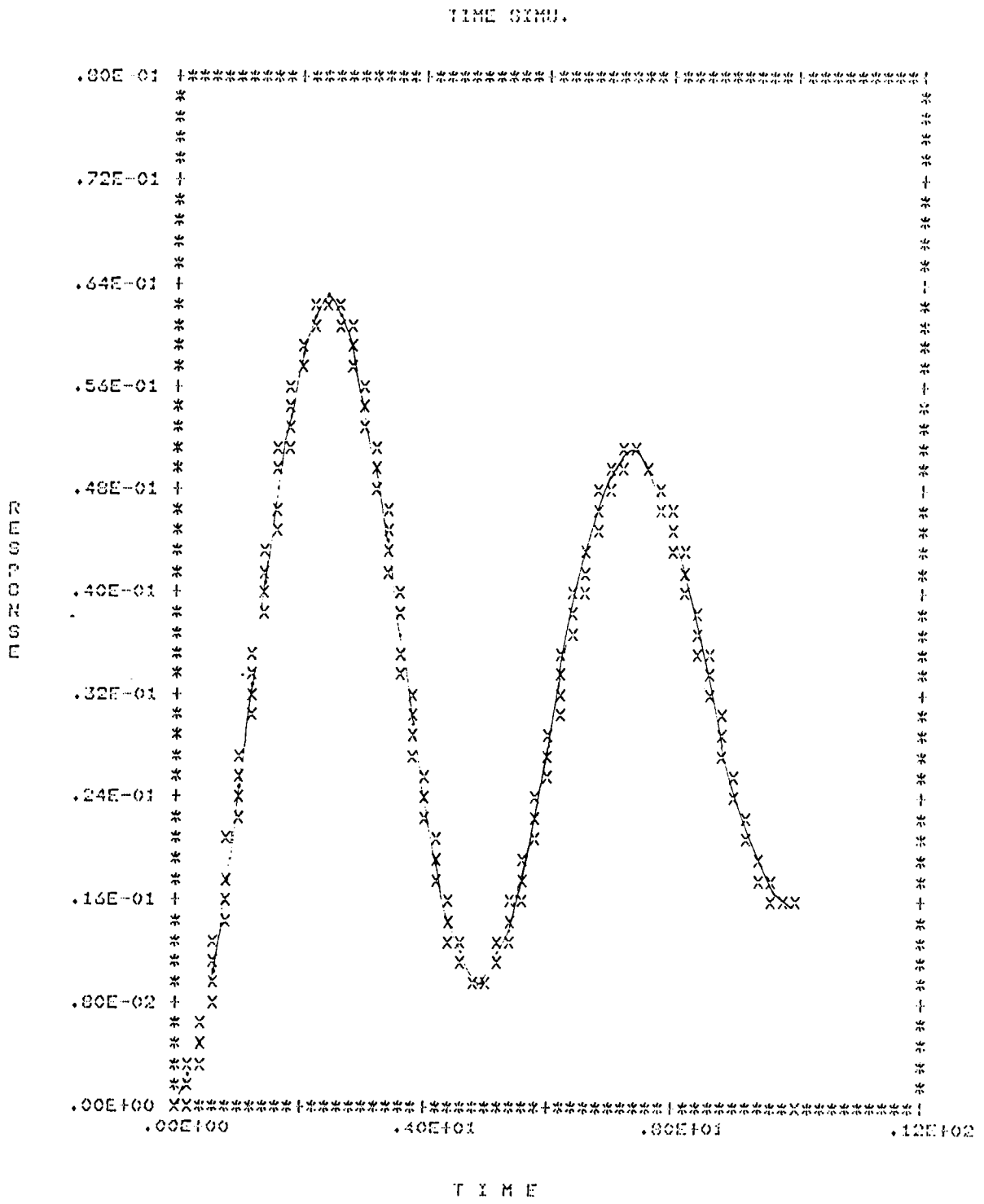


Figure 5.2. Yaw Rate-Open Loop Response for  $\phi(0) = 1^\circ$

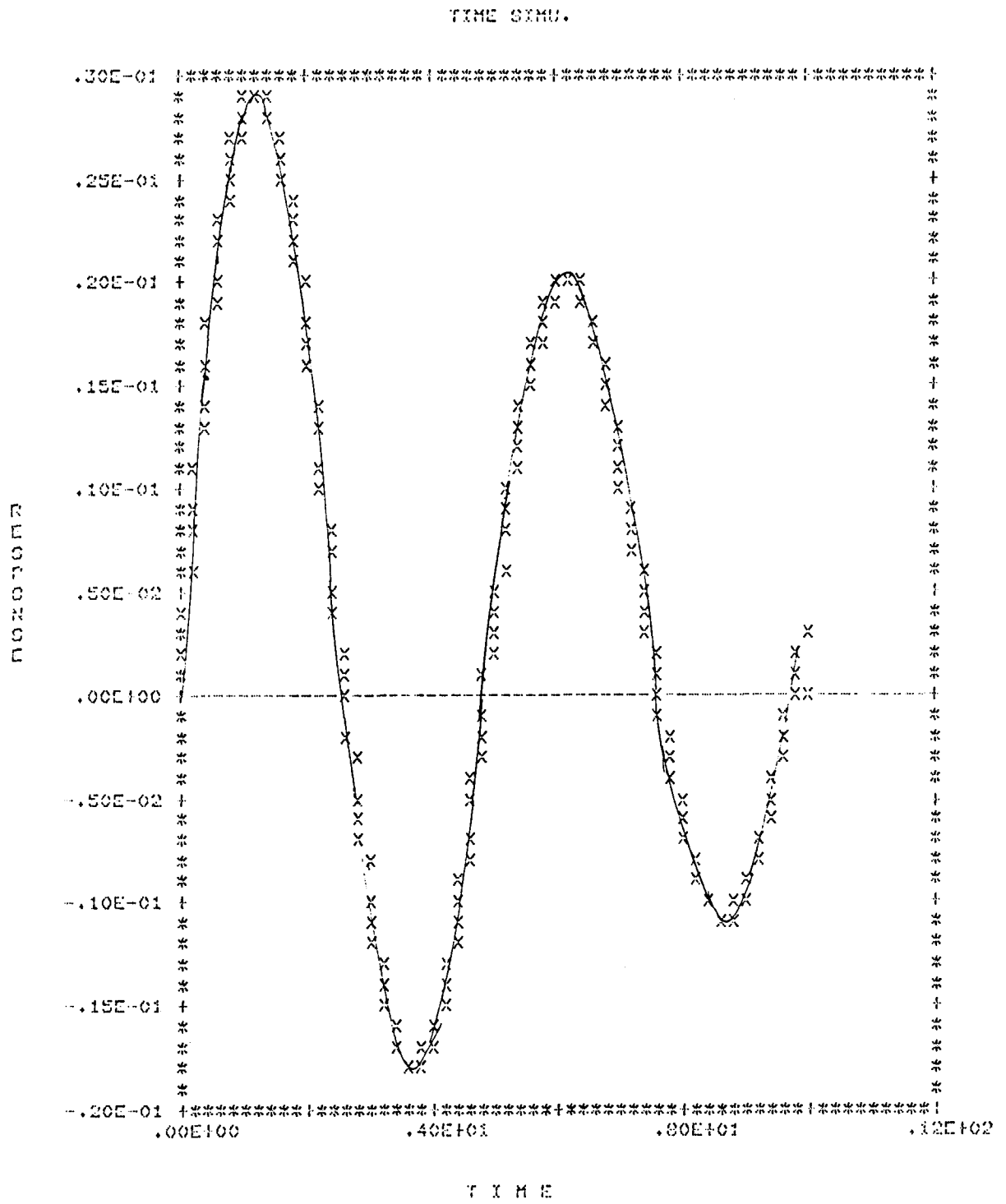
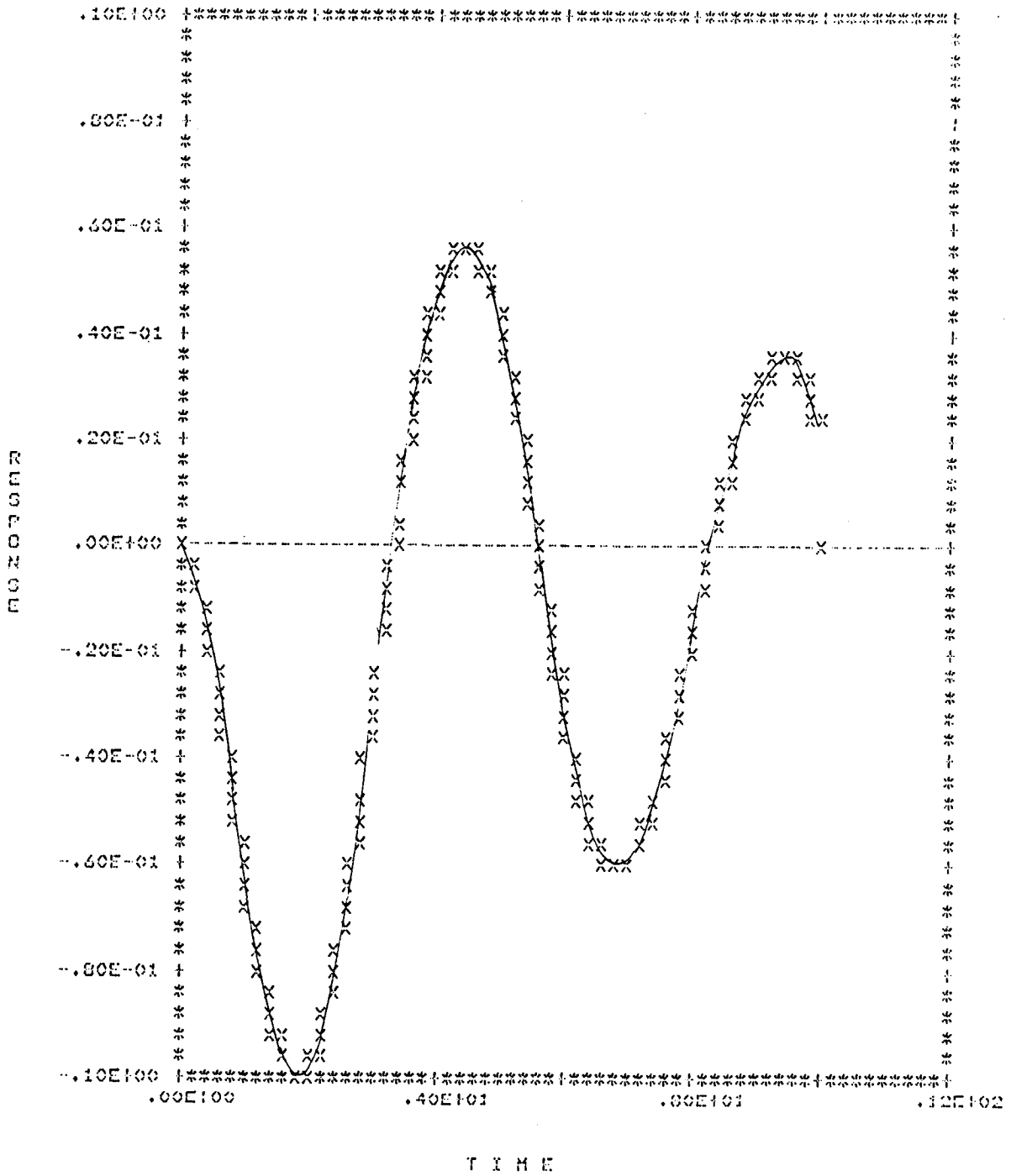


Figure 5.3. Sideslip Angle-Open Loop Response for  $\phi(0) = 1^\circ$



TIME SIMU.

Figure 5.4. Roll Rate-Open Loop Response for  $\phi(0) = 1^\circ$

TIME SIMU.

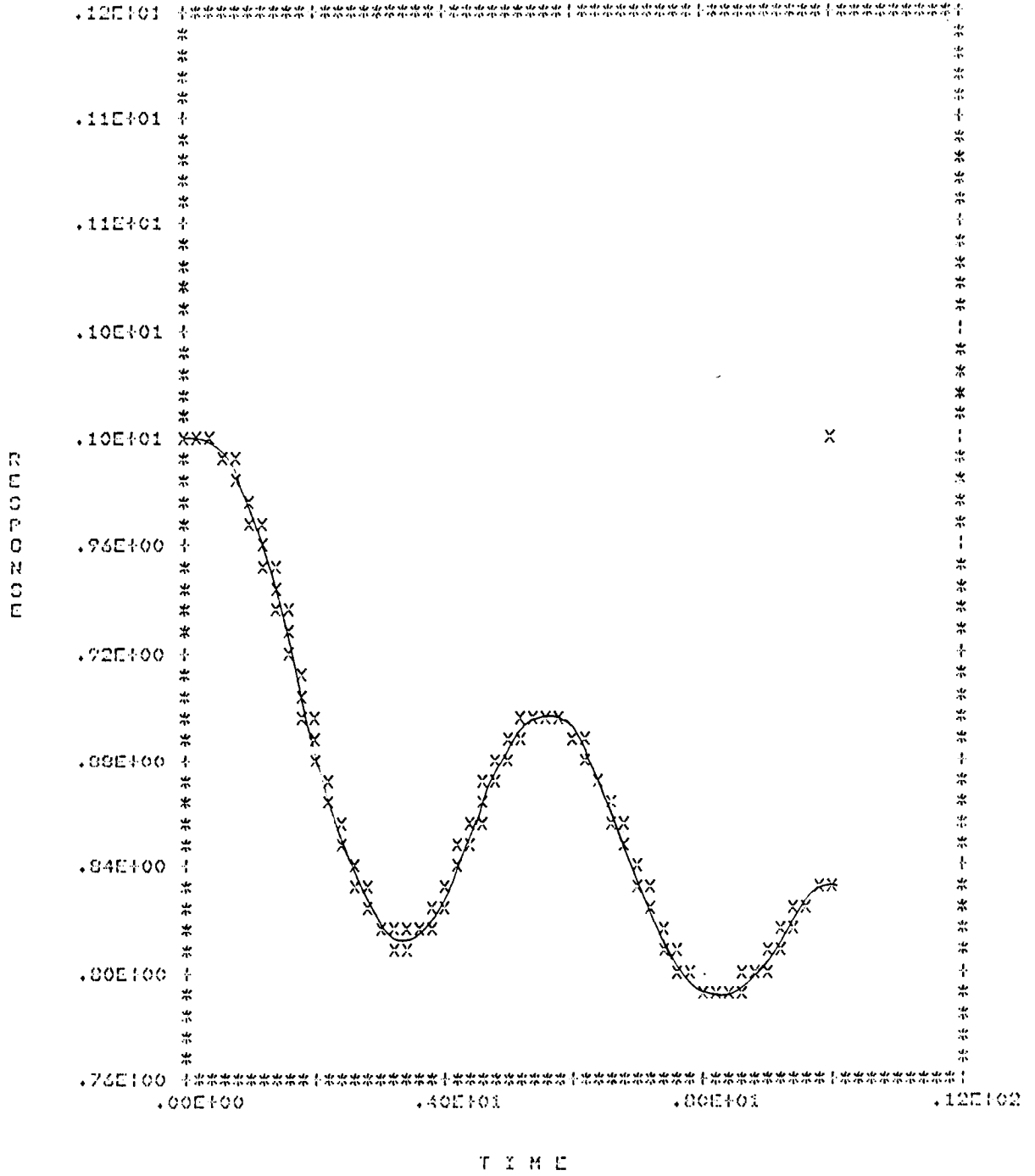


Figure 5.5. Bank Angle-Open Loop Response for  $\phi(0) = 1^\circ$

TIME SIMU.

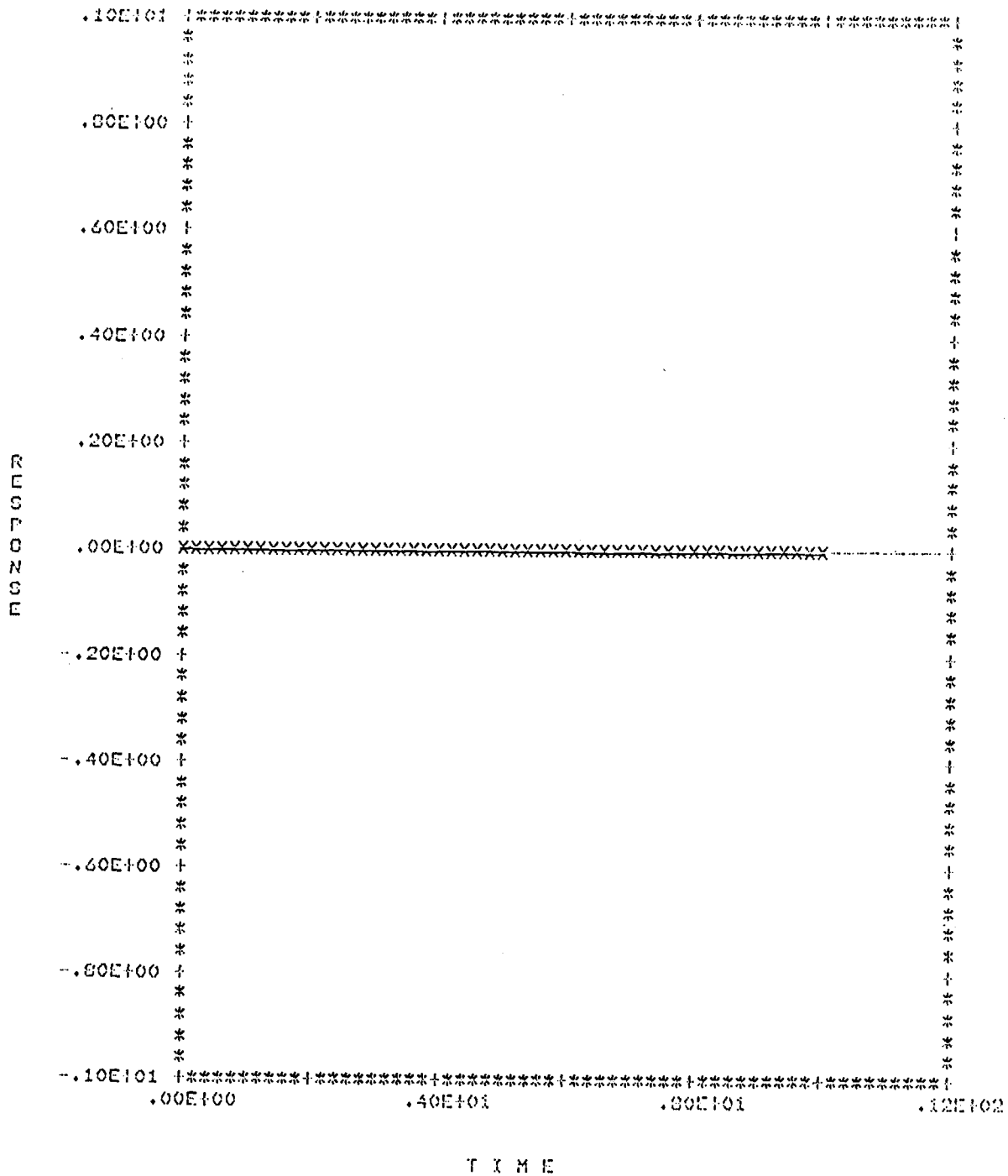


Figure 5.6. Rudder Deflection-Open Loop Response for  $\phi(0) = 1^\circ$

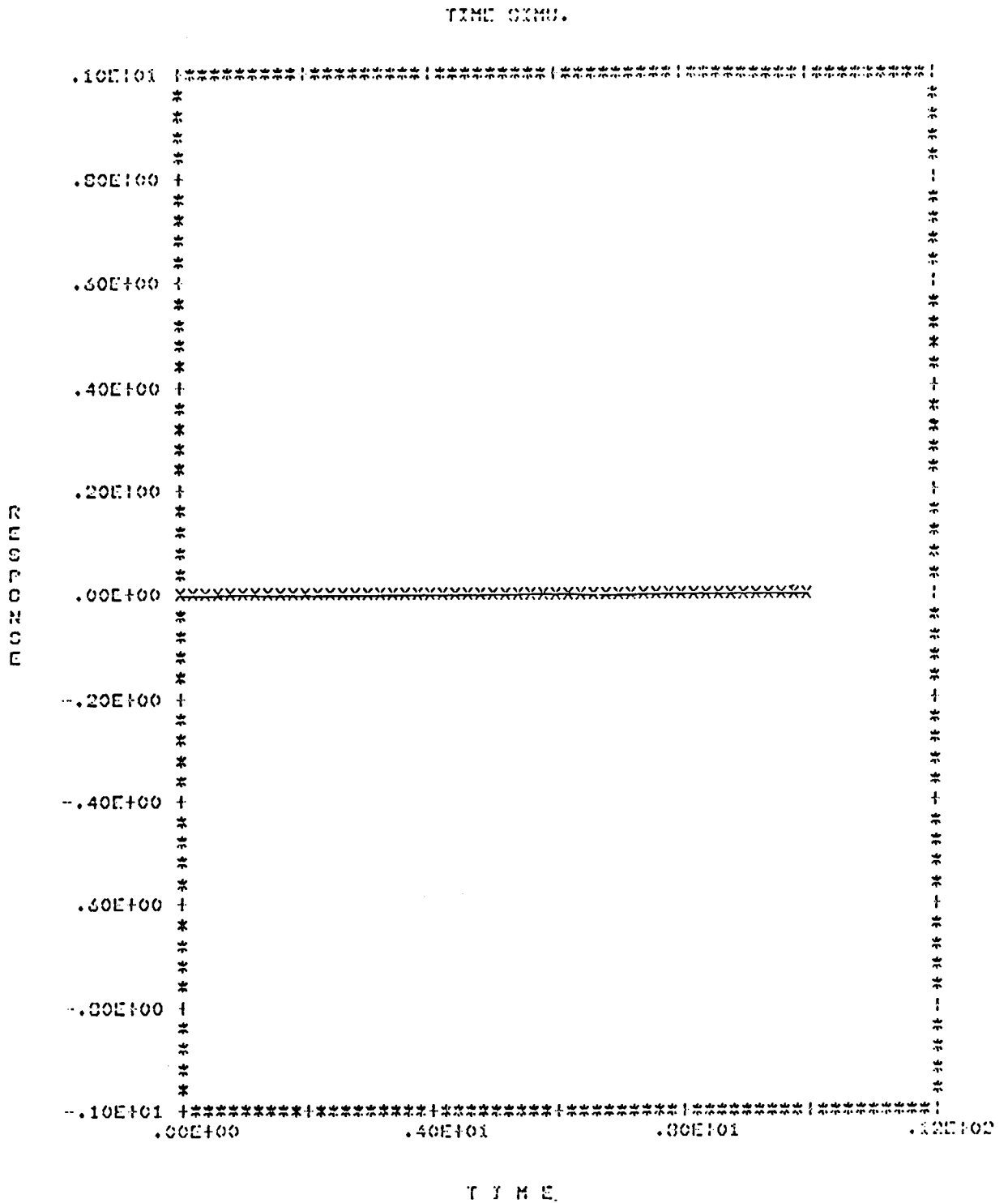


Figure 5.7. Aileron Deflection-Open Loop Response for  $\phi(0) = 1^\circ$

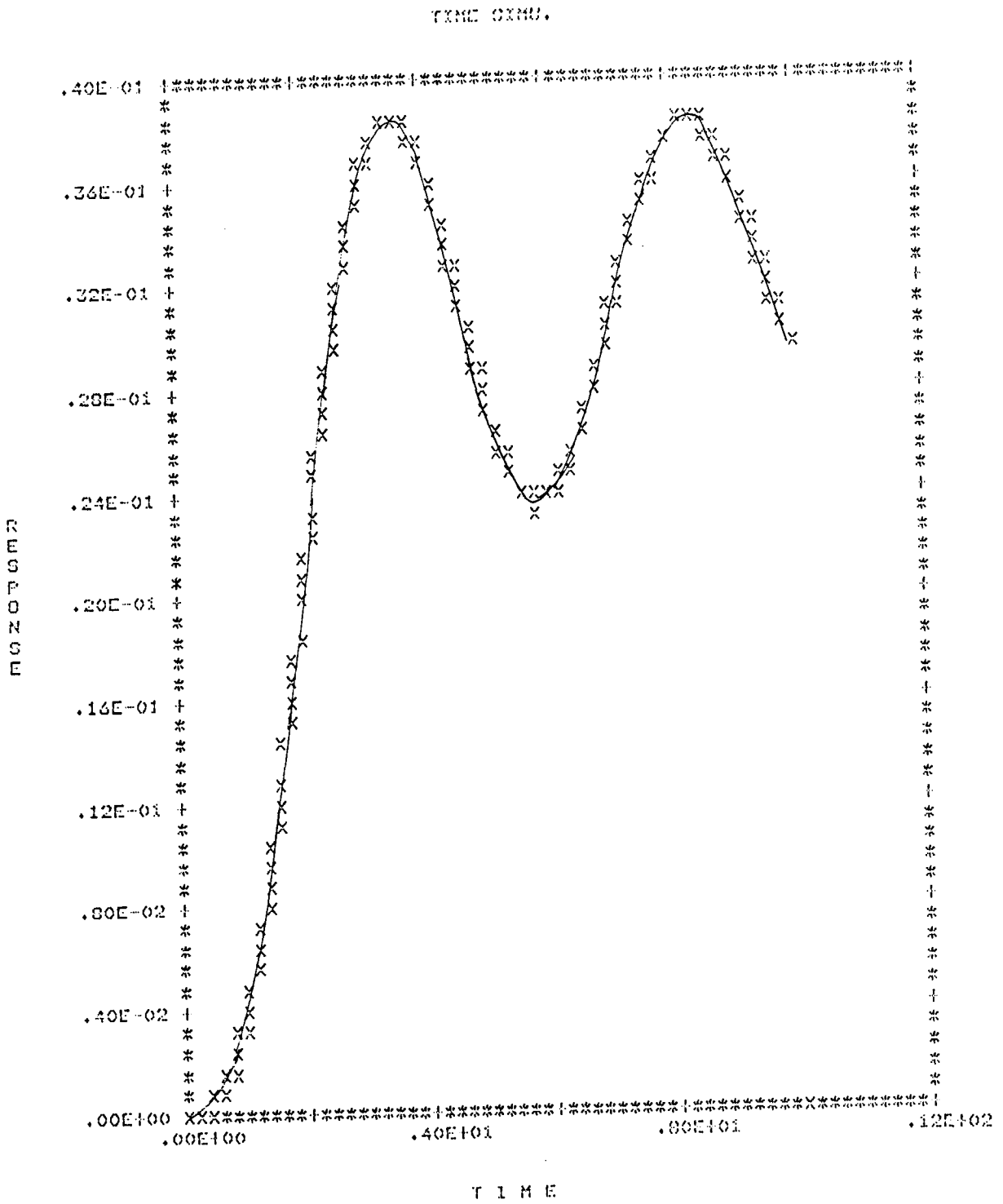


Figure 5.8. Washout Filter-Open Loop Response for  $\phi(0) = 1^\circ$

plex conjugate pair they are collectively referred to as the roll mode. It is also known to be desirable for the roll and dutch roll modes to be decoupled. This decoupling is accomplished by the eigenvector selection:

$$V_1 = \begin{bmatrix} 1 \\ X \\ 0 \\ 0 \\ X \\ X \\ X \\ X \end{bmatrix}, \quad V_2 = \begin{bmatrix} X \\ 1 \\ 0 \\ 0 \\ X \\ X \\ X \\ X \end{bmatrix}, \quad V_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ X \\ X \\ X \\ X \\ X \end{bmatrix}, \quad V_4 = \begin{bmatrix} 0 \\ 0 \\ X \\ 1 \\ X \\ X \\ X \\ X \end{bmatrix},$$

where  $X$  denotes "don't care." Andry, Shapiro and Chung [15] closely approximate the above eigenvalue and eigenvector assignment for this system using constant output feedback. Eigenvalue/eigenvector assignment techniques are used to design the constant output feedback matrix

$$K = \begin{bmatrix} 3.35 & -0.159 & -4.88 & -0.379 \\ 1.42 & 2.38 & -6.36 & 3.8 \end{bmatrix}.$$

The closed loop time response is shown in Figures 5.9-5.15. The closed loop eigenvalues of the design are

$$\begin{aligned} \lambda_{1,2} &= -1.052 \pm j 1.497 \\ \lambda_{3,4} &= -2.001 \pm j 0.9995 \\ \lambda_5 &= -17.05 \\ \lambda_6 &= -22.01 \\ \lambda_7 &= -0.6989. \end{aligned}$$

TIME SIMU.

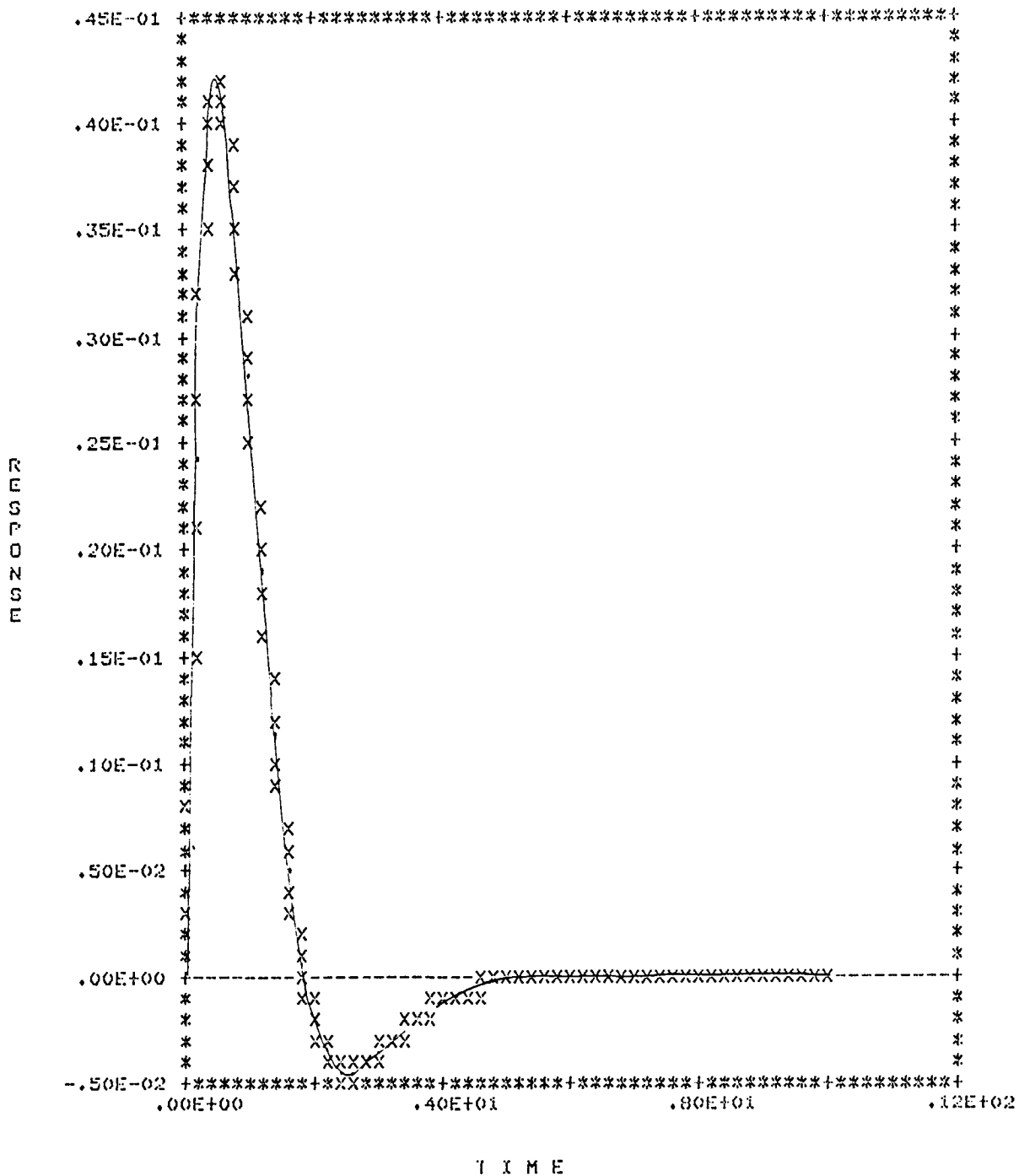


Figure 5.9. Yaw Rate-First Closed Loop Response for  $\phi(0) = 1^\circ$

TIME SIMU.

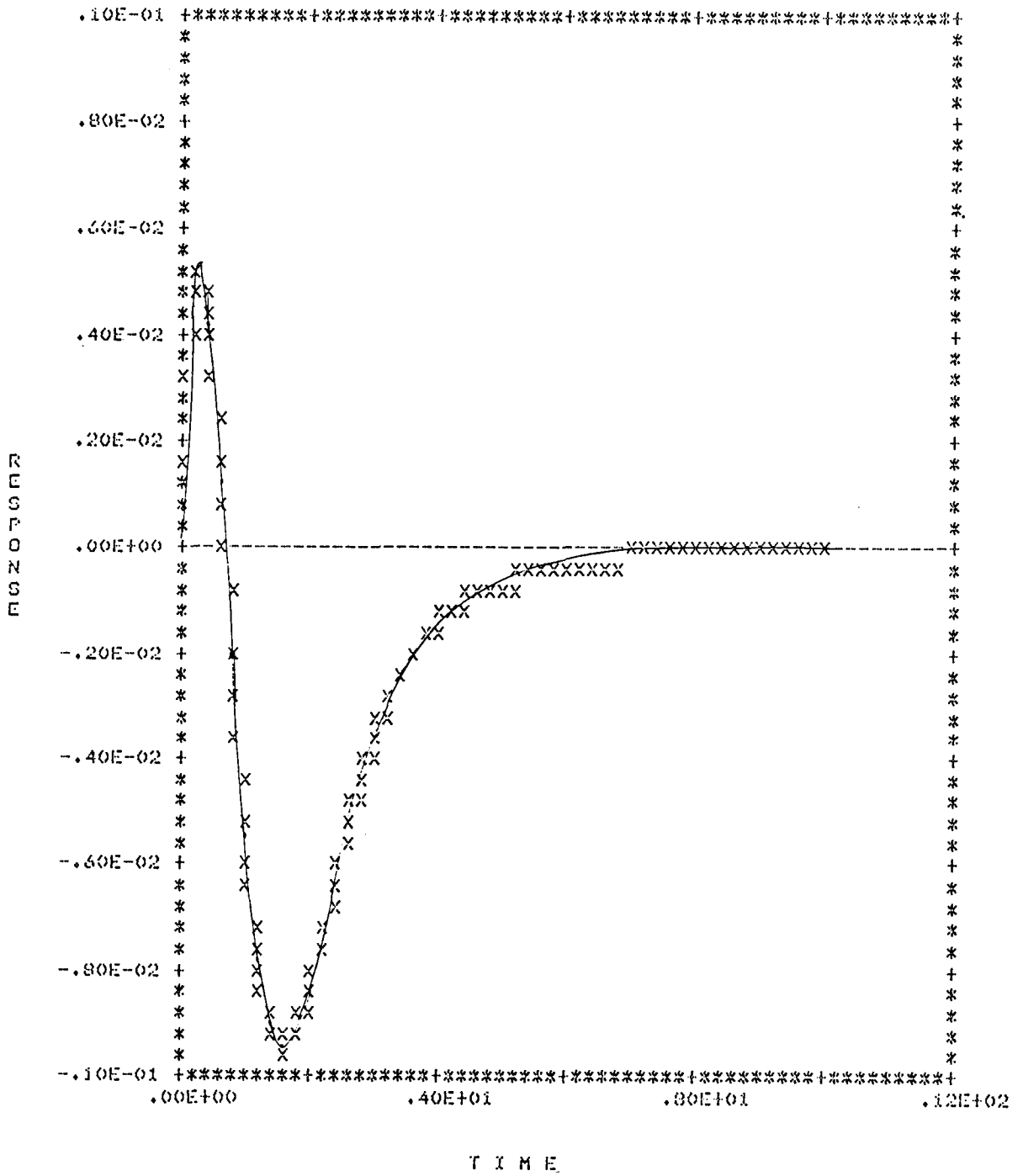
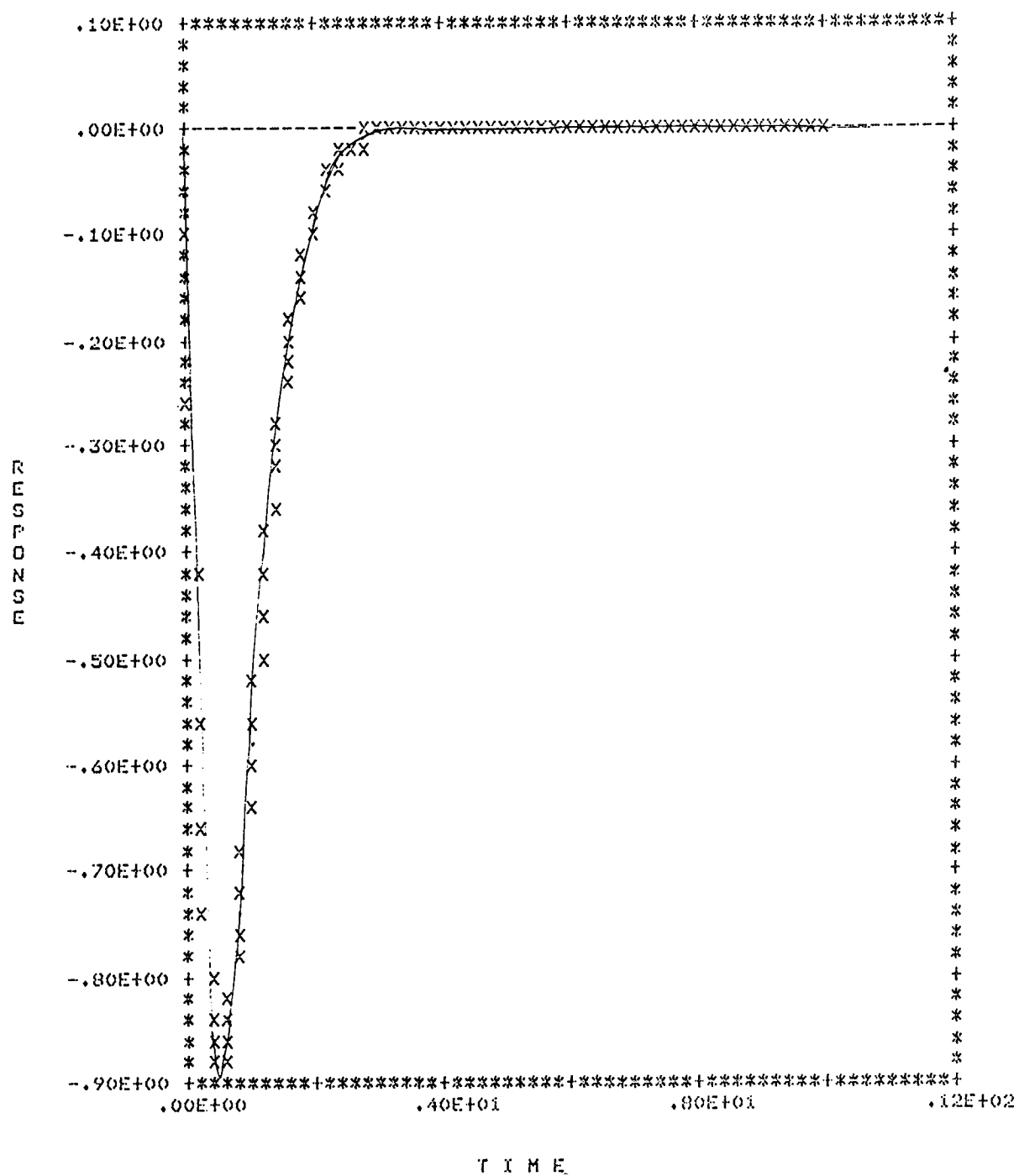


Figure 5.10. Sideslip Angle-First Closed Loop Response for  $\phi(0) = 1^\circ$



TIME SIMU.

Figure 5.11. Roll Rate-First Closed Loop Response for  $\phi(0) = 1^\circ$

TIME SIMU.

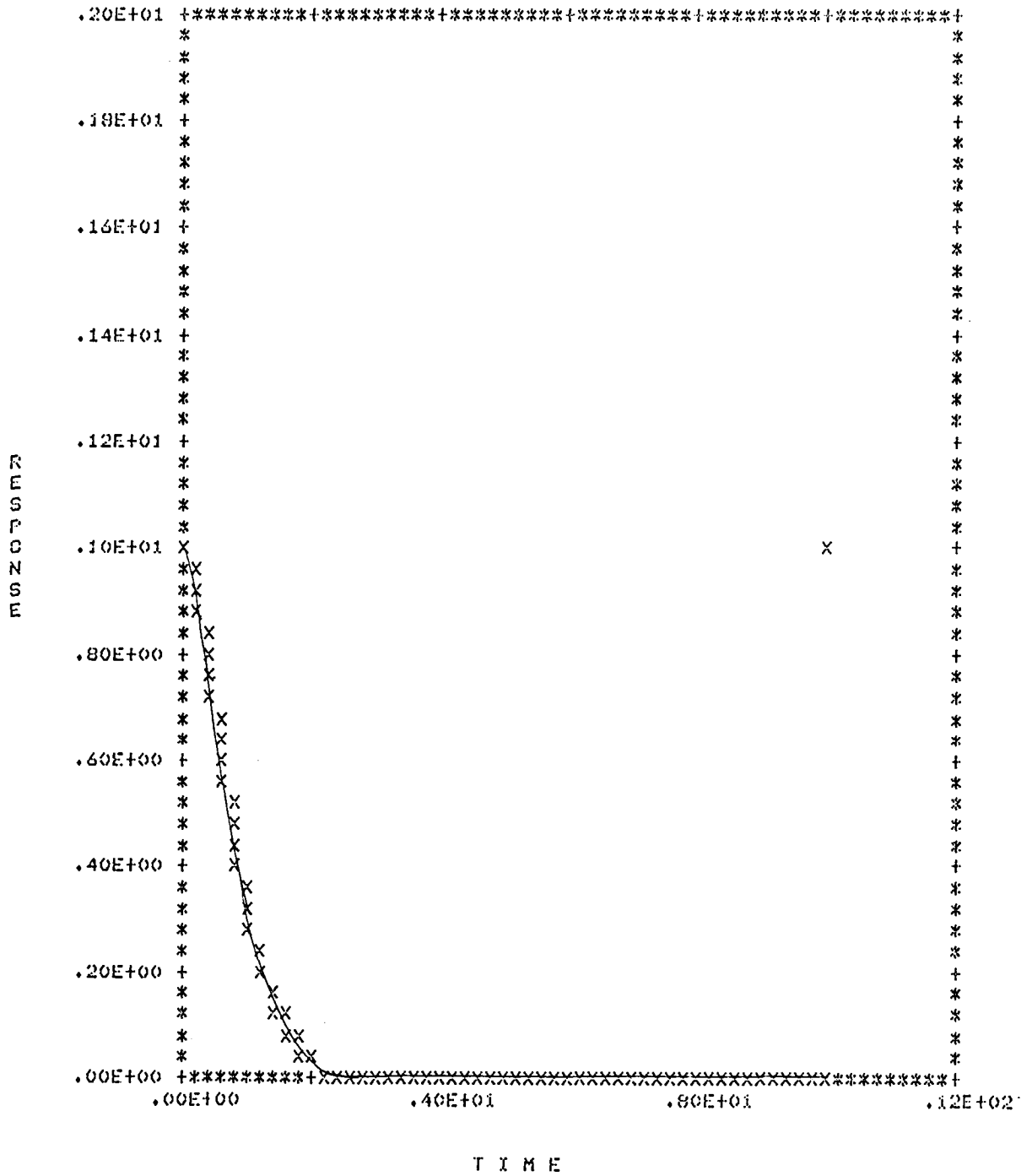


Figure 5.12. Bank Angle-First Closed Loop Response for  $\phi(0) = 1^\circ$

TIME SIMU.

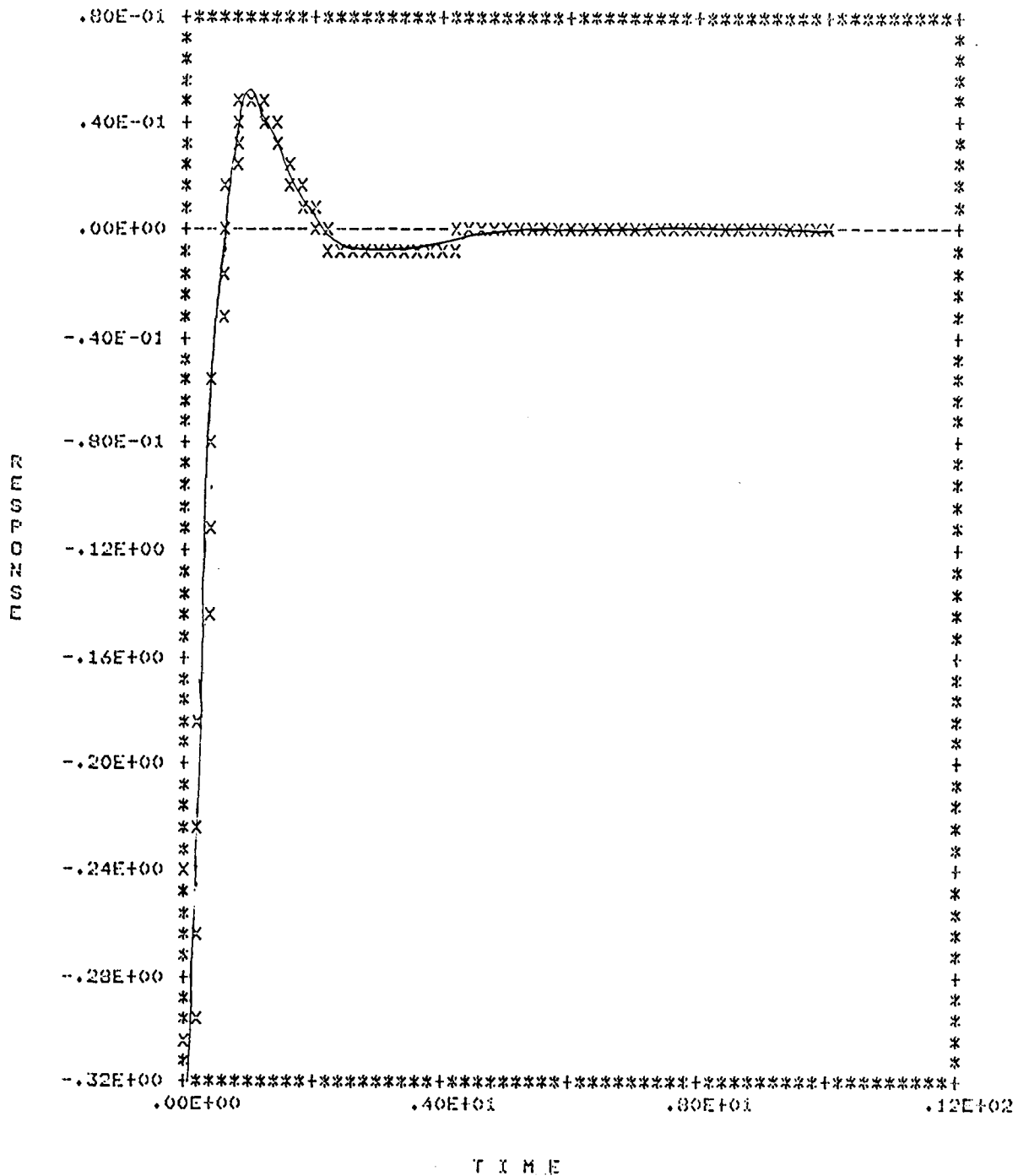


Figure 5.13. Rudder Deflection-First Closed Loop Response for  $\phi(0) = 1^\circ$

TIME SIMU.

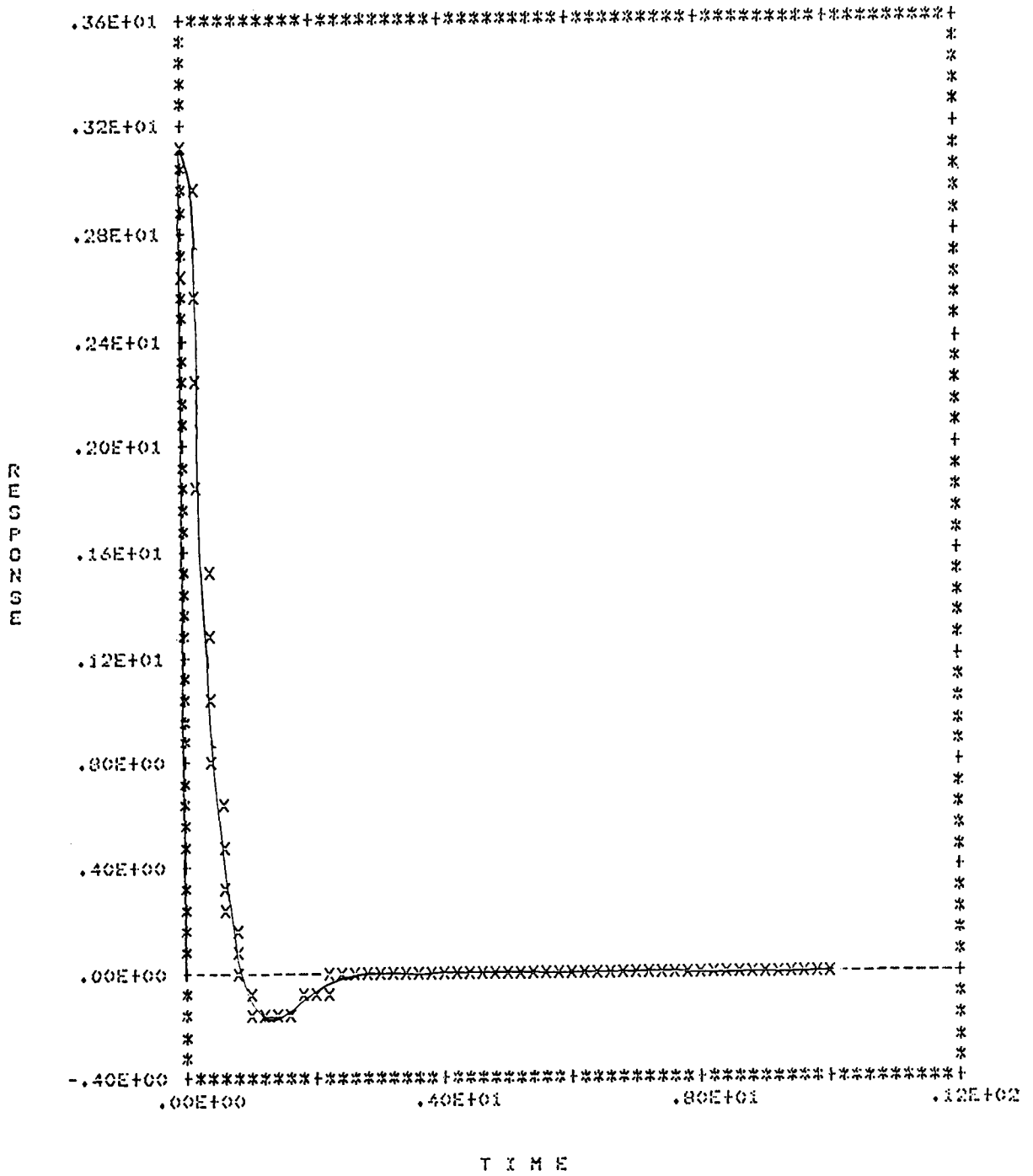


Figure 5.14. Aileron Deflection-First Closed Loop Response for  $\phi(0) = 1^\circ$

TIME SIMU.

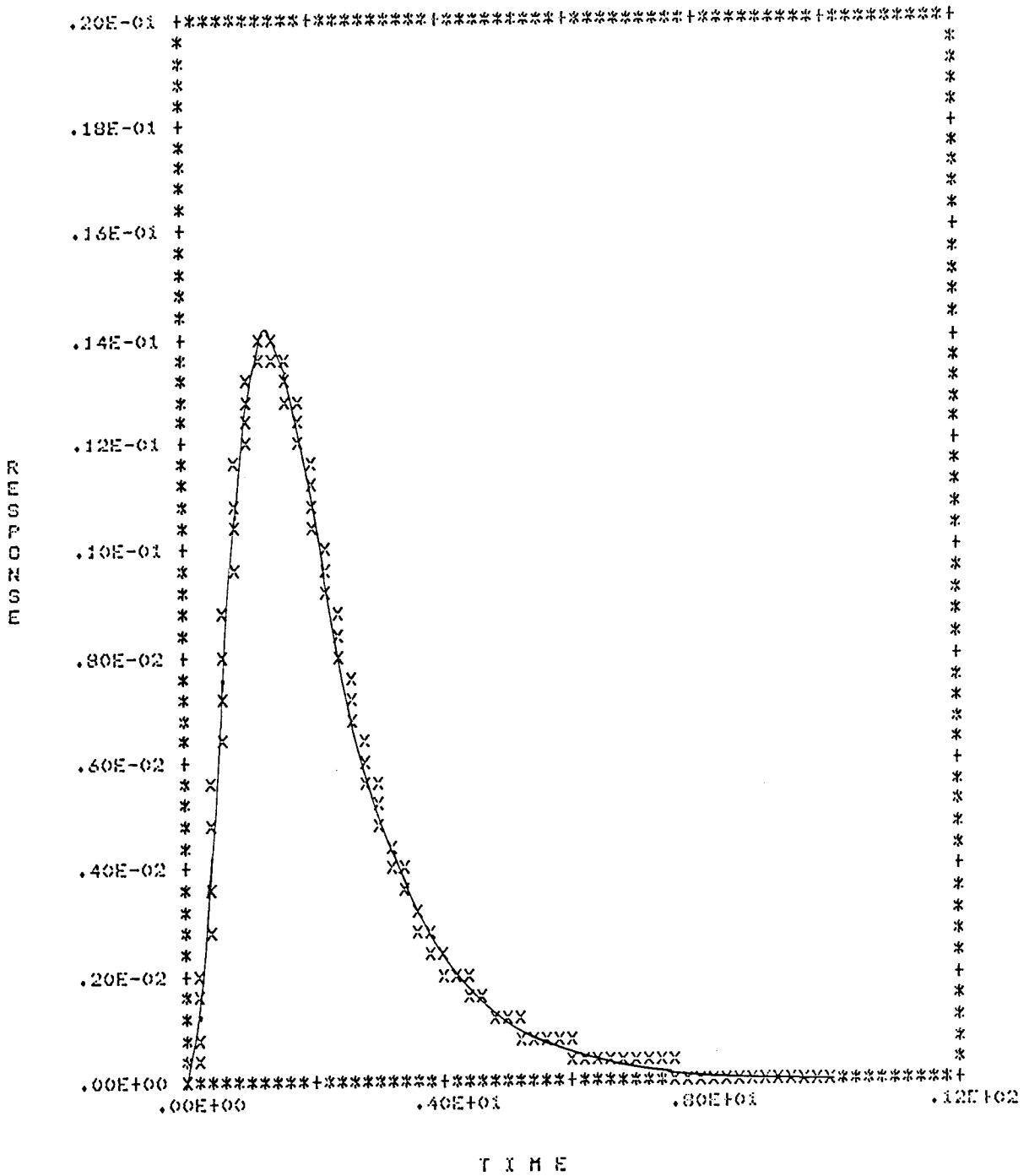


Figure 5.15. Washout Filter-First Closed Loop Response for  $\phi(0) = 1^\circ$

The first four components of the first four eigenvectors are:

$$V_{1,2} = \begin{bmatrix} 1 \\ 0.03066 \pm j 0.3488 \\ -0.0036 \pm j 0.0004 \\ 0.0013 \pm j 0.0011 \end{bmatrix}, \quad V_{3,4} = \begin{bmatrix} -0.0029 \mp j 0.0012 \\ 0.0045 \pm j 0.0053 \\ 1 \\ -0.3999 \mp j 0.2000 \end{bmatrix}.$$

It is noted that by using constant output feedback that the four eigenvalues  $\lambda_1 - \lambda_4$  are placed almost exactly and that the roll and dutch roll modes are decoupled. However, the other eigenvalues ( $\lambda_5 - \lambda_7$ ) are also moved by the design. The design procedure described in Chapter 4 is now used to formulate an alternate design that exactly places  $\lambda_1 - \lambda_4$  without changing  $\lambda_5 - \lambda_7$ . Eigenvectors for  $\lambda_1 - \lambda_4$  are also assigned to achieve roll and dutch roll mode decoupling. This is achieved without modifying the eigenvectors associated with  $\lambda_5 - \lambda_7$ . Furthermore, this design is done using a reduced-order model to specify a constant feedback matrix for the original full-order system. The full state feedback matrix is then implemented by dynamic output feedback.

## 5.2 Reduced-Order Model Design

Since the open loop values of  $\lambda_5 - \lambda_7$  are known to be desirable, and the rudder, aileron, and washout filter states are unspecified, no reassignment of these modes will be made. Therefore, they are not included in the reduced-order model. On the other hand,  $\lambda_1 - \lambda_4$  are to be reassigned and are included in the reduced-order model. The full order system matrices are transformed by equation (3.3) and partitioned

as in equation (3.5) to yield the reduced-order model system matrices

$$\Lambda_1 = \begin{bmatrix} -0.08819 & 1.269 & 0 & 0 \\ -1.269 & -0.08819 & 0 & 0 \\ 0 & 0 & -1.085 & 0 \\ 0 & 0 & 0 & -0.009165 \end{bmatrix}, \quad B_1 = \begin{bmatrix} -1.706 & -0.02580 \\ 0.3961 & 0.06988 \\ -0.2772 & -0.2878 \\ -0.2698 & -0.1528 \end{bmatrix}.$$

Spectral assignment synthesis methods are then used to assign the eigenvalues

$$\lambda_{1,2} = -1.5 \pm j 1.5$$

and

$$\lambda_{3,4} = -2.0 \pm j 1.0.$$

A partial eigenvector assignment that achieves roll and dutch roll mode decoupling is given by

$$V_{1,2} = \begin{bmatrix} 20 \\ 6 \pm j 7 \\ 0 \\ 0 \end{bmatrix}, \quad V_{3,4} = \begin{bmatrix} 0 \\ 0 \\ 20 \\ -8 \mp j 4 \end{bmatrix}.$$

An initial attempt is made to assign the eigenvectors using equation (3.24). The partial assignment in the full order system is found to be

$$V_{1,2} = \begin{bmatrix} 19.44 \pm j 0.33 \\ 6.76 \pm j 7.25 \\ 1.06 \mp j 0.10 \\ -0.42 \pm j 0.64 \end{bmatrix}, \quad V_{3,4} = \begin{bmatrix} -0.10 \mp j 0.04 \\ 0.08 \pm j 0.10 \\ 20.20 \pm j 0.11 \\ -8.54 \mp j 3.91 \end{bmatrix}.$$

The gradient search routine described in Section 2.6 is now used to improve the initial vector assignment. Elements of a weighting matrix are entered into the computer and a value is calculated for the cost function  $J$  using equation (4.4). A cost function gradient is calculated as in Section 4.6 and the initial eigenvector assignment is varied to reduce the cost function. The weighting matrix is varied to increase or decrease the relative importance of each eigenvector component and the gradient search is continued. This procedure is repeated until a satisfactory improvement of the initial assignment is achieved. In this example the final partial eigenvector assignment in the full system model is given by

$$V_{1,2} = \begin{bmatrix} 19.45 \pm j 0.34 \\ 6.76 \pm j 7.25 \\ 0.45 \pm j 0.33 \\ -0.07 \pm j 0.68 \end{bmatrix}, \quad V_{3,4} = \begin{bmatrix} -0.10 \mp j 0.04 \\ 0.08 \pm j 0.10 \\ 20.20 \pm j 0.11 \\ -8.54 \mp j 3.91 \end{bmatrix}.$$

The vectors are scaled to give

$$V_{1,2} = \begin{bmatrix} 1 \\ 0.35 \pm j 0.37 \\ 0.02 \pm j 0.02 \\ 0.003 \pm j 0.04 \end{bmatrix}, \quad V_{3,4} = \begin{bmatrix} -0.005 \mp j 0.002 \\ 0.004 \pm j 0.005 \\ 1 \\ -0.424 \mp j 0.191 \end{bmatrix}.$$

It is seen from the above vectors that the roll and dutch roll modes have been significantly decoupled. The required gain matrix in the reduced-order model is given by



$$F = \begin{bmatrix} 1.319 & -1.650 & 0.169 & -1.724 \\ -3.854 & 0.583 & -5.810 & 31.18 \end{bmatrix}.$$

The constant state feedback matrix in the full order system is computed using equations (3.9) and (3.10) to be

$$F = \begin{bmatrix} 3.66 & -3.13 & -0.176 & -0.372 & -0.137 & 0.003 & 0 \\ 1.54 & -6.03 & 2.70 & 4.35 & -0.008 & -0.120 & 0 \end{bmatrix}.$$

In order to implement this full state feedback matrix, an observer is now designed by the procedure described in Section 3.4. The observer eigenvalues,  $\lambda_{0i}$ , are selected so that

$$\lambda_{01} = -5 \quad , \quad \lambda_{02} = -6 \quad , \quad \lambda_{03} = -7.$$

This selection makes the observer modes faster than the modes contained in the reduced-order model. The observer eigenvectors,  $v_{0i}$ , are arbitrarily assigned to be

$$v_{01} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad , \quad v_{02} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad , \quad v_{03} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

The observer matrices are then calculated to be:

$$E = \begin{bmatrix} -5 & 0 & 0 \\ 0 & -6 & 0 \\ 0 & 0 & -7 \end{bmatrix},$$

$$R = \begin{bmatrix} 0.287 & -0.029 & -28.3 & -0.372 \\ 0.184 & 0.710 & -17.1 & 4.35 \end{bmatrix},$$

$$G = \begin{bmatrix} -89.2 & 2.35 & -17.8 & 0.124 \\ -42.2 & -83.7 & 57.6 & -0.116 \\ -5.29 & -0.035 & 48.5 & 0.270 \end{bmatrix},$$

and

$$T\tilde{B} = \begin{bmatrix} 20 & 0 \\ 0 & 25 \\ 0 & 0 \end{bmatrix}.$$

The observer is now used to implement the full order system feedback matrix  $\bar{F}$ . The closed loop time response is shown in Figures 5.16-5.25. It is shown that the response of the yaw rate and sideslip angle for this design are more desirable than for the previous design since there is less disturbance and faster settling time for both states. On the other hand, the roll rate and bank angle responses are almost identical for both designs. The controlling surfaces and washout filter states are all well within the physical limitations of the system. This illustrates that a viable constant full state feedback control system can be designed for a large system using a reduced-order model and that this feedback design can be successfully implemented with a reduced-order observer.

### 5.3 Summary and Concluding Remarks

A new design procedure for the control of large systems using reduced-order models, reduced-order observers, and spectral assignment techniques is presented. A reduced-order model is formulated containing

TIME SIMU.

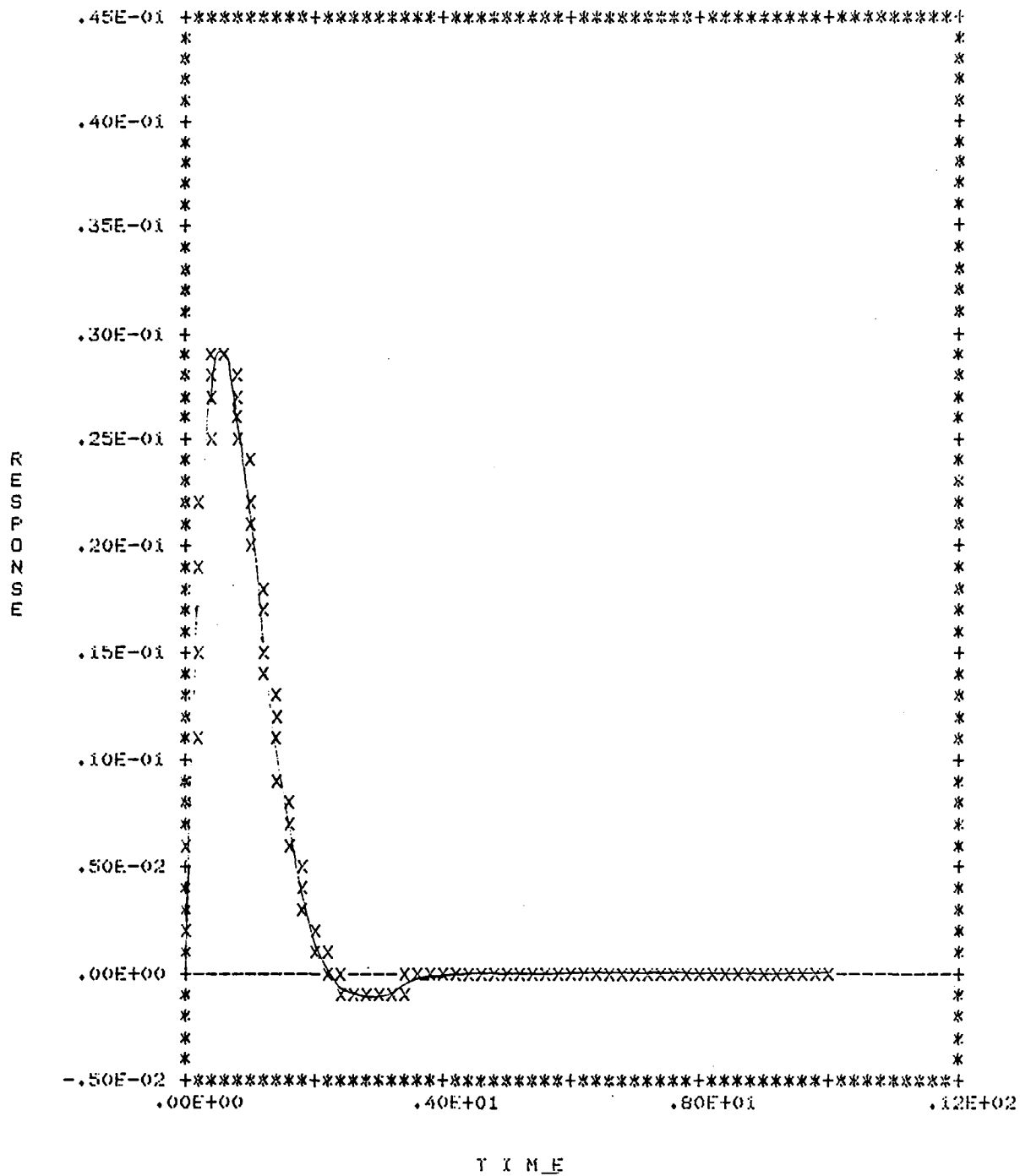


Figure 5.16. Yaw Rate-Second Closed Loop Response for  $\phi(0) = 1^\circ$

TIME SIMU.

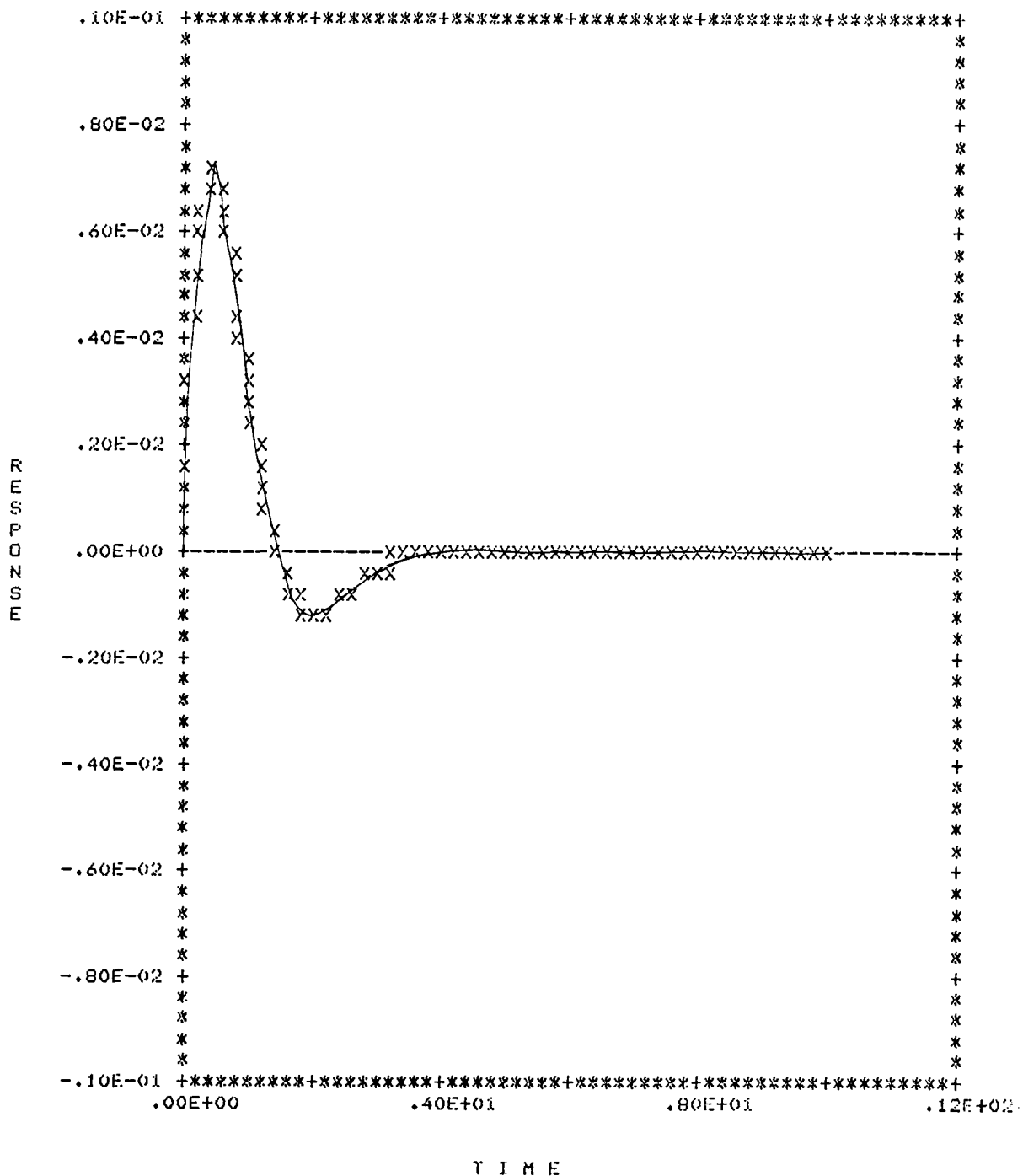


Figure 5.17. Sideslip Angle-Second Closed Loop Response for  $\phi(0) = 1^\circ$

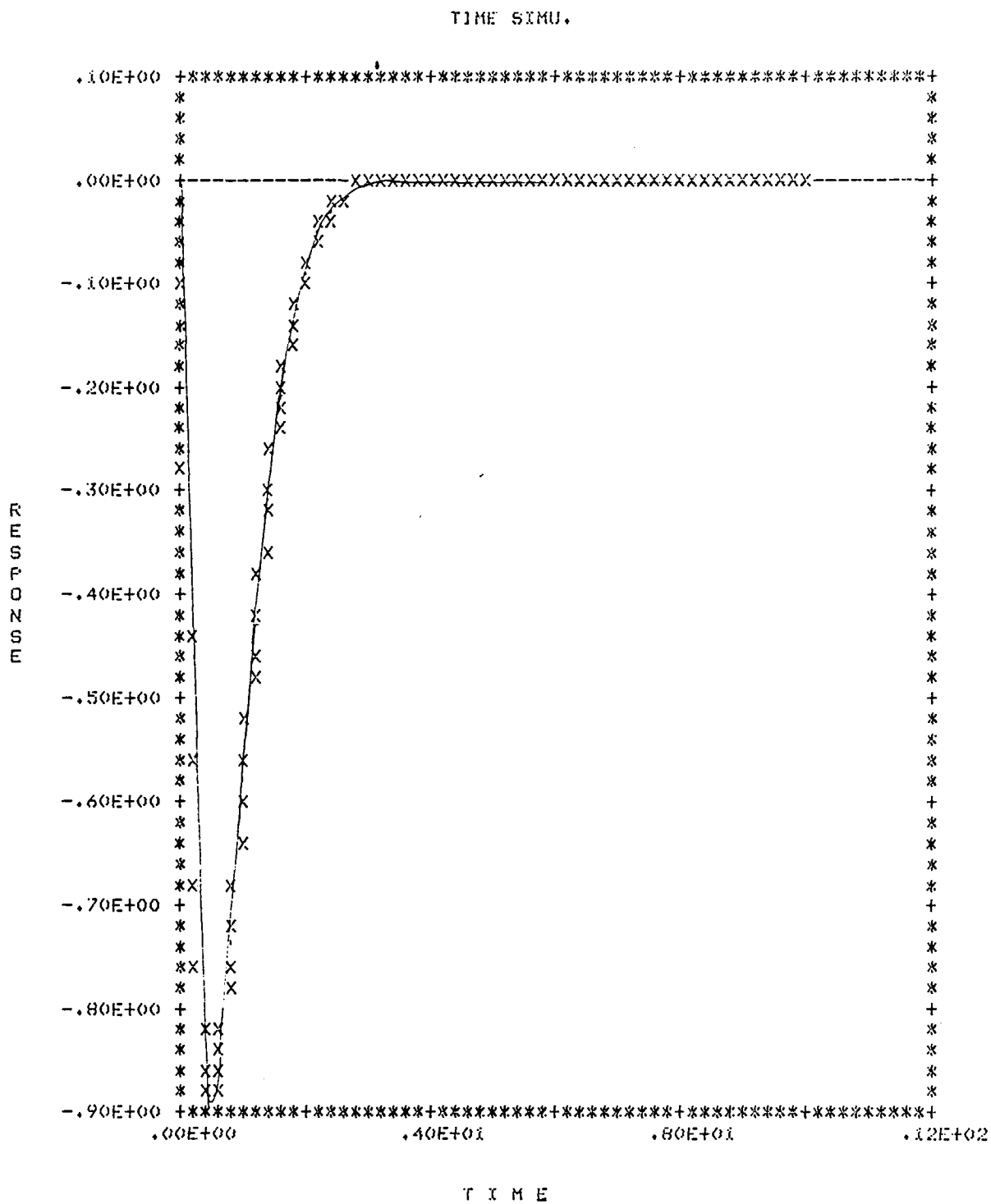


Figure .5.18. Roll Rate-Second Closed Loop Response for  $\phi(0) = 1^\circ$

TIME SIMU.

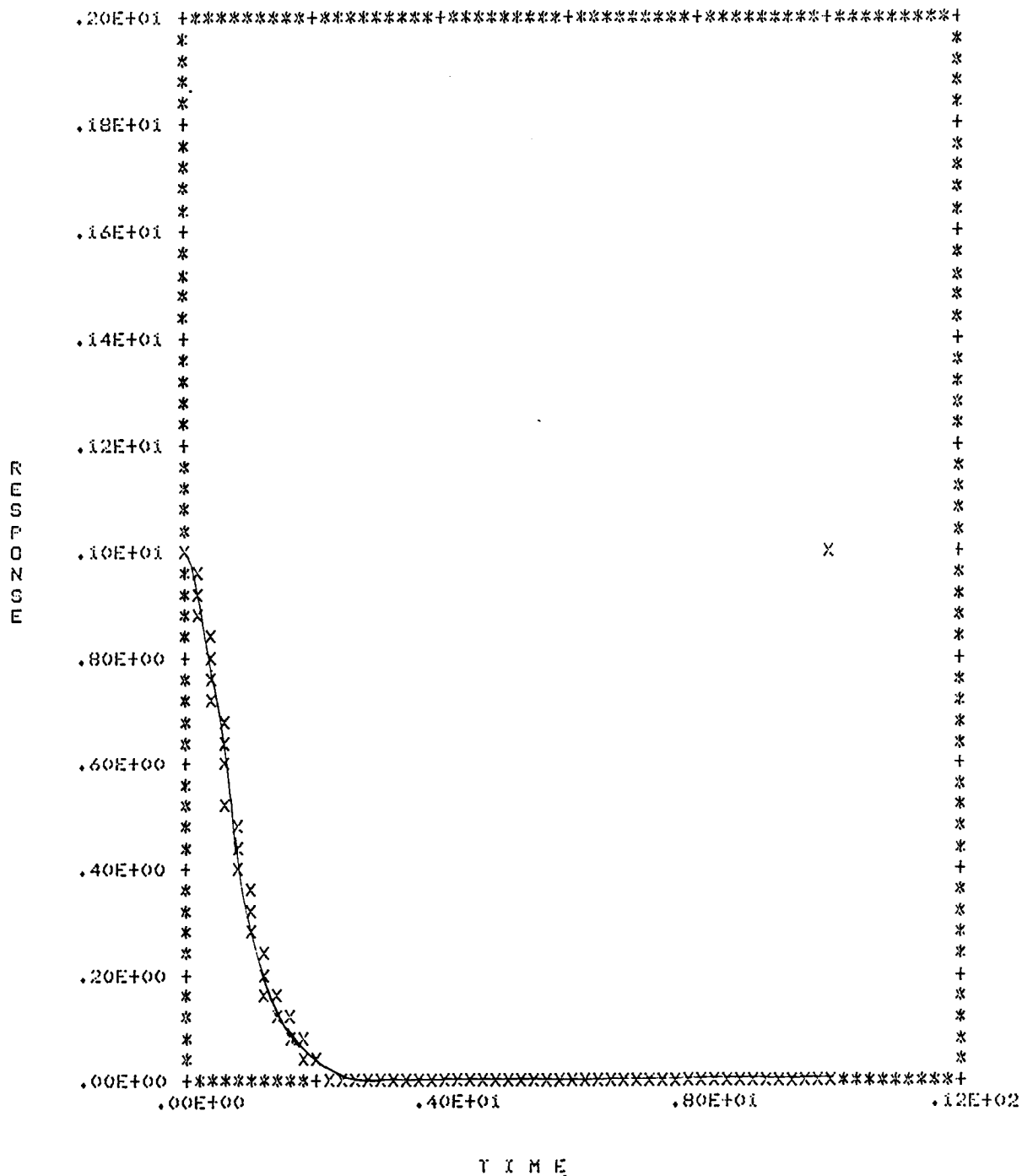


Figure 5.19. Bank Angle-Second Closed Loop Response for  $\phi(0) = 1^\circ$

TIME SIMU.

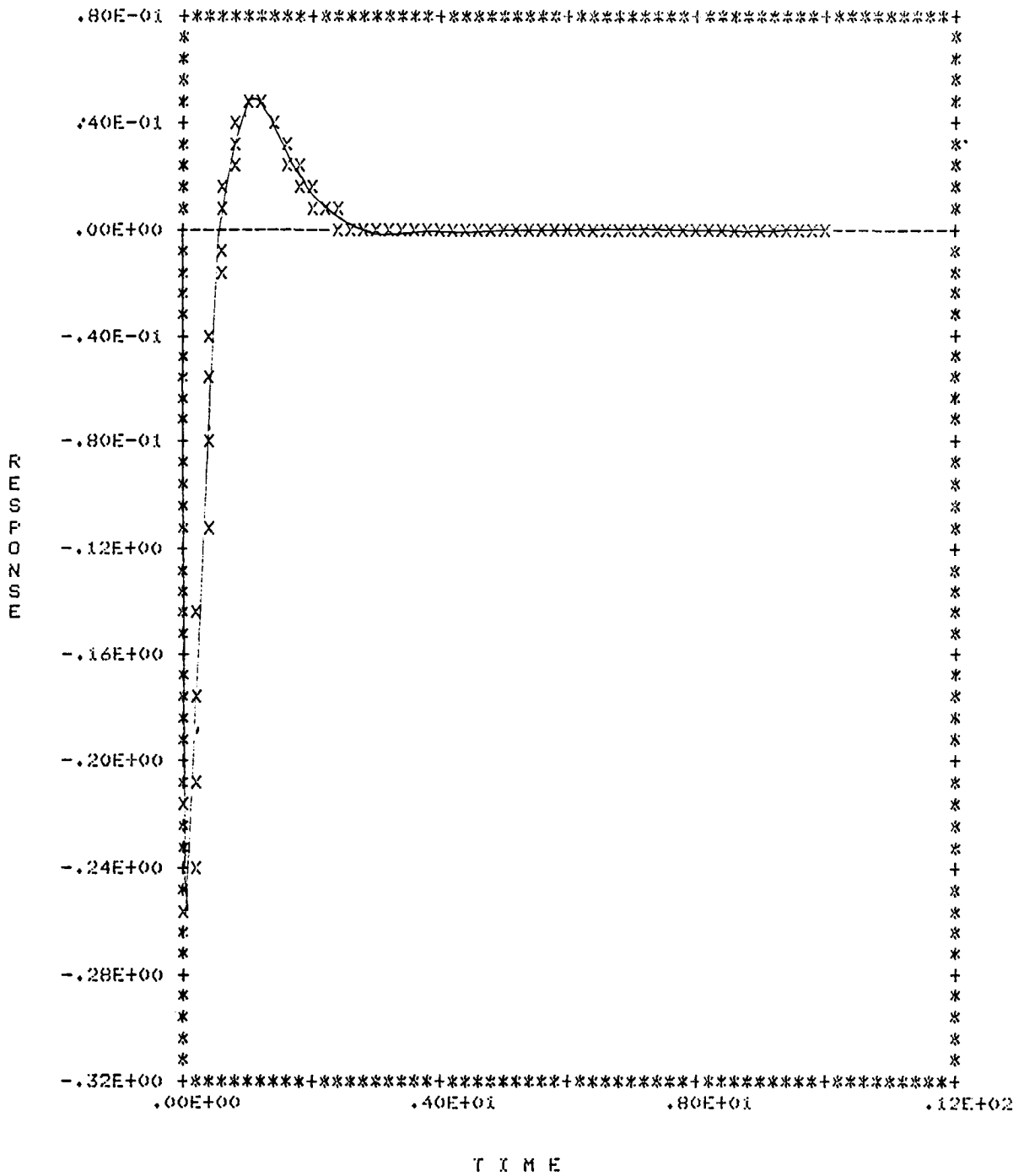


Figure 5.20. Rudder Deflection-Second Closed Loop Response for  $\phi(0) = 1^\circ$

TIME SIMU.

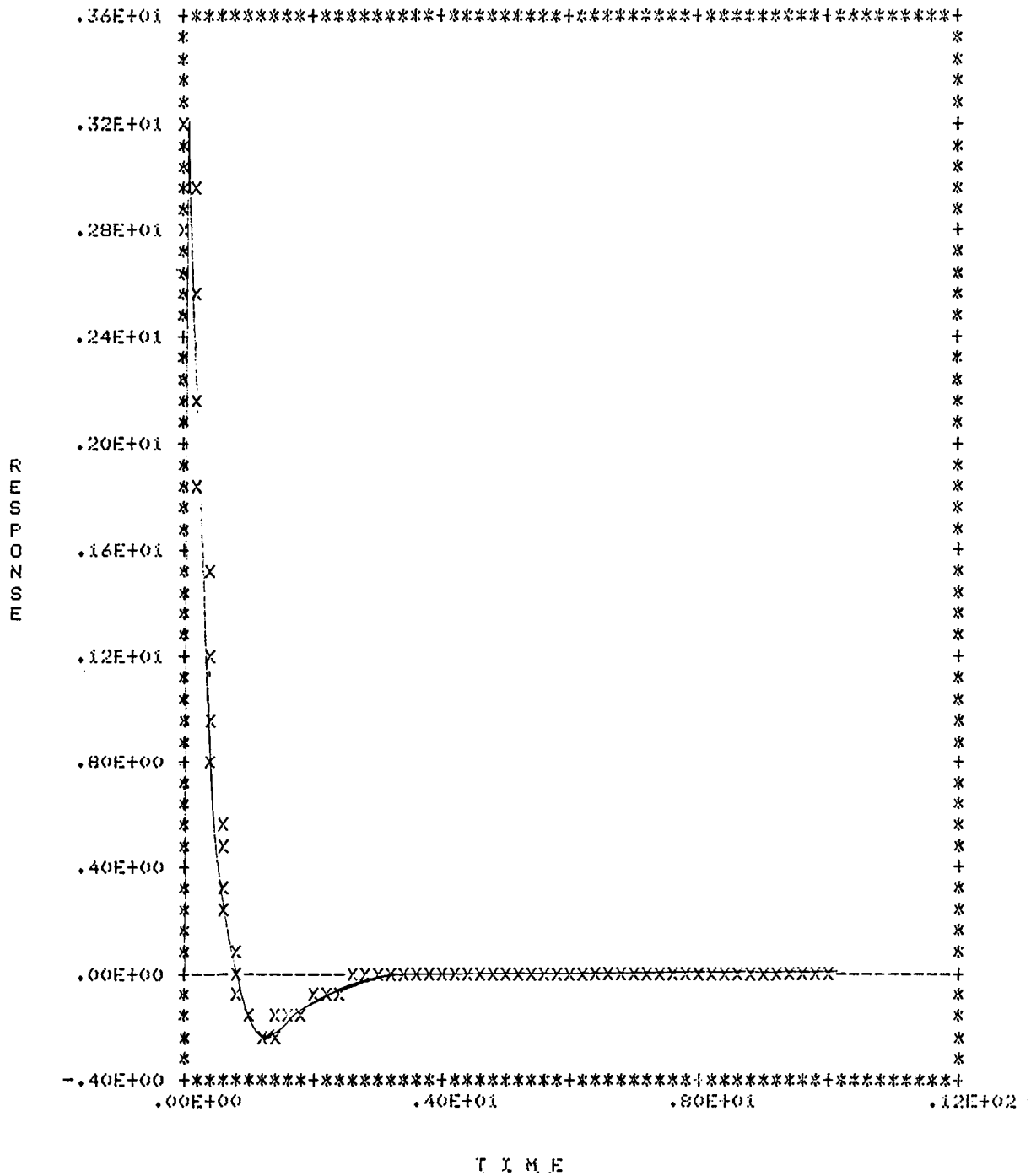


Figure 5.21. Aileron Deflection-Second Closed Loop Response for  $\phi(0) = 1^\circ$



TIME SIMU.

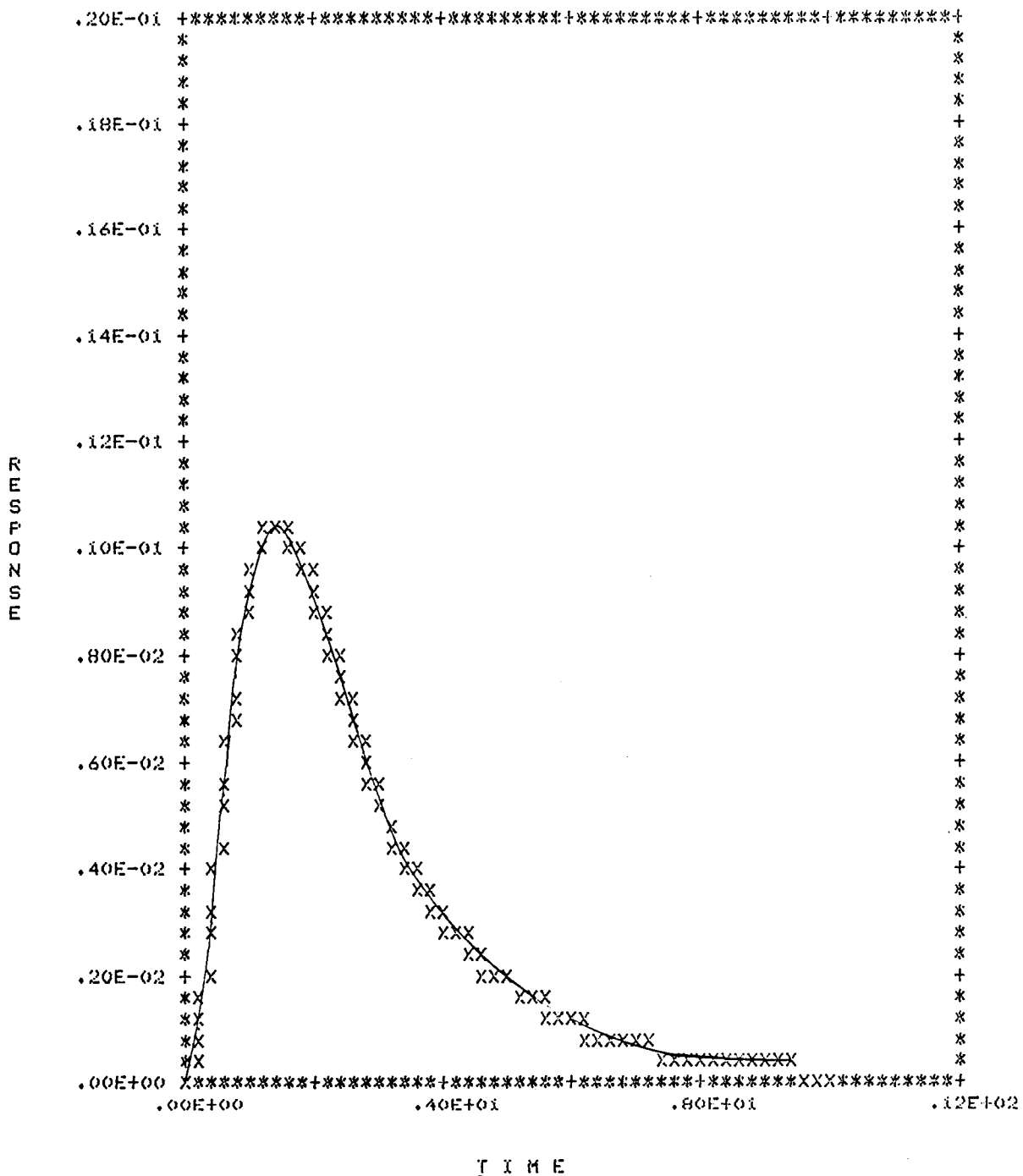


Figure 5.22. Washout Filter-Second Closed Loop Response for  $\phi(0) = 1^\circ$

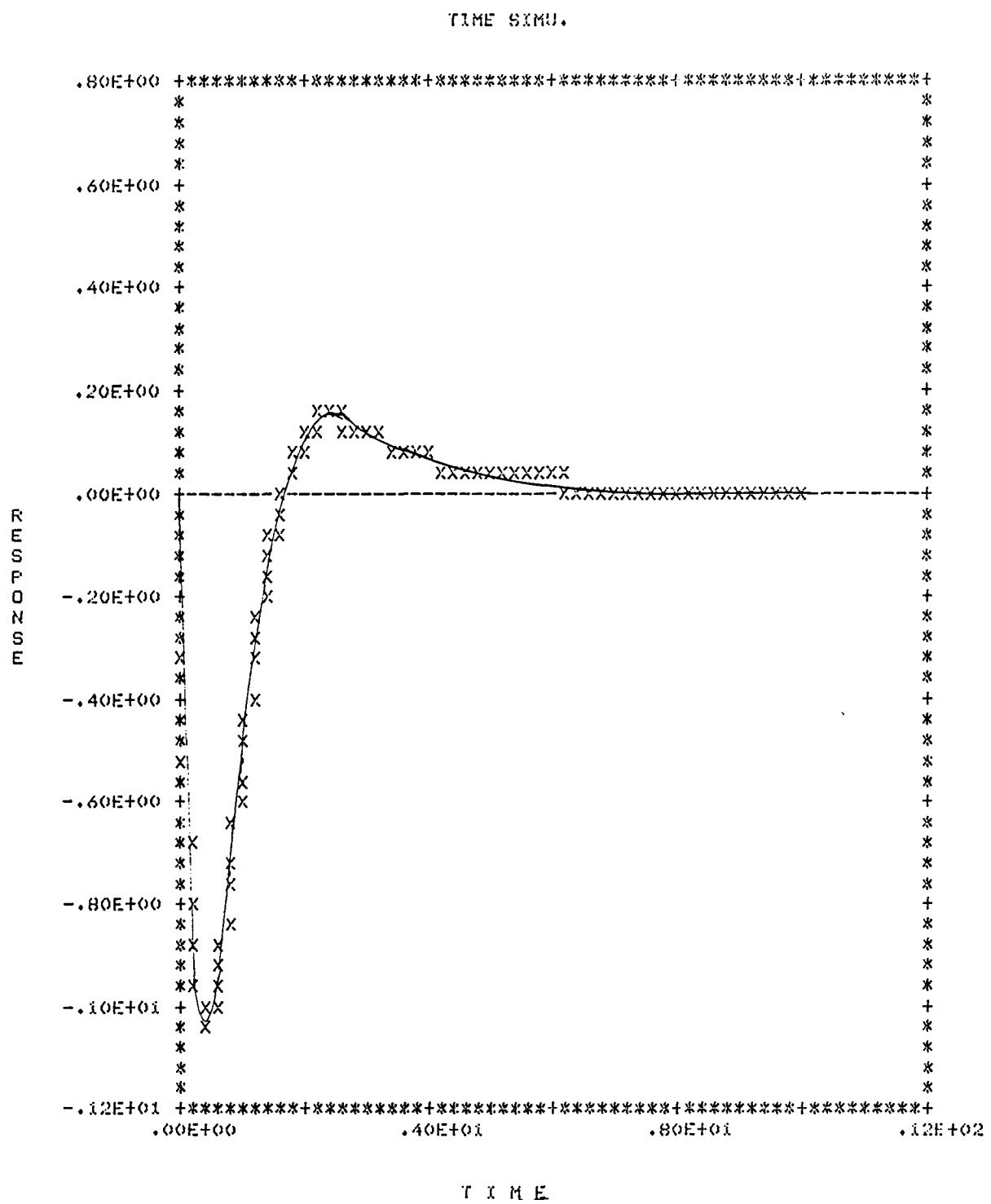


Figure 5.23. Observer State #1 - Closed Loop Response for  $\phi(0) = 1^\circ$

TIME SIMU.

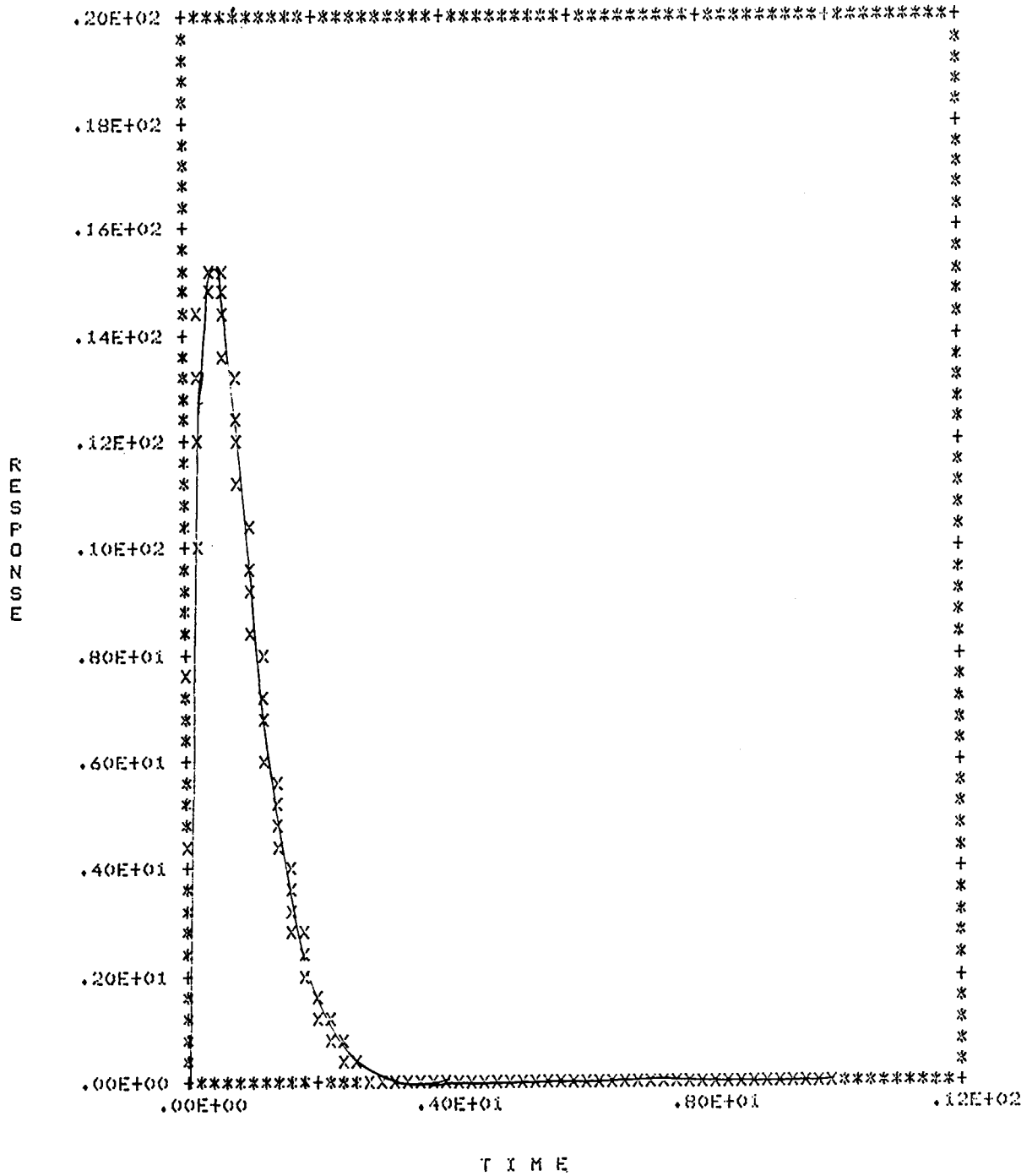


Figure 5.24. Observer State #2 - Closed Loop Response for  $\phi(0) = 1^\circ$

TIME SIMU.

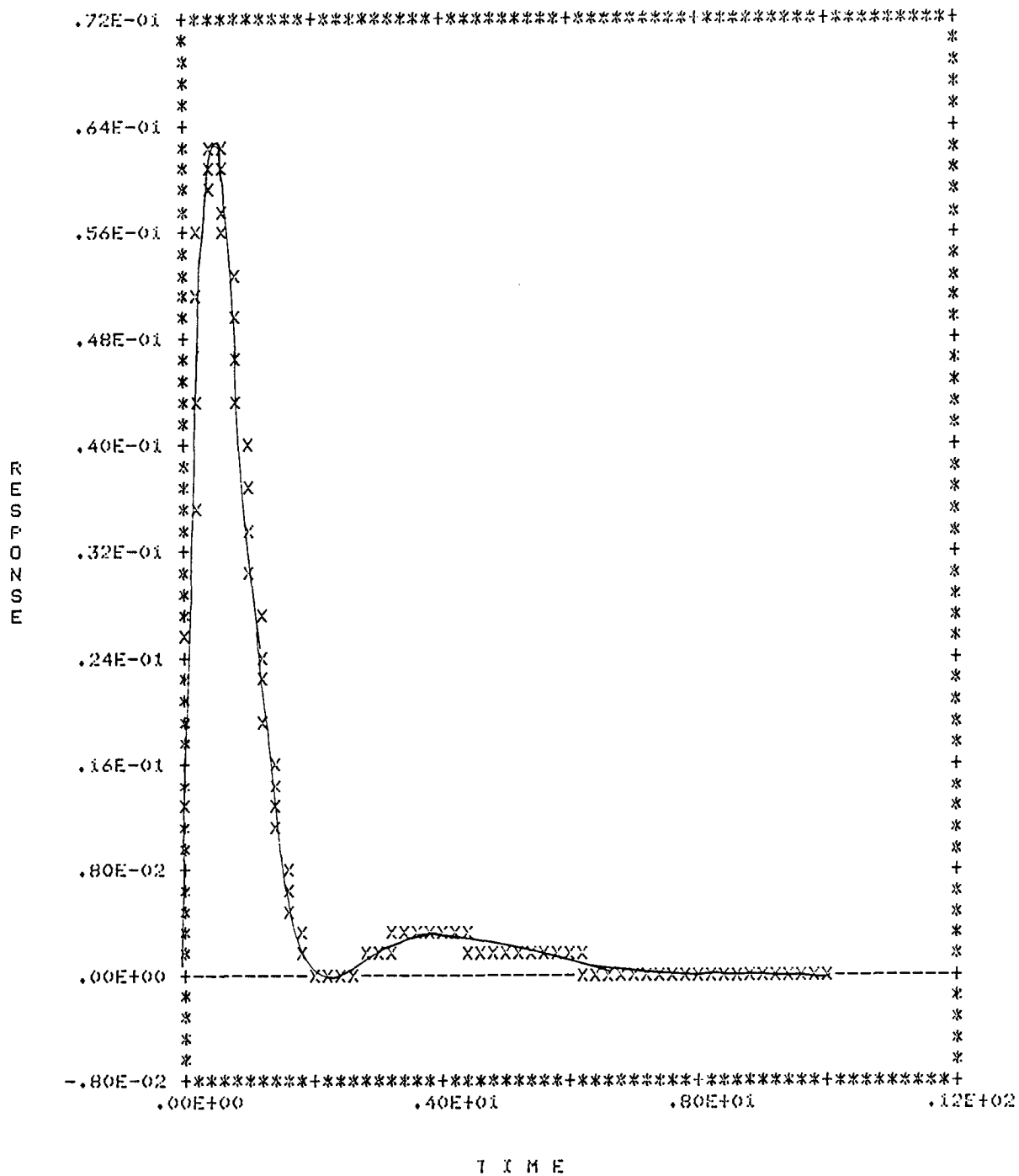


Figure 5.25. Observer State #3 - Closed Loop Response for  $\phi(0) = 1^\circ$

eigenvalues of interest from an original full-order system. A constant state feedback matrix is designed for the reduced-order model that, when implemented about the full-order system, reassigns the eigenvalues contained in the reduced-order model while those eigenvalues not included in the reduced-order model are retained in the full-order system. It is then shown that the full state constant feedback matrix for the original full-order system is implemented by a reduced-order observer if all of the system states are not simultaneously available.

## REFERENCES

- [1] P. I. Carraway and R. R. Mielke, "Spectral Assignment Using Reduced-Order Models," Allerton Conference on Communication, Control, and Computing, Vol. 21, October 1983.
- [2] D. J. Luenberger, "An Introduction to Observers," IEEE Transactions on Automatic Control, Vol. AC-16, No. 6, pp. 596-602, December 1971.
- [3] S. Srinathkumar, "Eigenvalue/Eigenvector Assignment Using Output Feedback," IEEE Transactions on Automatic Control, Vol. AC-23, February 1978.
- [4] B. C. Moore, "On the Flexibility Offered by State Feedback in Multivariable Systems Beyond Closed Loop Eigenvalue Assignment," IEEE Trans. Automatic Control, Vol. AC-21, October 1976.
- [5] H. Kimura, "A Further Result on the Problem on Pole Assignment by Output Feedback," IEEE Trans. Automatic Control, Vol. AC-22, June 1977.
- [6] E. J. Davison and S. H. Wang, "On Pole Assignment in Linear Multivariable Systems," IEEE Trans. Automatic Control, Vol. AC-20, August 1975.
- [7] R. A. Maynard, R. R. Mielke, S. R. Liberty, and S. Srinathkumar, "A New Spectral Synthesis Procedure for Multivariable Regulators," Allerton Conference on Communication, Control and Computing, Vol. 16, October 1978.
- [8] R. R. Mielke, R. A. Maynard, S. R. Liberty, and S. Srinathkumar, "Design of Robust State Feedback Controllers Via Eigenvalue/Eigenvector Assignment," Asilomar Conference on Circuits, Systems and Computers, Vol. 12, November 1978.
- [9] R. A. Maynard, R. R. Mielke and S. R. Liberty, "Eigenvector Assignment for Noise Suppression," Southeastern Symposium on System Theory, Vol. 13, March 1981.
- [10] T. M. Owens and R. R. Mielke, "A New Gain Modification Procedure Via Spectral Assignment," IEEE Proceedings of Southeastern Symposium on System Theory, Vol. 14, April 1982.

- [11] R. R. Mielke, L. J. Tung and M. Marefat, "Design of Multivariable Feedback Controllers Via Spectral Assignment," NASA CR-169889, 1983.
- [12] L. Zadeh and C. Desoer, Linear System Theory - A State-Space Approach, McGraw Hill, New York, 1963.
- [13] T. Kailath, Linear Systems, Prentice Hall, Englewood Cliffs, New Jersey, 1980.
- [14] W. M. Wonham, "On Pole Assignment in Multi-input, Controllable Linear Systems," IEEE Transactions on Automatic Control, Vol. AC-12, pp. 660-665, 1967.
- [15] A. Andry, E. Shapiro and J. Chung, "On Eigenstructure Assignment for Linear Systems," IEEE Trans. on Aerospace and Electronic Systems, September 1983.

## APPENDICES

Appendix A contains a software listing of the modifications to the spectral assignment computer aided design package discussed in Chapter 4. This is followed by an example of an interactive design session in Appendix B.



# APPENDIX A: SOFTWARE LISTING

```

SUBROUTINE HODEY
C-FUNCTION: HATH ROUTINE FOR COMPONENT MODIFICATION
C-INSL ROUTINES CALLED: DENSET,USWFH.
C-SPECTRAL ASSIGNMENT ROUTINES: ROGRAD,ROCOST,SEARCH,DSPLAY,VACT.
C-LOGICAL DEVICES: INPUT UNIT: 5 OUTPUT UNIT: 5
C- STORAGE UNIT(S): IU=20,IU=20+J TAPE J=1,NS,(UT=20+NS)I
INTEGER ORIGO,ILR(10)
COMPLEX WEIB(10),Z(10,10)
REAL LROR(10),LIOR(10),VO(10,10),VFS(10,10),VDES(10,10)
REAL BLAM2(10,10),VT(10),VOIRV(10,10),ALAM(10,10),BLAM(10,10)
REAL ALAM1(10,10),ALAM2(10,10),ALAM21(10,10),ALAM22(10,10)
REAL M(10,10),FTIL1(10,10),FTIL2(10,10),L(10,10),EOPS(10,10)
REAL BLAM1(10,10),CLAM1(10,10),CLAM2(10,10),FLAM(10,10)
C-RANDOM ACCESS FILE: SYSTEM,FORXX WHERE XX=20+J ,CURRHT
REAL AL(10,10),B(10,10),CLAM(10,10),MIRV(10,10)
REAL XX(10,10),VA(20),E(20),X(20),LRE(10),LIM(10),WJ(10)
REAL W(10,10),V(10,10),VIRV(10,10),F(10,10),AHAT(10,10)
REAL A(10,10),B(10,10),C(10,10),WKAREA(130)
COMMON/SYS/A,B,C,ZERO,IDUT,NS,NI,NO
COMMON/ROD/F,AHAT/EJO/LRE,LIM/PAR/AL/OR/O
COMMON/VEC/VA,E,X,WJ,W,XX,V,VINU
COMMON/RO/ORIGO,VO,VFS,VDES,BLAM2,LROR,LIOR,ALAM22
DIMENSION CHAR(2)
EXTERNAL ROCOST,ROGRAD,MODE2,MODE3,MODE4
CALL DENSET(3,LEVOLD)
OPEN(FILES='ROINFO',ACCESS='DIRECT',RECL=102,UNIT=32)
IU=20
READ(IU,REC=1)NS,NI,NO,IDUT,ZERO
READ(IU,REC=2)((A(I,I),I,J=1,NS),II=1,NS)
READ(IU,REC=3)((B(I,I),I,J=1,NI),II=1,NS)
READ(IU,REC=4)((C(I,I),I,J=1,NS),II=1,NO)
C***** ENTER ORIGINAL EIGENVECTORS *****
WRITE(6,8)
0 FORMAT(1X,41H WANT TO ENTER NEW ORIGINAL EIGENVECTORS?)
READ(5,*)KK
IF(KK.LE.0)GO TO 61
J=1
7 WRITE(6,5)J
5 FORMAT(20HENTER ORIGINAL EIGENVECTOR V,12)
READ(5,*)(VO(I,J),VT(I),I=1,NS)
CHTEST=0.
DO 40 I=1,NS
CHTEST=CHTEST+VT(I)**2
40 CONTINUE
IF(CHTEST.LE.0.)GO TO 52
DO 7 I=1,NS
VO(I,J+1)=VT(I)
6 CONTINUE
J=J+1
52 J=J+1
IF(J.LE.NS)GO TO 7
WRITE(32,REC=1)((VO(II,I),I,J=1,NS),II=1,NS)
61 CONTINUE
READ(32,REC=1)((VO(II,I),I,J=1,NS),II=1,NS)
615 WRITE(32,REC=1)((VO(II,I),I,J=1,NS),II=1,NS)
CALL USWFH(9H MATRIX 1,9,VO,10,NS,NS,4)
WRITE(6,612)
612 FORMAT(1X,31HWISH TO CHANGE ANY VALUES OF V?)
READ(5,*)KK
IF(KK.LE.0)GO TO 613
WRITE(6,614)
614 FORMAT(1X,22HENTER I,J, & NEW VALUE)
READ(5,*)I,J,VO(I,J)
GO TO 615
C***** TRANSFORM SYSTEM BY MODAL MATRIX *****
613 CALL LYR2F(VO,NS,10,VOIRV,10,WT,WKAREA,IER)
PAUSE 1
C CALL USWFH(6HVOIRV1,6,VOIRV,10,NS,NS,4)
CALL VHULFF(VOIRV,A,NS,NS,NS,10,10,W,10,ZER)
CALL VHULFF(W,VO,NS,NS,NS,10,10,ALAM,10,IER)
DO 616 I=1,NS
DO 616 J=1,NS
ABSAL=ABS(ALAM(I,J))
IF(ABSAL.LT.ZERO)ALAM(I,J)=FLOAT(0)
616 CONTINUE
C PAUSE 2
CALL USWFH(5HBLAM1,5,ALAM,10,NS,NS,4)
CALL VHULFF(VOIRV,B,NS,NS,NI,10,10,BLAM,10,IER)
C PAUSE 3
CALL USWFH(5HBLAM1,5,BLAM,10,NS,NI,4)
CALL VHULFF(C,VO,NO,NS,NS,10,10,CLAM,10,IER)
C PAUSE 4
C CALL USWFH(5HCLAM1,5,CLAM,10,NO,NS,4)
C***** PARTITION TILDA SYSTEM *****
DO 600 I=1,NO
DO 600 J=1,NO
ALAM1(I,J)=ALAM(I,J)
600 CONTINUE
DO 599 I=1,NS
DO 599 J=1,NS
IF((I.GT.NO).AND.(J.GT.NO))ALAM22(I-NO,J-NO)=ALAM(I,J)
599 CONTINUE
DO 601 I=1,NO
DO 601 J=1,NI
BLAM1(I,J)=BLAM(I,J)
601 CONTINUE
DO 602 I=1,NO
DO 602 J=1,NO
CLAM1(I,J)=CLAM(I,J)
602 CONTINUE
C***** DISPLAY RO MODEL *****
WRITE(6,804)
804 FORMAT(1X,25HWISH TO DISPLAY RO MODEL?)
READ(5,*)KK
IF(KK.LE.0)GO TO 805
WRITE(6,603)
603 FORMAT(1X,///,20H REDUCED ORDER MODEL,/)
CALL USWFH(9HMATRIX A1,9,ALAM1,10,NO,NO,4)
CALL USWFH(9HMATRIX B1,9,BLAM1,10,NO,NI,4)
CALL USWFH(9HMATRIX C1,9,CLAM1,10,NO,NO,4)
C***** TEMPORARILY STORE ORIGINAL SYSTEM *****
805 WRITE(32,REC=2)NS,NI,NO
WRITE(32,REC=3)((A(II,I),I,J=1,NS),II=1,NS)
WRITE(32,REC=4)((B(II,I),I,J=1,NI),II=1,NS)
WRITE(32,REC=5)((C(II,I),I,J=1,NS),II=1,NO)
C***** STORE RO MODEL INTO FILE #20 *****

```

```

WRITE(20,REC=1)NO,N1,NO,IDOT,ZERO
WRITE(20,REC=2)((ALAM1)(I,I),I,J=1,NO),II=1,NO)
WRITE(20,REC=3)((BLAM1)(I,I),I,J=1,NI),II=1,NO)
WRITE(20,REC=4)((CLAM1)(I,I),I,J=1,NO),II=1,NO)
C***** GO TO MODE2 AND ASSIGN EIGENVALUES FOR RD SYSTEM *****
WRITE(6,604)
604 FORMAT(1X,18(1H*),36H REDUCED ORDER EIGENVALUE ASSIGNMENT,17(1H*))
CALL MODE2
C***** ENTER DESIRED PARTIAL EIGENVECTOR ASSIGNMENT *****
WRITE(6,605)
605 FORMAT(1X,44HENTER DESIRED PARTIAL EIGENVECTOR ASSIGNMENT)
J=1
609 WRITE(6,606)J
606 FORMAT(1X,13HEIGENVECTOR V, I2)
READ(5,*) (VDES(I,J),VT(I),I=1,NO)
IF(ABS(LIM(J)).LT.ZERO)GO TO 608
DO 607 I=1,NO
VDES(I,J+1)=VT(I)
607 CONTINUE
J=J+1
608 J=J+1
IF(J.LE.NO)GO TO 609
C***** CALCULATE AND DISPLAY INITIAL GUESS *****
DO 610 I=1,NO
DO 610 J=1,NO
G(I,J)=VDES(I,J)
610 CONTINUE
CALL LINV2F(8,NO,10,AHAT,IDOT,WKAREA,IER)
C PAUSE 5
C CALL USWFM(7HV11INV1,7,AHAT,10,NO,NO,4)
CALL UHULFF(AHAT,VDES,NO,NO,NO,10,10,8,10,IER)
WRITE(6,611)
611 FORMAT(1X,49HUSE THE FOLLOWING V MATRIX FOR INITIAL ASSIGNMENT,/,
129H REMEMBER WHICH V ARE COMPLEX)
CHAR(1)='INITIAL GU'
CHAR(2)='ESS FOR V1'
CALL USWFM(CHAR,20,8,10,NO,NO,4)
C***** ASSIGN INITIAL EIGENVECTOR GUESS *****
CALL MODE3
IU=20
READ(IU,REC=1)NS,N1,NO,IDOT,ZERO
READ(IU,REC=2)((A(I,I),I,J=1,NS),II=1,NS)
READ(IU,REC=3)((B(I,I),I,J=1,NI),II=1,NS)
READ(IU,REC=4)((C(I,I),I,J=1,NS),II=1,NO)
C PAUSE 6
WRITE(6,810)
810 FORMAT(1X,37HTHE FOLLOWING IS RD MODEL AFTER MODE3)
C CALL USWFM(2HA1,2,A,10,NS,NS,4)
C CALL USWFM(2HB1,2,B,10,NS,NI,4)
C PAUSE 7
C CALL USWFM(2HC1,2,C,10,NO,NS,4)
DO 10 J=1,NS
IU=20+J
OPEN(ACCESS='DIRECT',RECL=202
1,UNIT=IU)
READ(IU,REC=1)LRE(J),LIM(J)
10 CONTINUE
PAUSE 8
DO 102 I=1,NS
C WRITE(6,811)I,LRE(I),LIM(I)
C 811 FORMAT(1X,61HLAMBDA, I2, 3X, 5HREAL =, F6.3, 3X, 5HIMAG =, F6.3)
812 CONTINUE
IU=20+NS+1
OPEN(FILE='CURRN',ACCESS='DIRECT',RECL=102
1,UNIT=IU)
READ(IUT,REC=1)((V(I,I),I,J=1,NS),II=1,NS)
READ(IUT,REC=2)((XX(I,I),I,J=1,NS),II=1,NI)
READ(IUT,REC=4)((F(I,I),I,J=1,NS),II=1,NI)
READ(IUT,REC=5)((AHAT(I,I),I,J=1,NS),II=1,NS)
C CALL USWFM(10HMATRIX V 1,10,V,10,NS,NS,4)
C CALL USWFM(10HMATRIX XXI,10,XX,10,NI,NS,4)
PAUSE 9
CHAR(1)='GAIN MATRI'
CHAR(2)='X F1'
CALL USWFM(CHAR,14,F,10,NI,NS,4)
C PAUSE 10
CHAR(1)='MATRIX AHA'
CHAR(2)='T1'
C CALL USWFM(CHAR,12,AHAT,10,NS,NS,4)
READ(32,REC=2)ORIGD,NIX,NOX
I2=ORIGD-NS
C***** ASSIGN WEIGHTING CONSTANTS FOR GRADIENT SEARCH *****
905 DO 70 I=1,NS
DO 70 J=1,NS
AL(I,J)=FLOAT(1)
70 CONTINUE
WRITE(6,99)
99 FORMAT(1X,20(1H*),27H RD EIGENVECTOR IMPROVEMENT,20(1H*))
WRITE(6,9)
9 FORMAT(1X,20H WEIGHTING CONSTANTS)
CALL USWFM(8HWIGHTS1,8,AL,10,NS,NS,4)
WRITE(6,11)
11 FORMAT(1X,16H WISH TO CHANGE?)
READ(5,*)KK
IF(KK.LE.0)GO TO 30
WRITE(6,3)
3 FORMAT(1X,17HENTER NEW VALUE81)
READ(5,*)((AL(I,J),J=1,NS),I=1,NS)
C***** CONDUCT GRADIENT SEARCH *****
30 CALL ROCOST(CJ)
WRITE(6,4)CJ
4 FORMAT(1X,5HCOST=,E15.6)
CALL ROGRAD
CALL SEARCH(CJ,ROCOST,ROGRAD,5)
WRITE(IUT,REC=1)((V(I,I),I,J=1,NS),II=1,NS)
WRITE(IUT,REC=2)((XX(I,I),I,J=1,NS),II=1,NI)
WRITE(IUT,REC=4)((F(I,I),I,J=1,NS),II=1,NI)
WRITE(IUT,REC=5)((AHAT(I,I),I,J=1,NS),II=1,NS)
CALL USWFM(10HMATRIX V 1,10,V,10,NS,NS,4)
CALL VACT
CALL USWFM(10HMATRIX VFS,10,VFS,10,NS,NS,4)
WRITE(6,904)
904 FORMAT(1X,24HWANT TO CONTINUE SEARCH?)
READ(5,*)KK
IF(KK.GT.0)GO TO 905
WRITE(6,902)
902 FORMAT(1X,44HWISH TO DISPLAY THE NORMALIZED EIGENVECTORS?)
C READ(5,*)KS
C IF(KS.LE.0)GO TO 903

```

```

C CALL DSPLAY (NS,ZERO)
703 CONTINUE
C CALL USWFH (10HMATRIX XX1,10,XX,10,NI,NS,4)
CHAR(1)=' GAIN MA'
CHAR(2)='TRIX F1'
CALL USWFH (CHAR,14,F,10,NI,NS,6)
CHAR(1)='MATRIX AHA'
CALL EIGKF(AHAT,NS,10,2,WEIG,Z,10,WKAREA,IER)
C CALL USWCV(BHLAM-GRAD,8,WEIG,NS,1,4)
CHAR(2)='T1'
CALL USWFH (CHAR,12,AHAT,10,NS,NS,4)
***** APPEND ZEROS TO FEEDBACK MATRIX AND XFORM TO ORIGINAL COG
DO 701 I=1,NI
DO 701 J=NO+1,ORIGO
F(I,J)=FLOAT(0)

READ(32,REC=4)((B(I,I),I,J=1,NI),I=1,NS)
READ(32,REC=5)((C(I,I),I,J=1,NS),I=1,NO)
READ(32,REC=6)((F(I,I),I,J=1,NS),I=1,NI)
CALL USWFH(SHF-AFT,5,F,10,NI,NS,4)
***** COMPUTE XFORM MATRIX M & M1RV *****
DO 703 I=1,NS
DO 703 J=1,NS
M(I,J)=FLOAT(0)
IF(I.EQ.J)M(I,J)=FLOAT(1)
IF(I.LE.NO)M(I,J)=C(I,J)
703 CONTINUE
CALL LINZF(M,NS,10,M1RV,IDOT,WKAREA,IER)
C PAUSE 13
C CALL USWFH(SHM1RV1,5,M1RV,10,NS,NS,4)
C CALL USWFH(2HM1,2,M,10,NS,NS,4)
***** XFORM SYSTEM AND F MATRIX *****
CALL VHULFF(M,A,NS,NS,NS,10,10,AHAT,10,IER)
CALL VHULFF(AHAT,M1RV,NS,NS,NS,10,10,ALAM,10,IER)
CALL VHULFF(M,D,NS,NS,NI,10,10,BLAM,10,IER)
CALL VHULFF(C,M1RV,NO,NS,NS,10,10,CLAM,10,IER)
CALL VHULFF(F,M1RV,NI,NS,NS,10,10,AHAT,10,IER)
DO 848 I=1,NI
DO 848 J=1,NS
FLAM(I,J)=AHAT(I,J)
848 CONTINUE
***** DISPLAY XFORMED SYSTEM *****
WRITE(6,801)
801 FORMAT(1X,30H WISH TO DISPLAY TILDA SYSTEM?)
READ(5,*)KK
IF(KK.LE.0)GO TO 802
WRITE(6,704)
704 FORMAT(1X,20H SYSTEM XFORMED BY M,/)
C PAUSE 15
CALL USWFH(7HA TILDA,7,ALAM,10,NS,NS,4)
CALL USWFH(7HD TILDA,7,BLAM,10,NS,NI,4)
PAUSE 14
CALL USWFH(7HC TILDA,7,CLAM,10,NO,NS,4)
CALL USWFH(7HF TILDA,7,AHAT,10,NI,NS,4)
***** PARTITION THE SYSTEM *****
802 DO 705 I=1,NO
DO 705 J=1,NO
ALAM1(I,J)=ALAM(I,J)
IF(I+NO.GT.NS)GO TO 823
ALAM21(I,J)=ALAM(I+NO,J)
701 CONTINUE
C PAUSE 11
C CALL USWFH(10HFAPPEDED1,10,F,10,NI,ORIGO,4)
CALL VHULFF(F,VOIDRV,NI,ORIGO,ORIGO,10,10,AHAT,10,IER)
DO 702 I=1,NI
DO 702 J=1,ORIGO
F(I,J)=AHAT(I,J)
702 CONTINUE
PAUSE 12
CALL USWFH(10HF XFORMED1,10,F,10,NI,ORIGO,4)
WRITE(32,REC=6)((F(I,I),I,J=1,ORIGO),I=1,NI)
C CALL USWFH(2HF1,2,F,10,NI,ORIGO,4)
***** RETRIEVE ORIGINAL SYSTEM DATA & STORE F *****
699 READ(32,REC=2)NS,NI,NO
READ(32,REC=3)((A(I,I),I,J=1,NS),I=1,NS)
823 IF(J+NO.GT.NS)GO TO 824
ALAM12(I,J)=ALAM(I,J+NO)
824 IF((I+NO.GT.NS).AND.(J+NO.GT.NS))GO TO 705

ALAM22(I,J)=ALAM(I+NO,J+NO)
705 CONTINUE
C PAUSE 40
C CALL USWFH(7HATIL121,7,ALAM12,10,NO,I2,4)
C CALL USWFH(7HATIL221,7,ALAM22,10,I2,I2,4)
DO 706 I=1,NO
DO 706 J=1,NI
BLAM1(I,J)=BLAM(I,J)
IF(I+NO.LE.NS)BLAM2(I,J)=BLAM(I+NO,J)
706 CONTINUE
DO 707 I=1,NO
DO 707 J=1,NO
CLAM1(I,J)=CLAM(I,J)
IF(J+NO.LE.NS)CLAM2(I,J)=CLAM(I,J+NO)
707 CONTINUE
DO 708 I=1,NI
DO 708 J=1,NO
FTIL1(I,J)=AHAT(I,J)
IF(J+NO.LE.NS)FTIL2(I,J)=AHAT(I,J+NO)
708 CONTINUE
***** TRANSPOSE ALAM22 & -ALAM12 TO ASSIGN OBSERVER DYNAMICS
WRITE(6,822)NS,NO
822 FORMAT(1X,30H NS=,I2,3X,30H NO=,I2)
NOBS=NS-NO
DO 709 I=1,NO
DO 709 J=1,NOBS
ALAM12(I,J)=-ALAM12(I,J)
709 CONTINUE
C PAUSE 16
C CALL USWFH(8H-ALAM121,8,ALAM12,10,NO,NOBS,4)
C CALL USWFH(7HALAM221,7,ALAM22,10,NOBS,NOBS,4)
CALL TRANS1(ALAM22,NOBS,NOBS)
CALL TRANS1(ALAM12,NO,NOBS)
C PAUSE 17
C CALL USWFH(9H-ALAM12Y1,9,ALAM12,10,NOBS,NO,4)
C CALL USWFH(8HALAM22Y1,8,ALAM22,10,NOBS,NOBS,4)
IF(NO.LE.NOBS)GO TO 831
IFLF=1

```

```

      CALL USWFH(9H-ALAM12T1,9,ALAM12,10,NOBS,RO,4)
      WRITE(6,833)NOBS
033 FORMAT(1X,21HYOU MUST SELECT WHICH,12,42H OUTPUTS WILL BE USED TO
      IFED THE OBSERVER)
      WRITE(6,834)
034 FORMAT(1X,45HTHE OUTPUTS CORRESPOND TO COLUMNS IN -ALAM12)
037 WRITE(6,835)
035 FORMAT(1X,60HSELECTED COLUMNS WILL FORM A MATRIX THAT MUST BE NON
      SINGULAR,/,34HENTER OUTPUTS TO BE USED (INTEGER))
      READ(5,*) (ILR(I),I=1,NOBS)
      DO 836 J=1,NOBS
      DO 836 I=1,NOBS
      AHAT(I,J)=ALAM12(I,ILR(J))
036 CONTINUE
      CALL USWFH(6H MOD1,6,AHAT,10,NOBS,ROBS,4)
      IER=0
      CALL LINV2F(AHAT,NOBS,10,AL,IDBT,WKAREA,IER)
      IF (IER.EQ.129)GO TO 837
      NOX=NOBS
      GO TO 845
031 NOX=RO
      DO 838 I=1,NOBS
      DO 838 J=1,NO
      AHAT(I,J)=ALAM12(I,J)
038 CONTINUE
045 CONTINUE
***** STORE ALAM22 & ALAM12 AS A & B AND CHANGE NS *****
      WRITE(20,REC=1)NOBS,NOX,NOBS,IDBT,ZERO
      WRITE(20,REC=2)((ALAM22(I,I),I,J=1,NOBS),I=1,NOBS)
      WRITE(20,REC=3)((AHAT(I,I),I,J=1,NOX),I=1,NOBS)
*****1 ASSIGN EIGENVALUES AND EIGENVECTORS FOR OBSERVER *****
      CLOSE(UNIT=XUT,STATUS='KEEP')
      DO 815 I=1,NO
      J=I+20
      CLOSE(UNIT=J)
015 CONTINUE
      WRITE(6,710)
710 FORMAT(1X,20(1H*),32H ASSIGN EIGENVALUES FOR OBSERVER,10(1H*)
      CALL MODE2
      WRITE(6,711)
711 FORMAT(1X,15(1H*),33H ASSIGN EIGENVECTORS FOR OBSERVER,14(1H*)
      CALL MODE3
      WRITE(6,735)
735 FORMAT(1X,33HDO YOU WANT TO REDUCE GAIN FOR OBSERVER?)
      READ(5,*)KK
      IF(KK.LE.0)GO TO 736
      CALL MODE6
736 CONTINUE
***** RETRANSPOSE ALAM22 & -ALAM12 & XPOSE F TO GET L *****
      NOX=ORIGO-NOBS
      DO 712 I=1,NOBS
      DO 712 J=1,NOX
      ALAM12(I,J)=-ALAM12(I,J)
712 CONTINUE
      CALL TRANS1(ALAM22,NS,NS)
      CALL TRANS1(ALAM12,NOBS,NOX)
C
      PAUSE 18
C
      CALL USWFH(7HALAM121,7,ALAM12,10,NOX,NOBS,4)
C
      CALL USWFH(7HALAM221,7,ALAM22,10,NOBS,NOBS,4)

```

```

      IUT=20+NS+1
      READ(XUT,REC=4)((F(I,I),I,J=1,NS),I=1,NI)
      IF(IFLF.NE.1)GO TO 841
      DO 839 I=1,NOX
      DO 839 J=1,NOBS
      L(I,J)=FLOAT(0)
039 CONTINUE
      DO 840 I=1,NS
      DO 840 J=1,NS
      L(YLR(I),J)=F(I,J)
040 CONTINUE
      GO TO 846
041 DO 842 I=1,NOX
      DO 842 J=1,NOBS
      L(I,J)=F(I,J)
042 CONTINUE
046 CONTINUE
      CLOSE(UNIT=XUT,STATUS='KEEP')
      CALL TRANS1(L,ROX,NS)
      PAUSE 19
      IFLF=0
      CALL USWFH(2HL1,2,L,10,RS,ROX,4)
***** COMPUTE R *****
      READ(32,REC=2)RS,R1,NO
      NOBS=NS-NO
      CALL VHULFF(FYLL2,L,R1,NOBS,RO,10,10,AHAT,10,IER)
      DO 713 I=1,NI
      DO 713 J=1,NO
C R IS REPRESENTED BY FYLL1
      FYLL1(I,J)=FYLL1(I,J)+AHAT(I,J)
713 CONTINUE
***** COMPUTE F *****
      CALL VHULFF(L,ALAM12,NOBS,NO,NOBS,10,10,AHAT,10,IER)
      DO 714 I=1,NOBS
      DO 714 J=1,NOBS
      EOB3(I,J)=ALAM22(I,J)-AHAT(I,J)
714 CONTINUE
***** COMPUTE B *****
      CALL VHULFF(EOB3,L,NOBS,NOBS,RO,10,10,AHAT,10,IER)
      CALL VHULFF(L,ALAM11,NOBS,NO,NO,10,10,ALAM22,10,IER)
      DO 715 I=1,NOBS
      DO 715 J=1,NO
C B IS REPRESENTED BY ALAM22
      ALAM22(I,J)=ALAM21(I,J)-ALAM22(I,J)+AHAT(I,J)
715 CONTINUE
***** COMPUTE B-TILDA *****
      CALL VHULFF(L,BLAM1,NOBS,NO,R1,10,10,AHAT,10,IER)
      DO 716 I=1,NOBS
      DO 716 J=1,NI
C AHAT REPRESENTS T-BTILDA
      AHAT(I,J)=BLAM2(I,J)-AHAT(I,J)
716 CONTINUE
***** DISPLAY R,B,E,& TB *****
      CALL USWFH(9HMATRIX E1,9,EOB3,10,NOBS,NOBS,4)

```

```

CALL USWFH(9HMATRIX R1,9,FTIL1,10,NI,NO,4)
PAUSE 20
CALL USWFH(9HMATRIX O1,9,ALAM22,10,NOBS,NO,4)
CALL USWFH(10HMATRIX TB1,10,AHAT,10,NOBS,NI,4)
C***** CONSTRUCT TOTAL SYSTEM MATRIX ATILDA *****
NS=NOBS+NO
CALL VMULFF(BLAM,FLAM,NS,NI,NS,10,10,AHAT,10,IER)
DO 718 I=1,NS
DO 718 J=1,NS
AHAT(I,J)=AHAT(I,J)+ALAM(I,J)
718 CONTINUE
C CALL EIGRF(AHAT,NS,10,2,WEIO,Z,10,WKAREA,IER)
C CALL USWCV(10EIGENVALUE,10,WEIO,NS,1,4)
CALL VMULFF(BLAM,FTIL2,NS,NI,NOBS,10,10,FTIL1,10,IER)
DO 719 I=1,NS
DO 719 J=1,NOBS
AHAT(I,J+NS)=FTIL1(I,J)
719 CONTINUE
DO 720 I=1,NOBS
DO 720 J=1,NOBS
AHAT(I+NS,J+NS)=EBS(I,J)
720 CONTINUE
DO 717 I=1,NOBS
DO 717 J=1,NS
AHAT(I+NS,J)=FLOAT(0)
717 CONTINUE
IROW=NS+NOBS
CHAR(1)=' ATILDA TO'
CHAR(2)='TAL'
C CALL USWFH(CHAR,13,AHAT,10,IROW,IROW,4)
C CALL EIGRF(AHAT,IROW,10,2,WEIO,Z,10,WKAREA,IER)
C CALL USWCV(10EIGENVALUE,10,WEIO,NS,1,4)
C***** CONSTRUCT XFORM TO GET X & W COORDINATES ***
C
C FIRST MODIFY L TO BECOME T=L-L11
C
DO 721 I=1,NOBS
DO 721 J=1,NO
L(I,J)=-L(I,J)

IF(J.GT.NS)GO TO 721
IF(J.EQ.I)L(I,J+NO)=FLOAT(1)
721 CONTINUE
C
C NOW CONSTRUCT M-INVERSE
C
DO 722 I=1,IROW
DO 722 J=NS+1,IROW
MINV(I,J)=FLOAT(0)
IF (I.EQ.J) MINV(I,J)=FLOAT(1)
722 CONTINUE
DO 724 I=1,NOBS
DO 724 J=1,NS
MINV(I+NS,J)=L(I,J)
724 CONTINUE
C
C CONSTRUCT M

```

```

C
DO 725 I=1,IROW
DO 725 J=NS+1,IROW
M(I,J)=FLOAT(0)
IF (I.EQ.J)M(I,J)=FLOAT(1)
725 CONTINUE
DO 724 I=1,NOBS
DO 724 J=1,NS
L(I,J)=-L(I,J)
724 CONTINUE
CALL VMULFF(L,M,NOBS,NS,NS,10,10,FTIL1,10,IER)
DO 727 I=1,NOBS
DO 727 J=1,NS
M(I+NS,J)=FTIL1(I,J)
727 CONTINUE
PAUSE 21
C CALL USWFH(2MM1,2,M,10,IROW,IROW,4)
C CALL USWFH(5MINV1,5,MINV,10,IROW,IROW,4)
C***** XFORM AHAT BY NEW M *****
CALL VMULFF(MINV,AHAT,IROW,IROW,IROW,10,10,FTIL1,10,IER)
CALL VMULFF(FTIL1,M,IROW,IROW,IROW,10,10,AHAT,10,IER)
C***** CREATE NEW B-MATRIX FOR TOTAL SYSTEM *****
READ(32,REC=4)((B(I,I),I,J=1,NI),I=1,NS)
DO 728 I=1,NOBS
DO 728 J=1,NS
L(I,J)=-L(I,J)
728 CONTINUE
CALL VMULFF(L,BLAM,NOBS,NS,NI,10,10,FTIL1,10,IER)
DO 729 I=1,NOBS
DO 729 J=1,NI
B(I+NS,J)=FTIL1(I,J)
729 CONTINUE
C***** CONSTRUCT NEW C-MATRIX FOR TOTAL SYSTEM *****
READ(32,REC=5)((C(I,I),I,J=1,NS),I=1,NO)
DO 730 I=1,NO
DO 730 J=1+NS,IROW
C(I,J)=FLOAT(0)
730 CONTINUE
PAUSE 22
CALL USWFH(5AHAT1,5,AHAT,10,IROW,IROW,4)
CALL EIGRF(AHAT,IROW,10,2,WEIO,Z,10,WKAREA,IER)
CALL USWCV(10EIGENVALUE,10,WEIO,IROW,1,4)
CALL USWCV(10EIGENVECT,10,Z,10,IROW,IROW,4)
C***** STORE NEW SYSTEM AHAT,B,C,NS,NI,NO *****
WRITE(20,REC=1)IROW,NI,NO,IDBT,ZERO
WRITE(20,REC=3)((B(I,I),I,J=1,NI),I=1,IROW)
WRITE(20,REC=4)((C(I,I),I,J=1,IROW),I=1,NO)
IUT=IROW+20+1
OPEN(UNIT=IUT,FILE='CURRNT',ACCESS='DIRECT',RECL=102)
WRITE(IUT,REC=5)((AHAT(I,I),I,J=1,IROW),I=1,IROW)
C***** ENTER INITIAL CONDITIONS AND XFORM TO FIND W(O) *****
WRITE(6,731)
731 FORMAT(1X,45H ENTER INITIAL CONDITIONS FOR ORIGINAL STATES)
READ(5,*) (VO(I,1),I=1,NS)
DO 852 I=1,NOBS
DO 852 J=1,NS
FTIL1(I,J)=-MINV(I+NS,J)

```

```

852 CONTINUE
CALL UMULFF(FTIL1,VO,NOBS,NS,1,10,10,V,10,IER)
DO 732 I=1,NOBS
  VO(I+NS,1)=VO(I+NO,1)-V(I,1)
732 CONTINUE
WRITE(6,733)
733 FORMAT(1X,51H USE THE FOLLOWING INITIAL CONDITIONS IN TIME RECF
CALL USWFM(SHX(0),5,VO,10,IROM,1,6)
C***** DO TIME RESPONSE *****
WRITE(6,463)
463 FORMAT(1X,25HWISH TO DO TIME RESPONSE?)
READ(5,*)KK
IF(KK.LE.0)GO TO 464
CALL NODE4
464 WRITE(6,734)
734 FORMAT(1X,40HWISH TO REASSIGN OBSERVER AND TRY AGAIN?)
READ(5,*)KK
IF(KK.GT.0)GO TO 699
C***** RESTORE INITIAL SYSTEM INFO *****
READ(32,REC=2)NS,NI,NO
READ(32,REC=3)((A(I,J),J=1,NS),I=1,NS)
READ(32,REC=4)((B(I,J),J=1,NI),I=1,NS)
READ(32,REC=5)((C(I,J),J=1,NS),I=1,NO)
WRITE(20,REC=1)NS,NI,NO,IDGT,ZERO
WRITE(20,REC=2)((A(I,J),J=1,NS),I=1,NS)
WRITE(20,REC=3)((B(I,J),J=1,NI),I=1,NS)
WRITE(20,REC=4)((C(I,J),J=1,NS),I=1,NO)
RETURN
END
C*****
C*****
SUBROUTINE TRANS1(A,IM,IN)
REAL A(10,10),AT(10,10)
DO 10 I=1,IM
  DO 10 J=1,IN
    AT(J,1)=A(I,J)
10 CONTINUE
DO 20 I=1,IN
  DO 20 J=1,IM
    A(I,J)=AT(I,J)
20 CONTINUE
RETURN
END
C*****
C*****
SUBROUTINE VACT
INTEGER ORIGO
REAL VO(10,10),LRORG(10),LJORG(10),VFS(10,10),VDES(10,10)
1,ALPHAY(10,10),REL(10,10),ALPHAR(10,10),EIGDIF(10,10),
2,V12(10,10),BLAK2(10,10),R(10,10),VOR(10,1),IML(10,10)
3,MI(10,10),VPI(10,1),VFSTP(10,1),VFGTP(10,1),ALAM22(10,10)
REAL W(10,10),V(10,10),VINV(10,10),F(10,10),ANAT(10,10)
REAL XX(10,10),VA(20),E(20),X(20),LRE(10),L16(10),MJ(10)
REAL BIG(20,20),BIGINV(20,20),WKBIG(450)
REAL A(10,10),B(10,10),C(10,10),WAREA(130)
COMMON/BYS/A,B,C,ZERO,IDGT,RS,NI,NO
COMMON/AUG/F,ANAT/EIG/LRE,LIM/FAR/AL/GR/G
COMMON/VEC/VA,E,X,WJ,W,XX,V,VINV

```

```

COMMON/RO/ORIGO,VO,VFS,VDES,BLAK2,LKORG,LJORG,ALAM22
I2=ORIGO-NS
IQ=0
I IQ=IQ+1
IF(ABS(LIM(IQ)).GT.ZERO)ICMPLX=1
C***** CREATE LAMBDA-0 I *****
DO 10 I=1,I2
  DO 10 J=1,I2
    REL(1,J)=FLOAT(0)
    IF(I.FQ,J)REL(I,J)=LRE(I)
10 CONTINUE
C CALL USWFM(3HREL,3,REL,10,I2,I2,4)
C***** CREATE LAMBDA-2 *****
DO 20 I=1,I2
  DO 20 J=1,I2
    ALPHAR(I,J)=ALAM22(I,J)
20 CONTINUE
C CALL USWFM(6HALPHAR,6,ALPHAR,10,I2,I2,4)
C***** REL-ALPHAR *****
DO 40 I=1,I2
  DO 40 J=1,I2
    EIGDIF(I,J)=REL(I,J)-ALPHAR(I,J)
40 CONTINUE
C CALL USWFM(6HEIGDIF,6,EIGDIF,10,I2,I2,4)
C PAUSE'40'
IF(ICMPLX.NE.1)GO TO 34
C***** CREATE LIM*I *****
DO 71 I=1,I2
  DO 71 J=1,I2
    IML(I,J)=FLOAT(0)
    IF(1.FQ,J)IML(I,J)=LIM(I)
71 CONTINUE
C***** CREATE BIG *****
DO 72 I=1,I2
  DO 72 J=1,I2
    BIG(I,J)=EIGDIF(I,J)
    BIG(I+I2,J)=-IML(I,J)
    BIG(J,J+I2)=IML(I,J)
    BIG(I+I2,J+I2)=EIGDIF(I,J)
72 CONTINUE
I22=2*I2
CALL LINV2F(BIG,I22,20,BIGINV,IDGT,WKBIG,IER)
DO 73 I=1,I2
  DO 73 J=1,I2
    ALPHAR(I,J)=BIGINV(I,J)
    IML(I,J)=BIGINV(I,J+I2)
73 CONTINUE
GO TO 35
C***** TAKE INVERSE OF EIGDIF *****
34 CONTINUE
CALL LINV2F(EIGDIF,I2,10,ALPHAR,IDGT,WKAREA,IER)
C***** PREMULT BY V12 *****
35 CONTINUE
DO 41 I=1,NS
  DO 41 J=1,I2
    V12(1,J)=VO(I,J#NS)
41 CONTINUE
C PAUSE'41'

```

```

C CALL USWFH(7)INVERSE,7,ALPHAR,10,I2,I2,4)
CALL VMULFF(VI2,ALPHAR,NS,I2,I2,10,10,REL,10,IER)
C PAUSE'REL'
C***** POSTMULT BY BLAM2 *****
CALL VMULFF(REL,BLAM2,NS,I2,NI,10,10,ALPHAR,10,IER)
C PAUSE'BLAM2'
C***** POSTMULT BY F *****
CALL VMULFF(ALPHAR,F,NS,NI,NS,10,10,REL,10,IER)
C PAUSE'F'
C*****AD RESULT TO V11 *****
DO 50 I=1,NS
DO 50 J=1,NS
R(1,J)=REL(I,J)+V0(I,J)

50 CONTINUE
C***** POST MULTIPLY BY ASSIGNED EIGENVECTOR *****
DO 60 I=1,NS
VQR(I,1)=V(I,IR)
60 CONTINUE
C CALL USWFH(3)VQR,3,VQR,10,NS,1,4)
C CALL USWFH(1)R,1,R,10,NS,NS,4)
CALL VMULFF(R,VQR,NS,NS,1,10,10,VFSTMP,10,IER)
DO 70 I=1,NS
VFS(I,IQ)=VFSTMP(I,1)
70 CONTINUE
IF(ICHPLX.NE.1) GO TO 111
C***** PREMULT BY VI2 *****
CALL VMULFF(VI2,IM,NS,I2,I2,10,10,REL,10,IER)
C***** POSTMULT BY BLAM2 *****
CALL VMULFF(REL,BLAM2,NS,I2,NI,10,10,ALPHAI,10,IER)
C***** POSTMULT BY F *****
CALL VMULFF(ALPHAI,F,NS,NI,NS,10,10,MI,10,IER)
C***** COMPUTE EIGENVECTORS *****
DO 80 I=1,NS
VQI(I,1)=V(I,IQ+1)
80 CONTINUE
CALL VMULFF(MI,VQI,NS,NS,1,10,10,VFSTMP,10,IER)
DO 90 I=1,NS
VFS(I,IQ)=VFS(I,IQ)-VFSTMP(I,1)
90 CONTINUE
CALL VMULFF(R,VQI,NS,NS,1,10,10,VFSTMP,10,IER)
CALL VMULFF(MI,VQR,NS,NS,1,10,10,VFSTP,10,IER)
DO 110 I=1,NS
VFS(I,IQ+1)=VFSTMP(I,1)+VFSTP(I,1)
110 CONTINUE
IQ=IQ+1
111 CONTINUE
ICHPLX=0
IF(IQ.LE.NS) GO TO 1
RETURN
END
C*****
C*****
SUBROUTINE ROCOST(CJ)
INTEGER ORIGO
REAL XX(10,10),VA(20),E(20),X(20),LKE(10),LIH(10),WJ(10)
REAL W(10,10),V(10,10),VINV(10,10),AL(10,10),ALAM22(10,10)
REAL VFS(10,10),VDES(10,10),BLAM2(10,10),V0(10,10)

```

```

REAL A(10,10),B(10,10),C(10,10),CRORG(10),LIORG(10)
COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
COMMON/VEC/VA,E,X,WJ,W,XX,V,VINV
COMMON/RO/ORIGO,V0,VFS,VDES,BLAM2,LRORG,LIORG,ALAM22
COMMON/EIG/LRE,LIH/PAR/AL

C***** CALCULATE ACTUAL V IN FULL SYSTEM *****
CALL VACT
C CALL USWFH(3)VFS,3,VFS,10,NS,NS,4)
C CALL USWFH(4)VDES,4,VDES,10,NS,NS,4)
C***** CALCULATE COST FUNCTION *****
CJ=0
J=0
5 J=J+1
I=0
10 I=I+1
CJTEMP=(VFS(I,J)-VDES(I,J))*2
CJ=CJTEMP*AL(I,J)+CJ
IF(ABS(LIH(J)).LE.ZERO) GO TO 20
CJ=CJ+CJTEMP*AL(I,J)
CJ=CJ+(VFS(I,J+1)-VDES(I,J+1))*2*AL(I,J+1)
IF(Y.NE.NS) GO TO 10
J=J+1
20 IF(I.NE.NS) GO TO 10
IF(J.NE.NS) GO TO 5
RETURN
END
C*****
C*****
SUBROUTINE ROGRAD
INTEGER ORIGO
REAL AUX1(10,10),AUX2(10,10),AUX3(10,10),PUCRE(10,1),FVQI(10,1)
1VAUX1(10,1),VAUX2(10,1),BETAR(10,1),ZETAR(10,1),
2V11(10,1),BETAI(10,1),ZETAI(10,1),RR(10,1),RR(10,1)
3,RR(10,1),OR(10,1),O(10,10),AL(10,10)
REAL V0(10,10),LRORG(10),LIORG(10),VFS(10,10),VDES(10,10)
1,ALPHAI(10,10),REL(10,10),ALPHAR(10,10),EIGDIF(10,10),
2VI2(10,10),BLAM2(10,10),R(10,10),VQR(10,1),IHL(10,10)
3,MI(10,10),VPI(10,1),VFSTMP(10,1),VFSTP(10,1)
REAL HL(10,10),HL(10,10),HLC(10,20),PLC(10,20),HLC(10,20)
REAL STAR(20,20),RL(10,20),RL(10,20)
REAL U(10,10),V(10,10),VINV(10,10),F(10,10),AHAT(10,10)
REAL XX(10,10),VA(20),E(20),X(20),LKE(10),LIH(10),WJ(10)
REAL A(10,10),B(10,10),C(10,10),WKAREA(130)
REAL DIB(20,20),BIGINV(20,20),WKDIB(460),ALAM22(10,10)
COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
COMMON/NSPA/HL,HL,HLC,PLC,HLC,STAR,RL,
COMMON/AUX/AUX1,AUX2,AUX3
COMMON/AUG/F,AHAT/EIG/LRE,LIH/PAR/AL/OR/G
COMMON/VEC/VA,E,X,WJ,W,XX,V,VINV
COMMON/RO/ORIGO,V0,VFS,VDES,BLAM2,LRORG,LIORG,ALAM22
I2=ORIGO-NS
IFLAG=0
DO 100 I=1,NS
DO 100 J=1,NI
G(1,J)=FLOAT(0)
100 CONTINUE
II=0
110 II=II+1

```

```

IU=0
120 IU=IU+1
IU=IU+20
IF (ABS(LIM(IQ)),GT,ZERO)GO TO 34
READ(IU,REC=3)((ML(I,J),J=1,NI),I=1,NS)
READ(IU,REC=4)((ML(I,J),J=1,NI),I=1,NI)
GO TO 35
34 IS=NS+NI
NI2=2*NI
NS2=NS*2
IHS=NS+1
READ(IU,REC=3)((MLC(I,J),J=1,IS),I=1,NS)
READ(IU,REC=4)((FLC(I,J),J=1,IS),I=1,NS)
READ(IU,REC=5)((MLC(I,J),J=1,IS),I=1,NI)
READ(IU,REC=6)((RL(I,J),J=1,NI2),I=1,NS)
READ(IU,REC=7)((RL(I,J),J=1,NI2),I=1,NS)
35 CONTINUE
C PAUSE 'PFY'
CALL PFX(I,IQ,IFLAG)
C PAUSE 'PFX'
CALL PVP(I,IQ)
C PAUSE 'PVP'
IF (ABS(LIM(IQ)),LE,ZERO)GO TO 125
IFLAG2=1
IFLAG1=1
125 CONTINUE
C***** CALCULATE ALPHAR *****
C***** CREATE LAMBDA-R I *****
DO 10 I=1,I2
DO 10 J=1,I2
REL(I,J)=FLOAT(0)
IF(I,EQ,J) REL(I,J)=LRE(IQ)
10 CONTINUE
C***** CREATE LAMBDA-2 *****
DO 20 I=1,I2
DO 20 J=1,I2
ALPHAR(I,J)=ALAM22(I,J)
20 CONTINUE
C***** REL-ALPHAR *****
DO 40 I=1,I2
DO 40 J=1,I2
EIGDIF(I,J)=REL(I,J)-ALPHAR(I,J)
40 CONTINUE
IF(IFLAG1,NE,1) GO TO 134
C***** CREATE LIM#I *****
DO 71 I=1,I2
DO 71 J=1,I2
IML(I,J)=FLOAT(0)
IF(Y,EQ,J)IML(I,J)=LIM(IQ)
71 CONTINUE
C***** CREATE BIG *****
DO 72 I=1,I2
DO 72 J=1,I2
BIG(I,J)=EIGDIF(I,J)
BIG(I+I2,J)=-IML(I,J)
BIG(I,J+I2)=IML(I,J)
BIG(I+I2,J+I2)=EIGDIF(I,J)
72 CONTINUE
I22=2*I2
CALL LINV2F(BIG,I22,20,BIGINV,IDBT,UKDIB,IER)
DO 73 I=1,I2
DO 73 J=1,I2
ALPHAR(I,J)=BIGINV(I,J)
IML(I,J)=BIG(I,J+I2)
73 CONTINUE
GO TO 136
C***** TAKE INVERSE OF EIGDIF *****
134 CONTINUE
CALL LINV2F(EIGDIF,I2,10,ALPHAR,IDBT,UKAREA,IER)
C***** PREMULT BY V12 *****
136 CONTINUE
DO 41 I=1,NS
DO 41 J=1,I2
V12(I,J)=V0(I,J*NS)
41 CONTINUE
CALL VMULFF(V12,ALPHAR,NS,I2,I2,10,10,REL,10,IER)
C***** POSTMULT BY BLAM2 *****
CALL VMULFF(REL,BLAM2,NS,I2,NI,10,10,ALPHAR,10,IER)
C***** CALCULATE BETAR *****
DO 130 I=1,NS
PVORE(I,1)=AUX1(I,IQ)
130 CONTINUE
CALL VMULFF(F,PVORE,NI,NS,1,10,10,VAUX1,10,IER)
DO 135 I=1,NS
VOR(I,1)=V(I,IQ)
135 CONTINUE
CALL VMULFF(V,VOR,NI,NS,1,10,10,VAUX2,10,IER)
DO 140 I=1,NS
BETAR(I,1)=VAUX1(I,1)+VAUX2(I,1)
140 CONTINUE
C***** CALCULATE ZETAR *****
DO 145 I=1,NS
DO 145 J=1,NS
V11(I,J)=V0(I,J)
145 CONTINUE
CALL VMULFF(V11,PVORE,NS,NS,1,10,10,ZETAR,10,IER)
C***** CALCULATE RR *****
CALL VMULFF(ALPHAR,BETAR,NS,NI,1,10,10,VAUX1,10,IER)
DO 150 I=1,NS
RR(I,1)=ZETAR(I,1)+VAUX1(I,1)
150 CONTINUE
IF(IFLAG1,NE,1) GO TO 167
C***** CALCULATE ALPHAI *****
C***** PREMULT BY V12 *****
CALL VMULFF(V12,IML,NS,I2,I2,10,10,REL,10,IER)
C***** POSTMULT BY BLAM2 *****
CALL VMULFF(REL,BLAM2,NS,I2,NI,10,10,ALPHAI,10,IER)
C***** CALCULATE BETAI *****
DO 155 I=1,NS
PV0I(I,1)=AUX1(I,IQ+1)
155 CONTINUE
CALL VMULFF(F,PV0I,NI,NS,1,10,10,VAUX1,10,IER)
DO 160 I=1,NS
V0I(I,1)=V(J,IQ11)

```



```

160 CONTINUE
   CALL VMULFF(W,VP1,NI,NS,1,10,10,VAUX2,10,IER)
   DO 165 I=1,NS
     BETAI(I,1)=VAUX1(I,1)+VAUX2(I,1)
165 CONTINUE
C***** COMPUTE ZETA1 *****
   CALL VMULFF(V1,VP1,NS,NS,1,10,10,ZETA1,10,IER)
C***** CALCULATE QU *****
   CALL VMULFF(ALPHA1,BETAI,NS,NI,1,10,10,QR,10,IER)
C***** CALCULATE QR *****
   CALL VMULFF(ALPHA1,BETAI,NS,NI,1,10,10,QR,10,IER)
   DO 170 I=1,NS
     QR(I,1)=ZETA1(I,1)+QR(I,1)
170 CONTINUE
C***** SEE IF THIS IS SECOND TIME THROUGH FOR COMPLEX ***
   IF(IFLAG2.NE.1) GO TO 168
167 JJ=I0
   GO TO 169
168 JJ=I0+1
169 CONTINUE
C***** COMPUTE GRADIENT FOR REAL CASE *****
   IP=0
175 IP=IP+1
   G(I1,JJ)=G(I1,JJ)+(VFS(IP,I0)-VDES(IP,IR))*(RR(IP,1)
     1*2*AL(IP,I0))
   IF(IP.NE.NS) GO TO 175
   IF(IFLAG1.NE.1)GO TO 203
C***** COMPUTE GRADIENT FOR COMPLEX CASE *****
   IP=0
180 IP=IP+1
   G(I1,JJ)=G(I1,JJ)-(VFS(IP,I0)-VDES(IP,I0))*(RR(IP,1)
     1*2*AL(IP,I0))+((VFS(IP,IR+1)-VDES(IP,IR+1))*(RR(IP,1)
     2*RR(IP,1))*2*AL(IP,I0+1))
   IF(IP.NE.NS) GO TO 180
C***** CHECK IF THIS IS FIRST TIME THROUGH *****
   IF(IFLAG2.NE.1) GO TO 201
   IFLAG2=0
   DO 185 I=1,NS
     DO 185 J=1,NS
       AUX1(I,J)=AUX2(I,J)
185 CONTINUE
   DO 190 I=1,NI
     DO 190 J=1,NS

       W(I,J)=AUX3(I,J)
190 CONTINUE
   GO TO 125
201 CONTINUE
   I0=I0+1
202 CONTINUE
   IF(I0.LT.NS)GO TO 120
   IF(I1.LT.NI)GO TO 110
   CALL DBKORR(NI,NS)
   RETURN
   END

```

## APPENDIX B: DESIGN SESSION EXAMPLE

```
*****
***** SPECTRAL ASSIGNMENT PACKAGE *****
```

ENTER DESIRED MODE OF OPERATION, MODE=0,1,2,...,9:

? 1

```
***** MODE 1: DATA ENTRY *****
```

\*\*\*\*\*ENTER OR CHANGE SYSTEM PARAMETERS:

PREVIOUS VALUES?

? 1

NS= 7                    NX= 2                    NO= 4                    IBGT= 8                    ZERO= .000000100000

WISH TO CHANGE?

? 0

MATRIX A :

	1 6	2 7	3	4	5
1	-.154000E+00 -.320000E-01	.154000E+01 .000000E+00	-.420000E-02	.000000E+00	-.744000E+00
2	-.996000E+00 .000000E+00	-.117000E+00 .000000E+00	-.295000E-03	.386000E-01	.200000E-01
3	.249000E+00 -.112000E+01	-.520000E+01 .000000E+00	-.100000E+01	.000000E+00	.337000E+00
4	.000000E+00 .000000E+00	.000000E+00 .000000E+00	.100000E+01	.000000E+00	.000000E+00
5	.000000E+00 .000000E+00	.000000E+00 .000000E+00	.000000E+00	.000000E+00	-.200000E+02
6	.000000E+00 -.250000E+02	.000000E+00 .000000E+00	.000000E+00	.000000E+00	.000000E+00
7	.500000E+00 .000000E+00	.000000E+00 -.500000E+00	.000000E+00	.000000E+00	.000000E+00

WISH TO CHANGE?

MATRIX B :

	1	2
1	.000000E+00	.000000E+00
2	.000000E+00	.000000E+00
3	.000000E+00	.000000E+00
4	.000000E+00	.000000E+00
5	.200000E+02	.000000E+00
6	.000000E+00	.250000E+02
7	.000000E+00	.000000E+00

WISH TO CHANGE?

? 0

MATRIX C :

	1 6	2 7	3	4	5
1	.100000E+01 .000000E+00	.000000E+00 -.100000E+01	.000000E+00	.000000E+00	.000000E+00
2	.000000E+00 .000000E+00	.000000E+00 .000000E+00	.100000E+01	.000000E+00	.000000E+00
3	.000000E+00 .000000E+00	.100000E+01 .000000E+00	.000000E+00	.000000E+00	.000000E+00
4	.000000E+00 .000000E+00	.000000E+00 .000000E+00	.000000E+00	.100000E+01	.000000E+00

WISH TO CHANGE?

? 0

WISH TO EXIT FROM THIS MODE?

? 1

\*\*\*\*\* EXITING MODE 1 \*\*\*\*\*  
 TERMINATE THIS RUN OR SELECT NEXT MODE:

WISH TO TERMINATE?

\*\*\*\*\*  
 \*\*\*\*\* SPECTRAL ASSIGNMENT PACKAGE \*\*\*\*\*

ENTER DESIRED MODE OF OPERATION, MODE=0,1,2,...,9:

? 9

WANT TO ENTER NEW ORIGINAL EIGENVECTORS?

? 0

V MATRIX:

	1 6	2 7	3	4	5
1	.383021E+00 .129240E-02	-.119002E+00 .000000E+00	-.851902E-01	.274481E+00	.374176E-01
2	.111648E+00 .551802E-04	.309671E+00 .000000E+00	.625909E-01	.256358E-01	.866465E-03
3	-.103327E+01 .466652E-01	-.359952E+00 .000000E+00	.405693E+01	-.655612E-01	-.179901E-01
4	-.225909E+00 -.186661E-02	.829627E+00 .000000E+00	-.373754E+01	.715357E+01	.899502E-03

5	.000000E+00 .000000E+00	.000000E+00 .000000E+00	.000000E+00	.000000E+00	.100000E+01
6	.000000E+00 .100000E+01	.000000E+00 .000000E+00	.000000E+00	.000000E+00	.000000E+00
7	.186997E-02 -.263755E-04	-.150251E+00 .100000E+01	.727556E-01	.279606E+00	-.959426E-03

WISH TO CHANGE ANY VALUES OF V?

? 0

ALAM:

	1 6	2 7	3	4	5
1	-.881901E-01 .000000E+00	.126948E+01 .000000E+00	.000000E+00	.000000E+00	.000000E+00
2	-.126948E+01 .000000E+00	-.881901E-01 .000000E+00	.000000E+00	.000000E+00	.000000E+00
3	.000000E+00 .000000E+00	.000000E+00 .000000E+00	-.108545E+01	.000000E+00	.000000E+00
4	.000000E+00 .000000E+00	.000000E+00 .000000E+00	.000000E+00	-.916482E-02	.000000E+00
5	.000000E+00 .000000E+00	.000000E+00 .000000E+00	.000000E+00	.000000E+00	-.200000E+02
6	.000000E+00 -.250000E+02	.000000E+00 .000000E+00	.000000E+00	.000000E+00	.000000E+00
7	.000000E+00 .000000E+00	.000000E+00 -.500000E+00	.000000E+00	.000000E+00	.000000E+00

WISH TO DISPLAY RO MODEL?

? 1

REDUCED ORDER MODEL

MATRIX A:

	1	2	3	4
1	-.881901E-01	.126948E+01	.000000E+00	.000000E+00
2	-.126948E+01	-.881901E-01	.000000E+00	.000000E+00
3	.000000E+00	.000000E+00	-.108545E+01	.000000E+00
4	.000000E+00	.000000E+00	.000000E+00	-.916482E-02

MATRIX B:

	1	2
1	-.163235E+01	-.164440E-01
2	.610923E+00	.722909E-01
3	-.277215E+00	-.287807E+00
4	-.269753E+00	-.152751E+00

MATRIX C:

	1	2	3	4
1	.381151E+00	.312491E-01	-.157946E+00	-.512508E-02

```

2  -.103327E+01  -.359952E+00  .405693E+01  -.655612E-01
3  .111648E+00  .309671E+00  .625909E-01  .256358E-01

4  -.225909E+00  .829627E+00  -.373754E+01  .715357E+01
***** REDUCED ORDER EIGENVALUE ASSIGNMENT*****
***** MODE 2: EIGENVALUE ASSIGNMENT *****

***** ENTER OR CHANGE EIGENVALUES:

PREVIOUS VALUES?
? 0
LAMBDA 1:
REAL= .000000E+00  IMAG= .000000E+00

WISH TO CHANGE?
? 1
ENTER NEW VALUE(S) :
? -1.5 1.5
LAMBDA 2: REAL= -.150000E+01  ,IMAG= -.150000E+01
NEXT EIGENVALUE:
PREVIOUS VALUES?
? 0
LAMBDA 3:
REAL= .000000E+00  IMAG= .000000E+00

WISH TO CHANGE?
? 1
ENTER NEW VALUE(S) :
? -2 1
LAMBDA 4: REAL= -.200000E+01  ,IMAG= -.100000E+01
WISH TO EXIT FROM THIS MODE?
? 1
***** EXITING MODE 2 *****
ENTER DESIRED PARTIAL EIGENVECTOR ASSIGNMENT
EIGENVECTOR V 1
? 20 0 6 7 0 0 0
EIGENVECTOR V 3
? 0 0 0 0 20 0 -8 -4
USE THE FOLLOWING V MATRIX FOR INITIAL ASSIGNMENT
REMEMBER WHICH V ARE COMPLEX
INITIAL GUESS FOR V:

      1          2          3          4
1  .486903E+02  .695530E+01  -.305837E+00  .343961E+00
2  -.136521E+01  .194087E+02  .-980851E+00  -.969533E-01
3  .124121E+02  .349015E+01  .478895E+01  .709221E-01
4  .818096E+01  -.207752E+00  .148787E+01  -.500000E+00
***** MODE 3: EIGENVECTOR ASSIGNMENT *****

***** ENTER OR CHANGE EIGENVECTORS:

PREVIOUS VALUES?
? 0
EIGENVECTOR V 1: (REAL)          (IMAG)
                  .000000E+00    .000000E+00
                  .000000E+00    .000000E+00
                  .000000E+00    .000000E+00
                  .000000E+00    .000000E+00

WISH TO CHANGE?

```

```

P 1
ENTER A NEW DESIRED VECTOR :
? 48.7 6.96 -1.37 19.4 12.4 3.49 8.18 -.208
COMPLEX VD:
      .487000E+02   -.137000E+01   .124000E+02   .818000E+01   .696000E+01
      .194000E+02   .349000E+01   -.208000E+00
ACTUAL VECTOR :
      .482275E+02   .121094E+01   .127816E+02   .800130E+01   .786399E+01
      .198133E+02   .373668E+01   -.837207E-02
ERROR VECTOR :
      .472546E+00   -.258094E+01   -.381573E+00   .178704E+00   -.903993E+00
      -.413257E+00   -.246684E+00   -.199628E+00
LENGTH OF THE DESIRED VECTOR = 55.056918
LENGTH OF THE PROJECTED VECTOR= 54.982847
LENGTH OF THE ERROR VECTOR   = 2.854955
IS THE ERROR ACCEPTABLE?
P 1
EIGENVECTOR V 2: (REAL) (IMAG)
      .487000E+02   -.696000E+01
      -.137000E+01   -.194000E+02
      .124000E+02   -.349000E+01
      .818000E+01   .208000E+00
NEXT EIGENVECTOR:
EIGENVECTOR V 3: (REAL) (IMAG)
      .000000E+00   .000000E+00
      .000000E+00   .000000E+00
      .000000E+00   .000000E+00
      .000000E+00   .000000E+00
WISH TO CHANGE?
P 1
ENTER A NEW DESIRED VECTOR :
? -.306 .344 -.981 -.097 4.79 .0709 1.49 -.5
COMPLEX VD:
      -.306000E+00   -.981000E+00   .479000E+01   .149000E+01   .344000E+00
      -.970000E-01   .709000E-01   -.500000E+00
ACTUAL VECTOR :
      -.414202E+00   -.700604E+00   .483360E+01   .139953E+01   .350915E+00
      .203737E+00   .127419E+00   -.493077E+00
ERROR VECTOR :
      .108202E+00   -.280396E+00   -.436019E-01   .904665E-01   -.691532E-02
      -.300737E+00   -.565194E-01   -.692320E-02
LENGTH OF THE DESIRED VECTOR = 5.157807
LENGTH OF THE PROJECTED VECTOR= 5.138951
LENGTH OF THE ERROR VECTOR   = .440622
IS THE ERROR ACCEPTABLE?
P 1
EIGENVECTOR V 4: (REAL) (IMAG)
      -.306000E+00   -.344000E+00
      -.981000E+00   .970000E-01
      .479000E+01   -.709000E-01
      .149000E+01   .500000E+00
=====CONTENTS OF 'CURRNT' DATA FILE INCLUDE:
MATRIX V :
      1 2 3
      4
1      .48227454187729E+02   .78639933058245E+01   -.41420195359601E+00
      .35091531544102E+00
2      .12109388238512E+01   .19813257401253E+02   -.70060416873149E+00

```

```

      .20373666886623E+00
3      .12781573152200E+02      .37366844596744E+01      .48336018720634E+01
      .12741939855255E+00
4      .80012957026792E+01      -.83720696296457E-02      .13995335459721E+01
      -.49307679752403E+00
WISH TO DISPLAY THE NORMALIZED EIGENVECTORS?
? 0
GAIN MATRIX F:
      1      2      3
      4
1      .13190086085262E+01      -.16514321136434E+01      .16927799649577E+00
      -.17238722488865E+01
2      -.38630638534669E+01      .60634704908539E+00      -.58085387760715E+01
      .31189348648067E+02
MATRIX AHAT:
      1      2      3
      4
1      -.21777513887797E+01      .39552250139616E+01      -.18080540749706E+00
      .23010868232135E+01
2      -.74293020740597E+00      -.10532544208819E+01      -.31648892353175E+00
      .12015542931493E+01
3      .74616871586856E+00      .28329043506728E+00      .53935776495116E+00
      -.84986350485817E+01
4      .23428120638957E+00      .35285780556458E+00      .84159727785996E+00
      -.43083519552895E+01
WISH TO EXIT FROM THIS MODE?
? 1

```

```

***** EXITING MODE 3 *****
GAIN MATRIX F:
      1      2      3      4
1      .131901E+01      -.165143E+01      .169278E+00      -.172387E+01
2      -.386306E+01      .606347E+00      -.580854E+01      .311893E+02

```

```

MATRIX VFS
      1      2      3      4
1      .194354E+02      .333632E+00      -.102906E+00      -.360320E-01
2      .676460E+01      .724726E+01      .752159E-01      .976052E-01
3      .106152E+01      -.974030E-01      .201980E+02      .113337E+00
4      -.424225E+00      .635155E+00      -.854180E+01      -.391375E+01
WANT TO CONTINUE SEARCH?
? 1

```

```

***** RO EIGENVECTOR IMPROVEMENT*****
WEIGHTING CONSTANTS
WEIGHTS:
      1      2      3      4
1      .100000E+01      .100000E+01      .100000E+01      .100000E+01
2      .100000E+01      .100000E+01      .100000E+01      .100000E+01
3      .100000E+01      .100000E+01      .100000E+01      .100000E+01
4      .100000E+01      .100000E+01      .100000E+01      .100000E+01
WISH TO CHANGE?
? 1

```

```

ENTER NEW VALUES:
? 1 1 1E5 1E5 1 1 1E5 1E5 1E7 1E7 1 1 1E7 1E7 1 1
COST= .343993E+08
GRADIENT SEARCH ROUTINE,SET SEARCH PARAMETERS:

DEFAULT VALUES ARE:
# OF STEPS,N= 1 STEP SIZE,H= .100000E-01 DMIN= .100000E-06

WISH TO CHANGE?
? 0
NEW COST= .305252E+08
COST FUNCTION= .305252E+08
WISH TO CONTINUE THE SEARCH?
? 1
GRADIENT SEARCH ROUTINE,SET SEARCH PARAMETERS:

DEFAULT VALUES ARE:
# OF STEPS,N= 1 STEP SIZE,H= .100000E-01 DMIN= .100000E-06

WISH TO CHANGE?
? 1

```

```

ENTER NEW VALUES:
? 10 .1 .1E-5
NEW COST= .128693E+09
LAST STEP NOT ACCEPTEDJ
STEP SIZE REDUCED TO: .500000E-01
NEW COST= .484935E+08
LAST STEP NOT ACCEPTEDJ
STEP SIZE REDUCED TO: .250000E-01
NEW COST= .317305E+08
LAST STEP NOT ACCEPTEDJ
STEP SIZE REDUCED TO: .125000E-01
NEW COST= .291831E+08
NEW COST= .317305E+08
2 STEPS WITH PRESENT GRADIENT AND DMIN= .125000E-01WERE TAKEN
LAST STEP NOT ACCEPTEDJ
NEW COST= .292885E+08
LAST STEP NOT ACCEPTEDJ
STEP SIZE REDUCED TO: .625000E-02
NEW COST= .287571E+08
NEW COST= .292885E+08
2 STEPS WITH PRESENT GRADIENT AND DMIN= .625000E-02WERE TAKEN
LAST STEP NOT ACCEPTEDJ
NEW COST= .282891E+08
NEW COST= .285721E+08
2 STEPS WITH PRESENT GRADIENT AND DMIN= .625000E-02WERE TAKEN
LAST STEP NOT ACCEPTEDJ
NEW COST= .277427E+08
NEW COST= .279597E+08
2 STEPS WITH PRESENT GRADIENT AND DMIN= .625000E-02WERE TAKEN
LAST STEP NOT ACCEPTEDJ
NEW COST= .272717E+08
NEW COST= .275363E+08
2 STEPS WITH PRESENT GRADIENT AND DMIN= .625000E-02WERE TAKEN
LAST STEP NOT ACCEPTEDJ
NEW COST= .267668E+08
NEW COST= .270331E+08
2 STEPS WITH PRESENT GRADIENT AND DMIN= .625000E-02WERE TAKEN
LAST STEP NOT ACCEPTEDJ
NEW COST= .263241E+08
NEW COST= .266436E+08
2 STEPS WITH PRESENT GRADIENT AND DMIN= .625000E-02WERE TAKEN
LAST STEP NOT ACCEPTEDJ
NEW COST= .258612E+08
NEW COST= .261890E+08
2 STEPS WITH PRESENT GRADIENT AND DMIN= .625000E-02WERE TAKEN
LAST STEP NOT ACCEPTEDJ
NEW COST= .254479E+08
NEW COST= .258223E+08
2 STEPS WITH PRESENT GRADIENT AND DMIN= .625000E-02WERE TAKEN
LAST STEP NOT ACCEPTEDJ

```



NEW COST= .250227E+08  
 COST FUNCTION= .250227E+08  
 WISH TO CONTINUE THE SEARCH?  
 "THE SEARCH IS CONTINUED AND A FINAL COST FUNCTION IS CALCULATED."  
 GRADIENT SEARCH ROUTINE, SET SEARCH PARAMETERS:

DEFAULT VALUES ARE:  
 # OF STEPS, N= 10 STEP SIZE, D= .625000E-02 DRIN= .100000E-05

COST FUNCTION= .155136E+07

MATRIX U :

	1	2	3
	4		
1	.48266531190235E+02 .35093058134550E+00	.78674405727293E+01	-.41419860399976E+00
2	.12225367214937E+01 .20373861449714E+00	.19802803020286E+02	-.70060734651333E+00
3	.12641703663988E+02 .12743165758104E+00	.38436583280420E+01	.48336001377353E+01
4	.79780115010709E+01 -.49307180141918E+00	.54529536274572E-01	.13995345304654E+01

MATRIX VFS

	1	2	3	4
1	.194545E+02	.344340E+00	-.102903E+00	-.360261E-01
2	.676321E+01	.725272E+01	.752152E-01	.976084E-01
3	.451053E+00	.332638E+00	.201900E+02	.113366E+00
4	-.672359E-01	.675862E+00	-.854179E+01	-.391376E+01

WANT TO CONTINUE SEARCH?

? 0

GAIN MATRIX

	1	2	3
	4		
1	.13188024894946E+01 -.17236128571337E+01	-.16504481491137E+01	.16932844087411E+00
2	-.38537988320367E+01 .31186019735019E+02	.58332789768427E+00	-.58101172833928E+01

MATRIX AHAT:

	1	2	3	4
1	-.217757E+01	.395400E+01	-.180862E+00	.230072E+01
2	-.742386E+00	-.105432E+01	-.316572E+00	.120147E+01
3	.743559E+00	.289643E+00	.539798E+00	-.849775E+01
4	.232922E+00	.356109E+00	.841825E+00	-.430791E+01

? 12

FORMED:

	1	2	3	4	5
	6	7			
1	.366241E+01 .294204E-02	-.313020E+01 .000000E+00	-.175650E+00	-.371862E+00	-.137152E+00
2	.153911E+01 -.119439E+00	-.603426E+01 .000000E+00	.269786E+01	.434680E+01	-.773670E-02

WISH TO DISPLAY TILDA SYSTEM?

? 0

NS= 7    NG= 4

-ALAM12T:

	1	2	3	4
1	.744000E+00	-.337000E+00	-.200000E-01	.000000E+00
2	.320000E-01	.112000E+01	.000000E+00	.000000E+00
3	.154000E+00	-.249000E+00	.996000E+00	.000000E+00

YOU MUST SELECT WHICH 3 OUTPUTS WILL BE USED TO FEED THE OBSERVER  
THE OUTPUTS CORRESPOND TO COLUMNS IN -ALAM12T  
SELECTED COLUMNS WILL FORM A MATRIX THAT MUST BE NONSINGULAR

ENTER OUTPUTS TO BE USED (INTEGER)

? 1 2 3

\*\*\*\*\* ASSIGN EIGENVALUES FOR OBSERVER\*\*\*\*\*  
\*\*\*\*\* MODE 2: EIGENVALUE ASSIGNMENT \*\*\*\*\*

\*\*\*\*\* ENTER OR CHANGE EIGENVALUES:

PREVIOUS VALUES?

? 0

LAMBDA 1:

REAL= -.150000E+01    IMAG= .150000E+01

WISH TO CHANGE?

? 1

ENTER NEW VALUE(S) :

? -5 0

NEXT EIGENVALUE:

PREVIOUS VALUES?

? 0

LAMBDA 2:

REAL= -.150000E+01    IMAG= -.150000E+01

WISH TO CHANGE?

? 1

ENTER NEW VALUE(S) :

? -6 0

NEXT EIGENVALUE:

PREVIOUS VALUES?

? 0

LAMBDA 3:

REAL= -.200000E+01    IMAG= .100000E+01

WISH TO CHANGE?

? 1

ENTER NEW VALUE(S) :

? -7 0

WISH TO EXIT FROM THIS MODE?

? 1

\*\*\*\*\* EXITING MODE 2 \*\*\*\*\*

\*\*\*\*\* ASSIGN EIGENVECTORS FOR OBSERVER\*\*\*\*\*  
\*\*\*\*\* MODE 3: EIGENVECTOR ASSIGNMENT \*\*\*\*\*

\*\*\*\*\* ENTER OR CHANGE EIGENVECTORS:

PREVIOUS VALUES  
 ? 0  
 EIGENVECTOR V 1: (REAL) (IMAG)  
                   .487000E+02 .696000E+01  
                   -.137000E+01 .194000E+02  
                   .124000E+02 .349000E+01

WISH TO CHANGE?  
 ? 1  
 ENTER A NEW DESIRED VECTOR:  
 ? 1 0 0 0 0  
 DESIRED VECTOR:  
                   .100000E+01 .000000E+00 .000000E+00  
 ACTUAL VECTOR:  
                   .100000E+01 .000000E+00 .000000E+00  
 ERROR VECTOR:  
                   .000000E+00 .000000E+00 .000000E+00  
 LENGTH OF THE DESIRED VECTOR = 1.000000  
 LENGTH OF THE PROJECTED VECTOR = 1.000000  
 LENGTH OF THE ERROR VECTOR = .000000  
 IS THE ERROR ACCEPTABLE?

? 1  
 NEXT EIGENVECTOR:  
 EIGENVECTOR V 2: (REAL) (IMAG)  
                   .487000E+02 -.696000E+01  
                   -.137000E+01 -.194000E+02  
                   .124000E+02 -.349000E+01

WISH TO CHANGE?  
 ? 1  
 ENTER A NEW DESIRED VECTOR:  
 ? 0 0 1 0 0 0  
 DESIRED VECTOR:  
                   .000000E+00 .100000E+01 .000000E+00  
 ACTUAL VECTOR:  
                   .000000E+00 .100000E+01 .000000E+00  
 ERROR VECTOR:  
                   .000000E+00 .000000E+00 .000000E+00  
 LENGTH OF THE DESIRED VECTOR = 1.000000  
 LENGTH OF THE PROJECTED VECTOR = 1.000000  
 LENGTH OF THE ERROR VECTOR = .000000  
 IS THE ERROR ACCEPTABLE?

? 1  
 NEXT EIGENVECTOR:  
 EIGENVECTOR V 3: (REAL) (IMAG)  
                   -.306000E+00 .344000E+00  
                   -.981000E+00 -.970000E-01  
                   .479000E+01 .709000E-01

WISH TO CHANGE?  
 ? 1  
 ENTER A NEW DESIRED VECTOR:  
 ? 0 0 0 0 1 0  
 DESIRED VECTOR:  
                   .000000E+00 .000000E+00 .100000E+01  
 ACTUAL VECTOR:  
                   .000000E+00 .000000E+00 .100000E+01  
 ERROR VECTOR:  
                   .000000E+00 .000000E+00 .000000E+00  
 LENGTH OF THE DESIRED VECTOR = 1.000000  
 LENGTH OF THE PROJECTED VECTOR = 1.000000  
 LENGTH OF THE ERROR VECTOR = .000000  
 IS THE ERROR ACCEPTABLE?

? 1  
 =====CONTENTS OF 'CURRENT' DATA FILE INCLUDE:  
 MATRIX V:  
                   1                  2                  3  
                   .10000000000000E+01 .00000000000000E+00 .00000000000000E+00

```

2      .00000000000000E+00      .10000000000000E+01      .00000000000000E+00
3      .00000000000000E+00      .00000000000000E+00      .10000000000000E+01
WISH TO DISPLAY THE NORMALIZED EIGENVECTORS?
? 0
GAIN MATRIX F:
      1              2              3
1      .19818625277969E+02      .76655611342999E+01      -.18571670273086E+00
2      -.56624643651339E+00      .16745269601877E+02      .53061915060189E-02
3      -.32058872043162E+01      .30010800563305E+01      -.69980706607602E+01
MATRIX AHAT:
      1              2              3
1      -.50000000000001E+01      -.31974423109205E-13      .17763568394003E-14
2      .35527136788005E-14      -.60000000000002E+01      -.55511151231258E-16
3      .14210854715202E-13      .28421709430404E-13      -.70000000000000E+01
WISH TO EXIT FROM THIS MODE?
? 1
*****EXITING MODE 3 *****
?
L:
      1              2              3              4
1      .198186E+02      -.566246E+00      -.320589E+01      .000000E+00
2      .766556E+01      .167453E+02      .300108E+01      .000000E+00
3      -.185717E+00      .530619E-02      -.699807E+01      .000000E+00
MATRIX E:
      1              2              3
1      -.500000E+01      .355271E-14      .142109E-13
2      -.319744E-13      -.600000E+01      .284217E-13
3      .177636E-14      -.555112E-16      -.700000E+01
MATRIX R:
      1              2              3              4
1      .286629E+00      -.292894E-01      -.283115E+02      -.371862E+00
2      .184378E+00      .710375E+00      -.171387E+02      .434600E+01
PAUSE 20
?
MATRIX G:
      1              2              3              4
1      -.891838E+02      .234728E+01      -.178108E+02      .123747E+00
2      -.421606E+02      -.836933E+02      .576151E+02      -.115842E+00
3      -.529284E+01      -.346816E-01      .484813E+02      .270126E+00
MATRIX TB:
      1              2
1      .200000E+02      .000000E+00
2      .000000E+00      .250000E+02
3      .000000E+00      .000000E+00
ENTER INITIAL CONDITIONS FOR ORIGINAL STATES
? 0 0 0 1 0 0 0
USE THE FOLLOWING INITIAL CONDITIONS IN TIME RESP.
X(0):
      1
1      .00000000000000E+00
2      .00000000000000E+00
3      .00000000000000E+00
4      .10000000000000E+01

```

5 .000000000000000E+00  
 6 .000000000000000E+00  
 7 .000000000000000E+00  
 8 .000000000000000E+00  
 9 .000000000000000E+00  
 10 .000000000000000E+00  
 WISH TO DO TIME RESPONSE?

? 1  
 \*\*\*\*\* MODE 4: TIME SIMULATION \*\*\*\*\*

\*\*\*\*\* CHOOSE SIMULATION OPTIONS:

-ENTER: 1 TO SIMULATE EA7, 2 TO SIMULATE EA871, (3 FOR EA11L);

? 2

ENTER 0 TO SIMULATE OUTPUTS, 1 TO SIMULATE STATE VARIABLES:

? 1

ENTER SIMULATION TIME, (REAL NUMBER IN SECONDS):

? 10

ENTER NUMBER OF POINTS TO BE CALCULATED, (200 MAX):

? 200

SPECIFY THE INITIAL CONDITIONS:

X 1(0):

? 0

X 2(0):

? 0

X 3(0):

? 0

X 4(0):

? 1

X 5(0):

? 0

X 6(0):

? 0

X 7(0):

? 0

X 8(0):

? 0

X 9(0):

? 0

X 10(0):

? 0

CHOOSE INPUT OPTIONS: 1 FOR NO INPUT, 2 FOR A STEP INPUT,  
 3 FOR A RAMP, AND 4 FOR A TRUNCATED RAMP:

INPUT OPTION FOR U 1:

? 1

INPUT OPTION FOR U 2:

? 1

ENTER 0 FOR 80 DISPLAY COLUMNS, 1 FOR 129 COLUMNS:

? 0

ENTER 0 FOR INDIVIDUAL AND 1 FOR MULTIPLE PLOTS:

? 0

DO YOU WISH TO SET THE MIN-MAX RANGES FOR THE AXES?

? 1

ENTER MIN X, MAX X, MIN Y, AND MAX Y VALUES:

? 0 12 -.005 .045

POSITION PAPER AT TOP OF FORM AND TYPE ANY INTEGER

YOU MAY ADD A SHORT NOTE (20 CHARACTERS.)

THE TIME RESPONSES ARE NOW PLOTTED. RESULTS ARE SHOWN IN FIGURES 5.16-5.25.

1. Report No. NASA CR-3889		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle Design of Multivariable Feedback Control Systems Via Spectral Assignment Using Reduced-Order Models and Reduced-Order Observers				5. Report Date April 1985	
				6. Performing Organization Code	
7. Author(s) Roland R. Mielke, Leonard J. Tung, and Preston Ivanhoe Carraway III				8. Performing Organization Report No.	
				10. Work Unit No.	
9. Performing Organization Name and Address Old Dominion University Department of Electrical Engineering Norfolk, VA 23508				11. Contract or Grant No. NSG-1650	
				13. Type of Report and Period Covered Contractor Report-Final October 1982-April 1984	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546				14. Sponsoring Agency Code 505-34-03-02	
15. Supplementary Notes Langley Technical Monitor: Ruben L. Jones					
16. Abstract The feasibility of using reduced-order models and reduced-order observers with eigenvalue/eigenvector assignment procedures is investigated. A review of spectral assignment synthesis procedures is presented. Then, a reduced-order model which retains essential system characteristics is formulated. A constant state feedback matrix which assigns desired closed loop eigenvalues and approximates specified closed loop eigenvectors is calculated for the reduced-order model. It is shown that the eigenvalue and eigenvector assignments made in the reduced-order system are retained when the feedback matrix is implemented about the full order system. In addition, those modes and associated eigenvectors which are not included in the reduced-order models remain unchanged in the closed loop full-order system. The full state feedback design is then implemented by using a reduced-order observer. It is shown that the eigenvalue and eigenvector assignments of the closed loop full-order system remain unchanged when a reduced-order observer is used. The design procedure is illustrated by an actual design problem.					
17. Key Words (Suggested by Author(s)) Spectral assignment Multivariable feedback control Reduced order model Reduced order observer			18. Distribution Statement Unclassified - Unlimited Subject Category 08		
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of Pages 148	22. Price A07



National Aeronautics and  
Space Administration

Washington, D.C.  
20546

Official Business

Penalty for Private Use, \$300

THIRD-CLASS BULK RATE

Postage and Fees Paid  
National Aeronautics and  
Space Administration  
NASA-451



**NASA**

POSTMASTER: If Undeliverable (Section 158  
Postal Manual) Do Not Return

---