29190636
/5620

MODEL IDENTIFICATION FOR THE SPACE SHUTTLE MAIN
ENGINE HIGH PRESSURE OXIDIZER TURBOPUMP

A Thesis Presented to
The Faculty of the College of Engineering and Technology
Ohio University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

Thesis
M
1992
BROW

by
Joseph R. Brown,
August, 1992

56/5636 1992/1993

Acknowledgements

I wish to express thanks to Dr. R. Dennis Irwin for his support, technical advice, supervision, patience, and friendship. It is my opinion that Dr. Irwin is a role-model professor, and I know that only my parents have taught me more than he. I would also like to express my appreciation to Dr. Jerrel Mitchell for his support, advice, and for giving me an appreciation for the southern accent. Both of these men have helped to make my graduate studies fruitful and enjoyable. Their contributions to my education will forever be appreciated.

To all of the professors at Ohio University that have contributed to my education in both undergraduate and graduate studies, thank you.

Thanks also to NASA for providing the financial support that has funded this research (Grant NASA-NAG8-167), and especially to Mr. Patrick Vallely for his guidance and advice.

To my mother and step-father, Doris and Ray Johnson, thank you for your support. I believe that you are the ones who are the most responsible for my success.

Finally, I wish to dedicate this work to my father, Fredrick R. Brown, who left us in November, 1991.

Table of Contents

Acknowledgements	i
Table of Contents	ii
List of Figures	v
 Chapter 1 Introduction	 1
1.1 Overview of the Problem	1
1.2 Organization	3
 Chapter 2 Random Signal Analysis in Linear Systems . . .	 5
2.1 Fundamental Relationships	5
2.1.1 Mean Value of Discrete Ergodic Data . .	5
2.1.2 Covariance and Correlation of Discrete Ergodic Data	6
2.1.3 Spectral Density Functions	8
2.2 Ideal Single Input/Single Output Relationships	10
2.3 Non-Ideal Single Input/Multiple Output Relationships	12
2.3.1 General Non-Ideal SIMO Relationships . .	13
2.3.2 SIMO Relationships Useful in System Identification	16
 Chapter 3 Modeling of Spectral Quantities	 18
3.1 The Eigensystem Realization Algorithm	18
3.1.1 Description of the Algorithm	19

3.1.2	Using the ERA M-file	24
3.1.3	Limitations of the Algorithm and M-file	25
3.2	Motivations for Modeling Spectral Data	26
3.3	Using ERA to Model Spectral Quantities	28
Chapter 4	Test of Theory on a Constructed Example . . .	31
4.1	Test System Description and Data Acquisition	
	Procedures	31
4.1.1	Description of the Test System	32
4.1.2	Obtaining a Discrete-Time Representation	33
4.1.3	Obtaining Experimental Time Domain	
	Output Data	34
4.1.4	Obtaining Spectral Estimates from Time	
	Domain Data	35
4.2	Application of Theory to Test System	37
Chapter 5	Application of Spectral Modeling Techniques to	
	the SSME High Pressure Oxidizer Turbopump	
	Pre-Flight Test Data	45
5.1	Description of the Pre-Flight SSME Test	45
5.2	Application of Modeling to Pre-Flight Test Data	48
5.2.1	Data Selection	48
5.2.2	Obtaining Averaged Spectral Estimates .	51
5.2.3	Usage of the M-file SPEC2MOD	53
5.2.4	Results of Applications to SSME Data . .	54
5.2.5	Separation of System and Excitation Modes	58

Chapter 6 Conclusion	61
Bibliography	65
Appendix Matlab M-Files	67

List of Figures

Figure 2.1	Ideal SISO System	10
Figure 2.1	Non-Ideal SIMO System	13
Figure 3.1	Discrete Correlation Function	29
Figure 4.1	Single Input/Two Output System	32
Figure 4.2	Magnitude Plot of S_{yy} (output #1)	37
Figure 4.3	TFDC Models of Individual Modes	38
Figure 4.4	2nd Order TFDC Model of the Ratio of Cross-Spectrum to Autospectrum	41
Figure 4.5	Magnitude [dB] of Approximated and Actual Transfer Functions	43
Figure 4.6	Phase [degrees] of Approximated and Actual Transfer Functions	43
Figure 5.1	Layout of the SSME Sensor Complement	46
Figure 5.2	HPOT Sensor Configuration	47
Figure 5.3	Thrust Profile for Pre-Flight Test	47
Figure 5.4	Cascade FFT Mapping from SG0-1	49
Figure 5.5	Magnitude, 200th Order ERA Based Model (SPEC2MOD) of the Averaged Autospectrum of the Signal Measured by SG0-1	54
Figure 5.6	Phase, 200th Order ERA Based Model (SPEC2MOD) of the Averaged Autospectrum of the Signal Measured by SG0-1	55

Figure 5.7	Magnitude, 200th Order ERA Based Model (SPEC2MOD) of the Ratio of Cross-Spectrum to Autospectrum for Signals Measured by SG0-1 and SG45-1	56
Figure 5.8	Phase, 200th Order ERA Based Model (SPEC2MOD) of the Ratio of Cross-Spectrum to Autospectrum for Signals Measured by SG0-1 and SG45-1 .	56
Figure 5.9	Relative Error of the Autospectrum Model .	57
Figure 5.10	Relative Error of the Model of the Ratio of Cross-Spectrum to Autospectrum	57

Chapter 1

Introduction

The goal of this work is to develop techniques that may be used to ascertain information about the poles and zeros of the space shuttle's main engine (SSME) high pressure oxidizer turbopump (HPOT) given pre-flight test data. Such information could allow NASA personnel to categorize engine components as faulty or functional, to identify maintenance and replacement needs, and to determine operational constraints.

1.1 Overview of the Problem.

Given estimates of the spectral characteristics of the output signals of a single input/multiple output (SIMO) system, and no exact information about the system input, it is possible to determine information about the excitation characteristics and the poles and zeros of the system. It will be shown that the poles of the autospectrum of any output signal are a combination of the poles of the system and the poles of the excitation. Given an accurate model of the autospectrum, it is possible to develop criteria to separate these poles, thereby allowing for identification of the system determinant. Once the pole locations have been identified, it is possible to determine the corresponding natural frequencies and damping ratios.

It will also be shown that the poles and zeros of the ratio of the cross-spectrum of any two output signals of a

system to the autospectrum of one of the output signals provides information about the system zeros. In particular, the poles of such a ratio are the zeros of an input/output (I/O) transfer function of the system. Similarly, the zeros of such a ratio are the zeros of another I/O transfer function of the system.

Two methods for modeling spectral quantities will be discussed. These are the Transfer Function Determination Code (TFDC) and a technique which utilizes the Eigensystem Realization Algorithm (ERA). TFDC is a least squares curve fitting algorithm which may be directly applied to spectral data in order to obtain a rational function model. ERA is an algorithm that may be used to obtain state space models from causal pulse response data. ERA is not directly capable of modeling either the frequency domain spectral data or the corresponding discrete correlation data (because of the non-causal characteristic). However, it will be shown that a discrete correlation function ($\Delta R_{AB}(k)$) may be represented by the sum of two pulse responses and a constant (namely $R_{AB}(k)$ for $k < 0$, $k > 0$, and $k = 0$, respectively). In equation form this is

$$R_{AB}(k) = h_{-}(-k)u(-k) + h_{+}(k)u(k) + R_{AB}(0)\delta(k)$$

where $u(k)$ is a discrete step, i.e. $u(k)=0$ for $k \leq 0$ and $u(k)=1$ for $k > 0$, and $h_{+}(k)$ and $h_{-}(k)$ are the pulse responses corresponding to the positive and negative time halves of

$R_{AB}(k)$ respectively. ERA may be used to form sub-models of the individual pulse responses and a composite model of the discrete correlation data may be obtained by combining these sub-models and the constant in a prescribed manner. The successes and shortcomings of each method will be discussed.

The ERA based spectral modeling technique will be applied to data provided to Ohio University by the NASA Marshall Space Flight Center. This data was taken from strain gauges and accelerometers mounted on the HPOT's housing weld during a pre-flight test conducted at the NASA Stennis Space Flight Center. The results of the application will be presented and discussed.

1.2 Organization.

This thesis consists of six chapters and an appendix. Chapter 1 is the introduction. In Chapter 2, the mathematical foundation is developed from the theory of random variables and spectral analysis. Chapter 3 provides an in-depth discussion of ERA for single input/single output (SISO) systems and outlines how ERA may be used to construct models of spectral quantities.

In Chapter 4, the TFDC algorithm is used to form models of spectral estimates of output signals obtained from a constructed test system. The resulting models are analyzed, and selected poles are used to form a model of one of the I/O transfer functions of the system. The frequency response of

the modeled I/O transfer function is compared with that of the true I/O transfer function.

In Chapter 5, the results of applications of the ERA based spectral modeling technique to the data provided by NASA are presented. The accuracy of these applications is evaluated, and two potential separation criteria for separating system and excitation modes are discussed.

Chapter 6 is the conclusion. Finally, several Matlab m-files which were developed for use during this research are listed in the Appendix.

Chapter 2

Random Signal Analysis in Linear Systems

The method presented in Chapter 3 for identifying the characteristics of a system from its measured output data relies heavily on the theory of random variables. This chapter outlines the necessary theory. In Section 2.1, fundamental equations of random variable theory are presented. These equations and theory are applied to an ideal single input/single output system in Section 2.2. Finally, in Section 2.3, equations for non-ideal (i.e., disturbances in measurements of output signals) single input/multiple output systems are presented and developed.

The equations in this chapter are presented for the discrete case. Many of the developments closely parallel the continuous time presentations found in references [1] and [2].

2.1 Fundamental Relationships.

In this section, a selection of useful relationships for discrete random variable analysis is presented. These relationships form the foundation of the single input/single output and single input/multiple output system theories that are developed in subsequent sections of this chapter.

2.1.1 Mean Value of Discrete Ergodic Data.

Given an ensemble of discrete data records, $x_i(k)$ ($i=1,2,\dots,n$), representing measurements of any single

phenomenon of interest, it is possible to estimate the mean value^[1,p.3] at any discrete point k_1 by averaging over the entire ensemble. In equation form this is

$$\mu_x(k_1) = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n x_i(k_1) . \quad (2.1)$$

If the mean value and all other moments of interest, such as the mean square value, are constants that are independent of their discrete origin, then the data is said to be stationary^[2,p.177]. In addition, if the data is assumed to be ergodic^[2,p.179], then each member of the ensemble will exhibit the same statistical behavior as the whole ensemble. For ergodic data the mean value may be computed by using the equation

$$\mu_x = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=0}^{N-1} x_i(k) , \quad (2.2)$$

where N is the total number of data points in $x_i(k)$. This is equivalent to solving for the expected value^[1,p.30] of any individual member of the ensemble $x_i(k)$.

2.1.2 Covariance and Correlation of Discrete Ergodic Data.

Consider two discrete data records, $x(k)$ and $y(k)$, of length N points each ($k=0,1,2,\dots,[N-1]$). Assuming identical sample rates, these data records can be represented by the discrete time-domain functions $x(kT_s+t_x)$ and $y(kT_s+t_y)$, respectively, where T_s is the sample period and t_x and t_y are

the respective time origins when $k=0$. Defining a new variable $\tau = t_y - t_x$ and assuming t_x as the reference (i.e. assuming $t_x=0$), these functions may be re-written as

$$x(kT_s + t_x) \big|_{t_x=0} = x(kT_s) \quad (2.3)$$

and

$$y(kT_s + t_y) \big|_{\substack{\tau = t_x - t_y \\ t_x=0}} = y(kT_s + \tau) \quad (2.4)$$

The covariance function^[1, p.47] is a function of the time delay, τ , and it is defined as

$$C_{xy}(\tau) = E[\{x(kT_s) - \mu_x\} \{y(kT_s + \tau) - \mu_y\}] \quad (2.5)$$

where the symbol E represents the expected value of the indicated quantity. By completing the inner multiplication and applying the rule for expectations of sums (the expected value of a sum is equal to the sum of the expected values), Equation 2.5 may be re-written into the form

$$\begin{aligned} C_{xy}(\tau) = & E[x(kT_s) y(kT_s + \tau)] - E[x(kT_s) \mu_y] \\ & - E[\mu_x y(kT_s + \tau)] + E[\mu_x \mu_y] \end{aligned} \quad (2.6)$$

Assuming that the data is stationary, this reduces to

$$\begin{aligned} C_{xy}(\tau) = & R_{xy}(\tau) - \mu_x \mu_y - \mu_x \mu_y + \mu_x \mu_y \\ = & R_{xy}(\tau) - \mu_x \mu_y \end{aligned} \quad (2.7)$$

where

$$\begin{aligned} R_{xy}(\tau) = & E[x(kT_s) y(kT_s + \tau)] \\ = & \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=0}^{N-1} x(kT_s) y(kT_s + \tau) \end{aligned} \quad (2.8)$$

is the cross-correlation function^{[1, p.47] [2, p.207]} between $x(kT_s)$ and

$y(kT_s)$. For the special case when $x(kT_s)$ and $y(kT_s)$ represent the same data (i.e. $x(kT_s)=y(kT_s)$) the covariance function is given by

$$C_{xx}(\tau) = R_{xx}(\tau) - \mu_x^2, \quad (2.9)$$

where

$$\begin{aligned} R_{xx}(\tau) &= E[x(kT_s) x(kT_s + \tau)] \\ &= \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=0}^{N-1} x(kT_s) x(kT_s + \tau) \end{aligned} \quad (2.10)$$

is the autocorrelation function^{[1,p.48] [2,p.190]} of $x(kT_s)$. An important property of the autocorrelation function which will be useful in later developments is that it is an even function of its independent variable τ , that is,

$$R_{xx}(\tau) = R_{xx}(-\tau). \quad (2.11)$$

The cross-correlation function is neither an even nor an odd function of τ , but it does possess the property

$$R_{xy}(\tau) = R_{yx}(-\tau). \quad (2.12)$$

2.1.3 Spectral Density Functions.

If the autocorrelation and cross-correlation functions are assumed to be discrete in nature, having been calculated for a discrete set of time delays, $\tau = k_\tau T_\tau$, where k_τ is the set of integers and T_τ is the period of the discrete variable τ , then the spectral density functions may be calculated via the double-sided z-transforms^[3,p.22] of the appropriate correlation functions. The autospectral density function^{[1,p.50] [2,p.236]}, or

autospectrum, may be calculated by taking the double-sided z-transform of the discrete autocorrelation function $R_{xx}(k_\tau T_\tau)$,

$$S_{xx}(z) = Z[R_{xx}(k_\tau T_\tau)] = \sum_{k_\tau=-\infty}^{\infty} R_{xx}(k_\tau T_\tau) z^{-k_\tau}, \quad (2.13)$$

where the symbol Z represents the double-sided z-transform of the indicated quantity. Similarly, the cross-spectral density, or cross-spectrum^{[1,p.50][2,p.259]}, may be calculated by taking the z-transform of the cross-correlation function $R_{xy}(k_\tau T_\tau)$,

$$S_{xy}(z) = Z[R_{xy}(k_\tau T_\tau)] = \sum_{k_\tau=-\infty}^{\infty} R_{xy}(k_\tau T_\tau) z^{-k_\tau}. \quad (2.14)$$

Certain properties of the spectral functions^[2,p.236,p.259] will be useful in later developments. These properties may be summarized as follows:

1. The autospectrum is a real, even function of frequency ω . Given that $z=e^{sT_\tau}$, where $s=\sigma+j\omega$, this translates into

$$S_{xx}(e^{j\omega T_\tau}) = S_{xx}(e^{-j\omega T_\tau}). \quad (2.15)$$

2. The cross-spectrum is a complex function whose real part is an even function of frequency and whose imaginary part is an odd function of frequency. Following the notational convention established above, this is written as

$$\text{Re}[S_{xy}(e^{j\omega T_\tau})] = \text{Re}[S_{xy}(e^{-j\omega T_\tau})] \quad (2.16)$$

and

$$\text{Im}[S_{xy}(e^{j\omega T_r})] = -\text{Im}[S_{yx}(e^{-j\omega T_r})] , \quad (2.17)$$

where the notation Re and Im represents the real and imaginary parts, respectively.

3. Assuming real coefficients, the cross-spectrum from signal x to signal y is equal to the complex conjugate of the cross-spectrum from signal y to signal x,

$$S_{xy}(e^{j\omega T_r}) = S_{yx}^*(e^{-j\omega T_r}) . \quad (2.18)$$

2.2 Ideal Single Input/Single Output Relationships.

Consider the discrete single input/single output system (SISO) shown in Figure 2.1. For this system, $h(k)$ represents

Figure 2.1 Ideal SISO System.

the discrete pulse response. The discrete output signal, $y(k)$, may be calculated using the discrete convolution formula^[3,p.34]

$$y(k) = \sum_{n=-\infty}^{\infty} h(n) x(k-n) . \quad (2.19)$$

In the time-domain, the output signal may be described by the discrete-time function $y(kT_s + t_y)$ for $k=0,1,\dots,N$. Assuming t_y as reference ($t_y=0$), this becomes $y(kT_s)$.

By using Equations 2.10 and 2.19 and constraining the period of τ to be equal to the period of the sampled system, $T_\tau = T_s$, it is possible to solve for a discrete autocorrelation function of the output signal in terms of the discrete pulse response and the input signal. This is,

$$\begin{aligned}
 R_{yy}(k_\tau T_s) &= E[y(k_\tau T_s) y([k+k_\tau] T_s)] \\
 &= \frac{1}{N} \sum_{k=0}^{N-1} \left(\sum_{n_1=-\infty}^{\infty} h(n_1) x([k-n_1] T_s) \right) \left(\sum_{n_2=-\infty}^{\infty} h(n_2) x([k+k_\tau-n_2] T_s) \right) \quad (2.20) \\
 &= \sum_{n_1=-\infty}^{\infty} h(n_1) \sum_{n_2=-\infty}^{\infty} h(n_2) \left(\frac{1}{N} \sum_{k=0}^{N-1} x([k-n_1] T_s) x([k+k_\tau-n_2] T_s) \right).
 \end{aligned}$$

Examination of the last bracketed element of Equation 2.20 reveals that it defines a discrete autocorrelation function of $x(kT_s)$. Thus, Equation 2.20 may be reduced to

$$R_{yy}(k_\tau T_s) = \sum_{n_1=-\infty}^{\infty} h(n_1) \sum_{n_2=-\infty}^{\infty} h(n_2) R_{xx}([k_\tau-n_2+n_1] T_s) \quad (2.21)$$

As stated in Equation 2.13, an equation describing the autospectrum may be found by taking the double-sided z-transform of a discrete autocorrelation function. Applying this to the autocorrelation function of Equation 2.21 yields

$$\begin{aligned}
 S_{yy}(z) &= \sum_{k_\tau=-\infty}^{\infty} \left(\sum_{n_1=-\infty}^{\infty} h(n_1) \sum_{n_2=-\infty}^{\infty} h(n_2) R_{xx}([k_\tau-n_2+n_1] T_s) \right) z^{(k_\tau-n_2+n_1)} z^{(n_2-n_1)} \\
 &= \sum_{n_1=-\infty}^{\infty} h(n_1) z^{-n_1} \sum_{n_2=-\infty}^{\infty} h(n_2) z^{n_2} \sum_{k_\tau=-\infty}^{\infty} R_{xx}([k_\tau-n_2+n_1] T_s) z^{(k_\tau-n_2+n_1)}, \quad (2.22)
 \end{aligned}$$

which may be expressed in the simple form

$$S_{yy}(z) = H(1/z) H(z) S_{xx}(z) \quad (2.23)$$

A discrete cross-correlation function between the input and output signals may be found by using Equations 2.8 and 2.19. By constraining $T_r = T_s$, this may be written as

$$\begin{aligned}
 R_{xy}(k_r T_s) &= E[x(k T_s) y([k+k_r] T_s)] \\
 &= \frac{1}{N} \sum_{k=0}^{N-1} x(k T_s) \left(\sum_{n=-\infty}^{\infty} h(n) x([k+k_r-n] T_s) \right) \\
 &= \sum_{n=-\infty}^{\infty} h(n) \left(\frac{1}{N} \sum_{k=0}^{N-1} x(k T_s) x([k+k_r-n] T_s) \right) \quad (2.24) \\
 &= \sum_{n=-\infty}^{\infty} h(n) R_{xx}([k_r-n] T_s) .
 \end{aligned}$$

Applying the double-sided z-transform, as in Equation 2.14, yields

$$\begin{aligned}
 S_{xy}(z) &= \sum_{k_r=-\infty}^{\infty} \left(\sum_{n=-\infty}^{\infty} h(n) R_{xx}([k_r-n] T_s) \right) z^{(k_r-n)} z^n \\
 &= \sum_{n=-\infty}^{\infty} h(n) z^n \sum_{k_r=-\infty}^{\infty} R_{xx}([k_r-n] T_s) z^{(k_r-n)} \quad (2.25) \\
 &= H(z) S_{xx}(z) .
 \end{aligned}$$

2.3 Non-Ideal Single Input/Multiple Output Relationships.

In this section, a selection of spectral relationships for non-ideal single input/multiple output systems (SIMO) are developed. In part one, Section 2.3.1, general equations for the non-ideal SIMO system are presented. In part two, Section 2.3.2, several equations that are particularly useful for system identification purposes are developed.

2.3.1 General Non-Ideal SIMO Relationships.

Consider the non-ideal SIMO system shown in Figure 2.2. For this system, $d_i(k)$ represents a random disturbance. This

Figure 2.2 Non-Ideal SIMO System.

disturbance is added to the *actual* system output signal, $y_i(k)$, causing the *measured* system output signal, $y'_i(k)$, to be

$$y'_i(k) = y_i(k) + d_i(k) . \quad (2.26)$$

An equation describing the autospectrum of any *actual* system output signal may be found by applying Equation 2.23,

$$S_{y_i y_i}(z) = H_i(1/z) H_i(z) S_{xx}(z) . \quad (2.27)$$

Also, an equation describing the cross-spectrum between the input signal, $x(k)$, and any *actual* output signal may be found by applying equation 2.25,

$$S_{xy_i}(z) = H_i(z) S_{xx}(z) . \quad (2.28)$$

A discrete cross-correlation function between any two *measured* output signals can be derived from Equations 2.8, 2.10, and 2.26. By assuming the time lag $\tau = k_\tau T_s$, this may be written as

$$\begin{aligned}
 R_{y_i' y_j'}(k_\tau T_s) &= E[y_i'(k T_s) y_j'([k+k_\tau] T_s)] \\
 &= E[\{y_i(k T_s) + d_i(k T_s)\} \{y_j([k+k_\tau] T_s) + d_j([k+k_\tau] T_s)\}] \\
 &= E[y_i(k T_s) y_j([k+k_\tau] T_s)] + E[y_i(k T_s) d_j([k+k_\tau] T_s)] \\
 &\quad + E[d_i(k T_s) y_j([k+k_\tau] T_s)] + E[d_i(k T_s) d_j([k+k_\tau] T_s)] \\
 &= R_{y_i y_j}(k_\tau T_s) + R_{y_i d_j}(k_\tau T_s) + R_{d_i y_j}(k_\tau T_s) + R_{d_i d_j}(k_\tau T_s) . \quad (2.29)
 \end{aligned}$$

If the disturbances are assumed to be mutually uncorrelated (i.e. $d_i(k)$ uncorrelated with $d_j(k)$ for $i \neq j$) and uncorrelated with all other signals in the system, then Equation 2.29 may be reduced to

$$\begin{aligned}
 R_{y_i' y_j'}(k_\tau T_s) &= R_{y_i y_j}(k_\tau T_s) \\
 &= E[y_i(k T_s) y_j([k+k_\tau] T_s)] \\
 &= \frac{1}{N} \sum_{k=0}^{N-1} \left(\sum_{n_i=-\infty}^{\infty} h_i(n_i) x([k-n_i] T_s) \right) \left(\sum_{n_j=-\infty}^{\infty} h_j(n_j) x([k+k_\tau-n_j] T_s) \right) \\
 &= \sum_{n_i=-\infty}^{\infty} h_i(n_i) \sum_{n_j=-\infty}^{\infty} h_j(n_j) \left(\frac{1}{N} \sum_{k=0}^{N-1} x([k-n_i] T_s) x([k+k_\tau-n_j] T_s) \right) \\
 &= \sum_{n_i=-\infty}^{\infty} h_i(n_i) \sum_{n_j=-\infty}^{\infty} h_j(n_j) R_{xx}(k_\tau - n_j + n_i) . \quad (2.30)
 \end{aligned}$$

An equation describing the cross-spectrum of any two *measured* output signals may be found by taking the double-sided z-transform of Equation 2.30,

$$\begin{aligned}
 S_{y_i' y_j'}(z) &= \sum_{k_\tau=-\infty}^{\infty} \left(\sum_{n_i=-\infty}^{\infty} h_i(n_i) \sum_{n_j=-\infty}^{\infty} h_j(n_j) R_{xx}([k_\tau - n_j + n_i] T_s) \right) z^{(k_\tau - n_j + n_i)} z^{(n_j - n_i)} \\
 &= \sum_{n_i=-\infty}^{\infty} h_i(n_i) z^{-n_i} \sum_{n_j=-\infty}^{\infty} h_j(n_j) z^{n_j} \sum_{k_\tau=-\infty}^{\infty} R_{xx}([k_\tau - n_j + n_i] T_s) z^{(k_\tau - n_j + n_i)} , \quad (2.31)
 \end{aligned}$$

which may be expressed in the simplified form

$$\begin{aligned} S_{y_i y_j'}(z) &= S_{y_i y_j}(z) \\ &= H_i(1/z) H_j(z) S_{xx}(z) . \end{aligned} \quad (2.32)$$

A discrete autocorrelation function for any *measured* output signal may be found by applying Equations 2.8, 2.10, and 2.26.

This is

$$\begin{aligned} R_{y_i y_i'}(k_\tau T_s) &= E[y_i'(k T_s) y_i'([k+k_\tau] T_s)] \\ &= E[\{y_i(k T_s) + d_i(k T_s)\} \{y_i([k+k_\tau] T_s) + d_i([k+k_\tau] T_s)\}] \\ &= E[y_i(k T_s) y_i([k+k_\tau] T_s)] + E[y_i(k T_s) d_i([k+k_\tau] T_s)] \\ &\quad + E[d_i(k T_s) y_i([k+k_\tau] T_s)] + E[d_i(k T_s) d_i([k+k_\tau] T_s)] \\ &= R_{y_i y_i}(k_\tau T_s) + R_{y_i d_i}(k_\tau T_s) + R_{d_i y_i}(k_\tau T_s) + R_{d_i d_i}(k_\tau T_s) , \end{aligned} \quad (2.33)$$

which may be reduced to

$$R_{y_i y_i'}(k_\tau T_s) = R_{y_i y_i}(k_\tau T_s) + R_{d_i d_i}(k_\tau T_s) \quad (2.34)$$

because of the assumed lack of correlation between disturbances and actual output signals.

Taking the z-transform of Equation 2.34 yields an equation describing the autospectrum of any *measured* output signal,

$$S_{y_i y_i'}(z) = S_{y_i y_i}(z) + S_{d_i d_i}(z) . \quad (2.35)$$

If it is assumed that there exists a large signal to noise ratio,

$$S_{y_i y_i}(z) / S_{d_i d_i}(z) \gg 1 , \quad (2.36)$$

then the autospectrum of the measured output signal will be approximately equal to the autospectrum of the actual output signal. In equation form this is written as

$$S_{y_i y_i'}(z) \approx S_{y_i y_i}(z) . \quad (2.37)$$

2.3.2 SIMO Relationships Useful in System Identification.

As stated in Section 2.1.3, autospectrums are real, even functions of frequency. In other words, they are symmetrical about the zero frequency axis. Because of this symmetry, it is possible to express autospectra in a factored rational form. Applying this to the input autospectrum yields

$$S_{xx}(z) = \frac{N_x(1/z)N_x(z)}{D_x(1/z)D_x(z)} . \quad (2.38)$$

The z-domain input-to-output (I/O) transfer functions, $H_i(z)$, may be expressed as a ratio of two z-domain polynomials. If the system is assumed to be linear, then the denominator polynomial, or system determinant, will be the same for each I/O transfer function. In equation form this is

$$H_i(z) = K_i \frac{N_i(z)}{D_s(z)} , \quad (2.39)$$

where the roots of $D_s(z)$ are the z-domain poles of the system and K_i represents a constant gain.

Combining Equations 2.27, 2.38, and 2.39 yields a new expression describing the autospectrum of any actual output signal

$$S_{y_i y_i}(z) = \frac{K_i^2 N_i(1/z)N_i(z)N_x(1/z)N_x(z)}{D_s(1/z)D_s(z)D_x(1/z)D_x(z)} . \quad (2.40)$$

Solving Equation 2.27 for a relationship describing the input autospectrum yields

$$S_{xx}(z) = \frac{S_{y_i y_i}(z)}{H_i(1/z)H_i(z)} \quad (2.41)$$

which may be combined with Equations 2.32 and 2.39 to yield a new equation describing the cross-spectrum between any two measured output signals

$$\begin{aligned}
 S_{y_i' y_j'}(z) &= \frac{K_i K_j N_i(1/z) N_j(z) D_s(1/z) D_s(z) S_{y_i y_i}(z)}{K_i^2 N_i(1/z) N_i(z) D_s(1/z) D_s(z)} \\
 &= \frac{K_j N_j(z) S_{y_i y_i}(z)}{K_i N_i(z)} \quad .
 \end{aligned} \tag{2.42}$$

Finally, Equation 2.42 may be manipulated into the form

$$\frac{S_{y_i' y_j'}(z)}{S_{y_i y_i}(z)} = \frac{K_j N_j(z)}{K_i N_i(z)} \quad . \tag{2.43}$$

An explanation of how the preceding equations and theory may be used in the system identification process may be found in Section 3.2.

Chapter 3

Modeling of Spectral Quantities

The Eigensystem Realization Algorithm (ERA) [4, pp.31-42] [5] [6] [7] is a modeling technique that is capable of producing a state space approximation of a multi-input/multi-output (MIMO) system from discrete pulse response data. ERA may also be used to produce state space models of spectral quantities. In Section 3.1, a description of the Eigensystem Realization Algorithm for the single input/single output case is presented. Section 3.2 explains the motivations for applying modeling techniques to spectral data. Finally, in Section 3.3, a description of how ERA may be utilized to construct models of spectral quantities is presented.

3.1 The Eigensystem Realization Algorithm.

A detailed description of the Eigensystem Realization Algorithm for the MIMO case is presented by Medina in reference [4]. For purposes of this thesis, however, it is only necessary to discuss the SISO application of the algorithm. Section 3.1.1 summarizes Medina's MIMO description for the SISO case. In Section 3.1.2 a description of how to use the m-file "ERA" (written by Medina) is presented. Finally, in section 3.1.3 the limitations of the algorithm and m-file are discussed.

3.1.1 Description of the Algorithm.

A linear, time invariant, discrete-time system may be described by the state equations

$$x(k+1) = Ax(k) + Bu(k) \quad (3.1)$$

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

where $x(k)$ is the n -dimensional state vector and A , B , and C are the matrices of the state space realization. The time domain I/O behavior of the system is described by the Markov parameters

$$Y(k) = CA^{k-1}B , \quad k=1,2,3,\dots \quad (3.3)$$

where the i,j element of the matrix $Y(k)$ is the k^{th} sample of the i^{th} output caused by a discrete pulse at the j^{th} input with all other inputs set to zero. For the SISO case, the Markov parameter matrix, $Y(k)$, is 1×1 and is equivalent to the discrete pulse response, $h(k)$. That is,

$$h(k) = Y(k) . \quad (3.4)$$

Step 1: Formation of Hankel Matrices.

The experimentally obtained discrete pulse response of a system, $h(k)$, is arranged into Hankel matrices^[8,p.70] of dimensions $r(\text{rows}) \times s(\text{columns})$ as follows:

$$H_{rs}(k-1) = \begin{bmatrix} h(k) & h(k+1) & \dots & h(k+s-1) \\ h(k+1) & h(k+2) & \dots & h(k+s) \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ h(k+r-1) & h(k+r) & \dots & h(k+r+s-2) \end{bmatrix} . \quad (3.5)$$

Ideally, the dimensions r and s should be chosen to be large

enough so that the rank of $H_{rs}(0)$ does not increase as either r or s or both are increased. If the measurements of the discrete pulse response are ideal (i.e. not corrupted by noise or nonlinearities), then the rank of $H_{rs}(0)$ as r and s become infinitely large will be an integer, n , equal to the order of a minimal realization of the system^[4, pp.33-34]. In equation form this is

$$\lim_{\substack{r \rightarrow \infty \\ s \rightarrow \infty}} \text{rank}[H_{rs}(0)] = n . \quad (3.6)$$

However, if noise or any other nonlinearities are introduced into the measurements of the discrete pulse response (i.e. non-ideal measurements), as will be the case in any actual system, then the condition described in Equation 3.6 will not hold. In this case, the rank of $H_{rs}(0)$ may become unbounded. Hence, the task of choosing the dimensions r and s is not clearly defined. The minimal order of the system must be estimated (i.e. count the modes, multiply by 2 to obtain estimate [multiply by 4 for autospectrum models]), and the values of both r and s must be chosen to be at least as large as the estimated order of the system.

Sub-step 1A: Choose r and s to be at least as large as the estimated order of the system.

Sub-step 1B: Form $H_{rs}(0)$.

Sub-step 1C: Form $H_{rs}(1)$.

Step 2: Perform Singular Value Decomposition of $H_{rs}(0)$.

The singular value decomposition of $H_{rs}(0)$ may be performed by using standard techniques so that

$$H_{rs}(0) = U_{rN} S_N V_{sN}^T, \quad (3.7)$$

where the subscripts rN and sN denote the row and column dimensions of the matrices U and V, respectively,

$$N = \min[r, s], \quad (3.8)$$

and

$$S_N = \text{diag}(s_1, s_2, \dots, s_N). \quad (3.9)$$

As was mentioned in Step 1, noise and nonlinearities may cause the rank of $H_{rs}(0)$ to be much larger than the order of a minimal realization of the system. An estimation of the order of a minimal realization may be obtained by examining the singular values of $H_{rs}(0)$. If the singular values $s_1, s_2, \dots, s_n$ are significantly larger (on a "relative" basis) than $s_{n+1}, \dots, s_N$ then an estimation of the order of the minimal realization would be "n".

Sub-step 2A: Find the singular value decomposition of $H_{rs}(0)$ as defined in Equation 3.7.

Sub-step 2B: Re-evaluate the estimated order of the minimal realization, n, by examining the singular values of $H_{rs}(0)$ (see above explanation).

Sub-step 2C: Truncate the matrix S_N , defined in Equation 3.9, to obtain $S_n = \text{diag}(s_1, s_2, \dots, s_n)$.

Sub-step 2D: Obtain the matrices U_{rn} and V_{sn} by

removing the last $N-n$ columns of U_{rn} and V_{sn} respectively.

Step 3: Formation of the State Space Approximation.

Defining a new Hankel matrix, $H_{rs}(k)$, and writing it into a form similar to that of Equation 3.5 yields

$$H_{rs}(k) = \begin{bmatrix} h(k+1) & h(k+2) & \dots & h(k+s) \\ h(k+2) & h(k+3) & \dots & h(k+s+1) \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ h(k+r) & h(k+r+1) & \dots & h(k+r+s-1) \end{bmatrix}. \quad (3.10)$$

It may easily be shown that the 1,1 element of the matrix defined above, $h(k+1)$, can be written as the product

$$h(k+1) = E_{1r} H_{rs}(k) E_{s1}, \quad (3.11)$$

where

$$E_{1r} = [1 \ 0 \ 0 \ \dots \ 0] \quad (1 \times r) \quad (3.12)$$

and

$$E_{s1}^T = [1 \ 0 \ 0 \ \dots \ 0] \quad (s \times 1)^T. \quad (3.13)$$

The matrix $H_{rs}(k)$ defined in Equation 3.10 may also be defined by the product

$$H_{rs}(k) = P A^k Q \quad (3.14)$$

where the matrices P and Q can be chosen to be any two matrices of rank n such that $H_{rs}(0) = PQ$. A convenient choice for P and Q is

$$P = U_{rn} S_n^{1/2} \quad P^T = S_n^{-1/2} U_{rn}^T \quad (3.15)$$

$$Q = S_n^{1/2} V_{sn}^T \quad Q^T = V_{sn} S_n^{-1/2} \quad (3.16)$$

where $P^{\dagger}$ and $Q^{\dagger}$ are the pseudo-inverses^[9,p.243] of P and Q , respectively.

Combining Equations 3.3, 3.4, 3.11, and 3.14 leads to the following derivation:

$$\begin{aligned}
 h(k+1) &= E_{1r} P A^k Q E_{s1} \\
 &= E_{1r} P P^T P A^k Q Q^T Q E_{s1} \\
 &= E_{1r} P [P^T P A Q Q^T]^k Q E_{s1} \\
 &= E_{1r} P [P^T H_{rs}(1) Q^T]^k Q E_{s1} \\
 &= E_{1r} U_{rn} S_n^{1/2} [S_n^{-1/2} U_{rn}^T H_{rs}(1) V_{sn} S_n^{-1/2}]^k S_n^{1/2} V_{sn}^T E_{s1} . \quad (3.17)
 \end{aligned}$$

By comparing the results of Equation 3.17 to those of Equations 3.3 and 3.4 it is easy to see that the discrete pulse response may be realized by the state space model

$$\begin{aligned}
 A &= S_n^{-1/2} U_{rn}^T H_{rs}(1) V_{sn} S_n^{-1/2} \\
 B &= S_n^{1/2} V_{sn}^T E_{s1} \\
 C &= E_{1r} U_{rn} S_n^{1/2} . \quad (3.18)
 \end{aligned}$$

Sub-step 3A: Form the vectors E_{1r} and E_{s1} as defined by Equations 3.12 and 3.13.

Sub-step 3B: Calculate the matrix $S_n^{1/2}$ using standard techniques.

Sub-step 3C: Calculate the state space model (A,B,C) using the definitions provided in Equation 3.18.

3.1.2 Using the ERA M-file.

The Eigensystem Realization Algorithm has been implemented in a MATLAB m-file called "ERA" by Medina (see appendix). This m-file may be used for the MIMO case. However, for purposes of this thesis it is only necessary to discuss the SISO application.

The command line for the m-file contains five input arguments and 3 output arguments. This is written as $[a,b,c]=era(h,m,r,s,norder)$, where the output arguments a , b , and c are the A , B , and C matrices, respectively. A description of the input arguments follows:

- h - Column vector (SISO case only) containing the experimentally measured discrete pulse response.
- m - Integer specifying the number of inputs to the system. For the SISO case, $m=1$.
- r,s - Integers specifying the row and column dimensions, respectively, of the Hankel matrices. Both r and s should be at least as large as the estimated order of the system.
- $norder$ - "optional" integer specifying the desired order of the state space model. If no value is entered for $norder$, then the singular values of the Hankel matrix will be displayed, allowing the user to form and enter a singular value based estimate of the system order (see explanation on page 17).

3.1.3 Limitations of the Algorithm and M-file.

The Eigensystem Realization Algorithm suffers from three notable limitations. First, the algorithm is not directly capable of modeling data exhibiting non-causal characteristics. This limitation follows from the fact that ERA forms models from pulse response data which is causal for any physically realizable system. However, it will be shown in later discussions that non-causal data may be manipulated into a form that ERA is capable of modeling. The models obtained from such manipulated data must also undergo various manipulations in order to arrive at a model of the original data (see Section 3.3).

The second limitation of the algorithm follows from the fact that ERA forms its models from a finite number of data points. From Section 3.1.1, ERA models are constructed by retaining the largest singular values of the Hankel matrix defined in Equation 3.5. Because of computational constraints, this Hankel matrix must have limited dimensions. Therefore, the number of data points used is limited to the first $r+s-1$ elements of the discrete pulse response. Hence, ERA is less capable of identifying low frequency characteristics which occur over longer periods of time (i.e. more data points of the discrete pulse response are needed).

The third limitation of the algorithm is that, unlike other modeling techniques such as the Transfer Function Determination Code^[4, pp.7-16] [10] and the Residue Identification

Algorithm^[4, pp.17-30], ERA does not allow for weighting of certain "important" frequencies. Using ERA, it is not possible to emphasize or de-emphasize particular portions of the frequency response.

The m-file "ERA", discussed in Section 3.1.2, naturally suffers from all of the limitations of the algorithm. Also, it has been observed that there exists a numerical problem within the mathematical manipulations of the spectral modeling process (described in Section 3.3). This problem does not appear to be caused by the implementation of ERA, but rather by the transformations that precede or follow its usage. The results of this numerical problem limit the application of the spectral modeling process to data that lies within a 80 decibel range. Beyond this range, obvious distortion occurs. This trend was observed on both the Compaq Deskpro 386/20e[™] and the SPARC 2[™] workstation.

3.2 Motivations for Modeling Spectral Data.

Given state space models of a system's spectral data, it is possible to gain information about the I/O characteristics of that system. In particular, it is possible to approximately determine the location of the poles and zeros of the system. Recalling from Chapter 2,

$$S_{y_i y_i}(z) = \frac{K_i^2 N_i(1/z) N_i(z) N_x(1/z) N_x(z)}{D_s(1/z) D_s(z) D_x(1/z) D_x(z)} \quad (3.19)$$

$$S_{y_i' y_i'}(z) = S_{y_i y_i}(z) \quad (3.20)$$

where y_i is any actual output signal, y_i' is any measured output signal, and Equation 3.20 follows from the fact that a large signal to noise ratio is assumed. By examining Equations 3.19 and Equation 3.20 it may be observed that if it is possible to form a state space approximation $(A_{\text{AUTO}}, B_{\text{AUTO}}, C_{\text{AUTO}}, D_{\text{AUTO}})$ of the autospectrum of any measured output signal, then it is also possible to approximately determine the location of the poles of both the system I/O transfer functions, $D_s(z)$, and the system inputs (or "excitations"), $D_x(z)$. Specifically, these poles are the eigenvalues of the matrix A_{AUTO} ,

$$\text{Eig}[A_{\text{AUTO}}] = \text{Roots}[D_s(1/z) D_s(z) D_x(1/z) D_x(z)] . \quad (3.21)$$

If a reliable set of criteria can be developed to separate the system poles from the excitation poles, then it is possible to construct an estimate of the denominator polynomial of the I/O transfer function as defined by

$$H_i(z) = K_i \frac{N_i(z)}{D_s(z)} . \quad (3.22)$$

A discussion of various potential separation criteria is presented in Chapter 5.

Similarly, if it is possible to find a state space model $(A_{\text{RATIO}}, B_{\text{RATIO}}, C_{\text{RATIO}}, D_{\text{RATIO}})$ of the ratio of cross-spectrum to autospectrum as defined by the equation

$$\frac{S_{y_i y_j}(z)}{S_{y_i y_i}(z)} = \frac{K_j N_j(z)}{K_i N_i(z)} \quad (3.23)$$

then an approximation of the I/O transfer function zeros may be found by examining the eigenvalues of the matrix A_{RATIO} ,

$$\text{Eig}[A_{\text{RATIO}}] = \text{Roots}[N_i(z)] \quad . \quad (3.24)$$

3.3 Using ERA to Model Spectral Quantities.

The discussion in the previous section revealed the motivation for modeling spectral quantities. In this section, a spectral modeling technique which utilizes the Eigensystem Realization Algorithm^{[4, pp.31-41] [5] [6] [7]} is presented.

Recalling from Section 3.1, ERA is an algorithm that is capable of constructing a discrete state space estimation (A,B,C) of a system given that system's discrete pulse response data. ERA is not directly capable of modeling non-causal data. However, it will be shown that non-causal data can be manipulated into a form that may be modeled by ERA.

Because spectral data ($\Delta S_{AB}(z)$) exists in the frequency domain, it is necessary to convert it into the time domain in order to be useful in ERA. This task is accomplished by use of the inverse z-transform^[3, pp.30-34], and the resulting data represents a discrete correlation function ($\Delta R_{AB}(k)$) which has values for both positive and negative k (non-causal). Because the discrete correlation function exhibits non-causal characteristics, it too must undergo further manipulation in order to be useful in ERA.

It is possible to express discrete correlation functions as a sum of two discrete pulse responses and a constant. In

equation form this is

$$R_{AB}(k) = h_+(k) u(k) + h_-(-k) u(-k) + R_{AB}(0) \delta(k) , \quad (3.25)$$

where $h_+(k)$ and $h_-(-k)$ represent pulse responses equal to the positive and negative time halves of the discrete correlation function respectively and $R_{AB}(0)$ is the value of $R_{AB}(k)$ for $k=0$. For the case where $R_{AB}(k)$ is obtained from an autospectrum (i.e. $R_{AB}(k)$ is a discrete autocorrelation function) the positive and negative time pulse responses will be equal, $h_+(k)=h_-(k)$, because autocorrelation functions are even functions of the discrete independent variable k (time delay). The block diagram representation of Equation 3.3 is shown in Figure 3.1. Taking the double sided z -transform of Equation 3.3 yields

$$S_{AB}(z) = H_+(z) + H_-(z) + R_{AB}(0) \quad (3.26)$$

ERA can be used to form a state space approximation (A_+, B_+, C_+) equivalent to the rational function $H_+(z)$ from $h_+(k)$.

Figure 3.1 Discrete Correlation Function.

Similarly, ERA may be used to form (A_-, B_-, C_-) equivalent to the rational function $H_-(1/z)$ from $h_-(k)$. The state space model $(A_{INV-}, B_{INV-}, C_{INV-})$ equivalent to the rational function $H_-(z)$ may be found by applying the following operations

$$\begin{aligned}
 H_-(z) &= C_- ([1/z] I - A_-)^{-1} B_- = C_{INV-} (zI - A_{INV-})^{-1} B_{INV-} \\
 A_{INV-} &= A_-^{-1} \\
 B_{INV-} &= A_-^{-1} B_- \\
 C_{INV-} &= -C_- A_-^{-1}
 \end{aligned} \tag{3.27}$$

The complete state space description of the spectral quantity may be found by placing the state space models (A_+, B_+, C_+) and $(A_{INV-}, B_{INV-}, C_{INV-})$ in parallel to find an equivalent (A, B, C) and setting the D matrix equal to the scalar $R_{AB}(0)$. In equation form this is

$$\begin{aligned}
 A &= \begin{bmatrix} A_+ & 0 \\ 0 & A_{INV-} \end{bmatrix} & B &= \begin{bmatrix} B_+ \\ B_{INV-} \end{bmatrix} \\
 C &= [C_- \quad C_{INV-}] & D &= R_{AB}(0)
 \end{aligned} \tag{3.28}$$

Hence, the order of the final, composite model is the sum of the orders of the individual sub-models found with ERA.

Chapter 4

Test of Theory on a Constructed Example

In Section 3.2 it was shown that it is possible to determine the location of the poles and zeros of a system by examining models of the spectral characteristics of the system outputs. In this chapter the theory of Section 3.2 will be verified by an application to a constructed test system with known characteristics. Section 4.1 describes the constructed test system and the procedures necessary for acquiring spectral data. In Section 4.2, modeling techniques are applied to the spectral data, an I/O transfer function for the system is approximated, and the results are compared to the true I/O transfer function.

4.1 Test System Description and Data Acquisition Procedures.

In this section a constructed test system is described and the procedures for acquiring spectral data are outlined. Section 4.1.1 describes a constructed continuous-time test system. A method for obtaining a discrete-time representation of the test system is presented in Section 4.1.2, and a technique for obtaining experimental output test data is discussed in Section 4.1.3. Finally, in Section 4.1.4, a method of obtaining estimates of spectral quantities from experimental test data is presented. Many of the procedures outlined in this section contain references to functions in the Matlab software package.

4.1.1 Description of the Test System.

Consider the single-input, two-output, continuous-time system shown in Figure 4.1. For this system, the I/O

Figure 4.1 Single Input/Two Output System.

characteristics are defined by the fourth order continuous-time I/O transfer functions

$$\begin{aligned}
 H_{XY_1}(s) &= \frac{0.8s^2 + 0.013s + 2.3}{s^4 + 0.03s^3 + 5.0002s^2 + 0.06s + 4} \triangleq \frac{K_1 N_{XY_1}(s)}{D_s(s)} \\
 H_{XY_2}(s) &= \frac{-0.85s^2 - 0.011s - 1.6}{s^4 + 0.03s^3 + 5.0002s^2 + 0.06s + 4} \triangleq \frac{K_2 N_{XY_2}(s)}{D_s(s)} .
 \end{aligned} \tag{4.1}$$

An equivalent continuous-time state space description of the system is given by

$$\begin{aligned}
 A_c &= \begin{bmatrix} 0 & 1.0 & 0 & 0 \\ -1.0 & -0.01 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -4.0 & -0.02 \end{bmatrix} & B_c &= \begin{bmatrix} 0 \\ 1.0 \\ 0 \\ 1.0 \end{bmatrix} \\
 C_c &= \begin{bmatrix} 0.5 & 0 & 0.3 & 0 \\ -0.25 & 0 & -0.6 & 0 \end{bmatrix} & D_c &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} .
 \end{aligned} \tag{4.2}$$

For this system, the first complex conjugate pair of poles has a corresponding damping ratio, ζ_1 , of 0.005 and natural

frequency, ω_{n1} , of 1 radian/second. For the second complex conjugate pair of poles, $\zeta_2=0.005$ and $\omega_{n2}=2$ radians/second.

4.1.2 Obtaining a Discrete-time Representation.

A discrete-time state space realization can be found by using the function C2D^[11,p.2-16] from Matlab's Control System Toolbox. The function C2D assumes that a zero-order hold is applied at the system inputs, and the discrete state space realization is calculated using the matrix exponential which is, in turn, calculated via Pade' approximation. The command line for C2D is

$$[A_d, B_d] = c2d(A_c, B_c, 0.3) ,$$

where the input argument 0.3 is the chosen sampling period, i.e. $T_s=0.3$ seconds. This choice of the sampling period ensures that the sampling frequency, $f_s=1/0.3=3.3333$ Hertz, is significantly larger than twice the largest natural frequency of the system. Because the largest natural frequency of the system is $\omega_n=2$ rad/sec, 0.3813 Hertz, the chosen sampling period and corresponding sampling frequency are sufficient. The resulting discrete-time realization is

$$A_d = \begin{bmatrix} 0.9554 & 0.2951 & 0 & 0 \\ -0.2951 & 0.9524 & 0 & 0 \\ 0 & 0 & 0.8257 & 0.2815 \\ 0 & 0 & -1.1259 & 0.8201 \end{bmatrix} \quad B_d = \begin{bmatrix} 0.0446 \\ 0.2951 \\ 0.0436 \\ 0.2815 \end{bmatrix} \quad (4.3)$$

$$C_d = C_c = \begin{bmatrix} 0.5 & 0 & 0.3 & 0 \\ -0.25 & 0 & -0.6 & 0 \end{bmatrix} \quad D_d = D_c = \begin{bmatrix} 0 \\ 0 \end{bmatrix} .$$

4.1.3 Obtaining Experimental Time Domain Output Data.

Experimental time domain output data may be found by using the function `DLSIM`^[11,p.2-61] from Matlab's Control System Toolbox. `DLSIM` propagates the states of a given realization and calculates the corresponding output vectors. The command line for `DLSIM` is

$$y = dlsim(A_d, B_d, C_d, D_d, x) ,$$

where "x" is the system input vector sequence and the output array "y" will have two columns representing the outputs $y_1(t)$ and $y_2(t)$, respectively.

When `DLSIM` is invoked, data should be collected for a time span that is at least as long as the longest time constant of the system. The longest time constant can be determined by examining the lowest natural frequency, ω_{n1} , and the corresponding damping ratio, ζ_1 . Specifically, the time constant, T_c , is determined by calculating $T_c = 1/(\omega_{n1}\zeta_1) = 1/(1 \times 0.005) = 200$ seconds. Hence, a time vector, t , of length 2400 seconds will include approximately 12 cycles of the slowest natural frequency, thereby guaranteeing that a sufficient amount of data is collected.

A linearly spaced time vector can be generated by using the Matlab function `Linspace`^[12,p.3-123] which is invoked by executing the command line

$$t = linspace(0, 2400, 8000)' ,$$

where 0 seconds is the start time, 2400 seconds is the stop

time, and 8000 is the number of elements in the vector (note: $8000 \times T_s = 2400$ seconds).

In order to simulate the input to the SSME system (see Chapter 5) the input vector, x , should be a combination of Gaussian white noise and periodic components. Gaussian white noise with zero mean and unity variance can be simulated by the Matlab function `RAND`^[12,p.3-158]. First, the command line `rand('normal')` is executed to select a "normal" (or "Gaussian") distribution in `RAND`. Next, the command line

$$x = rand(t) + 30 * (\sin(0.8*t) + \sin(1.3*t) + \sin(2.5*t))$$

is executed to obtain a input vector composed of simulated Gaussian white noise and three sinusoids (each having a peak magnitude of 30 [unitless] and frequencies of 0.8, 1.3, and 2.5 [rad/sec] respectively). The input vector, x , has the same dimensions as the time vector, t .

4.1.4 Obtaining Spectral Estimates from Time Domain Data.

Given the experimental output data obtained in Section 4.1.3, estimates of the spectral characteristics of the system may be obtained by using the Matlab function `SPECTRUM`^[13,p.2-96]. `SPECTRUM` uses the Welch periodogram technique^[14,pp.730-742] (with Hanning windows^[14,p.447]) to find averaged spectral estimates. The command line for `SPECTRUM` is

$$P = spectrum(x, y(:,1), 2048, 1200) ,$$

where x is the system input vector, $y(:,1)$ is the first output

vector (i.e. $y_1(t)$), 2048 is an integer specifying the number of data points for the Hanning windows, and 1200 is an integer specifying the number of overlapped points for each window (the selected window size and overlap will result in spectral estimates that have approximately eight averages). A similar command line should be executed using both output vectors as inputs so that the cross-spectrum between the output sequences may be obtained; i.e., replace vector "x" with "y(:,1)" and "y(:,1)" with "y(:,2)". The output array, P, has eight columns. The contents of these columns is as follows:

1. Sxx = autospectrum of input sequence x.
2. Syy = autospectrum of output sequence y.
3. Sxy = cross-spectrum from x to y.
4. Hxy = complex frequency response from x to y.
5. Cxy = coherence function between x and y.
6. Sxxc = 95% confidence interval for Sxx.
7. Syyc = 95% confidence interval for Syy.
8. Sxyc = 95% confidence interval for Sxy.

As an example of the output of SPECTRUM, Syy (output #1) is plotted in Figure 4.2. Examination of this plot reveals that there are five modes. As expected (see Section 3.2), two modes (which are caused by the system characteristics) occur at the frequencies $\omega=1$ rad/sec and $\omega=2$ rad/sec, respectively. The remaining three modes, which occur at the frequencies 0.8, 1.3, and 2.5 rad/sec respectively, are caused by the sinusoidal excitations contained in the input vector.

Figure 4.2 Magnitude Plot of S_{yy} (output #1).

4.2 Application of Theory to Test System.

Until now, discussions of modeling technique have been limited to the Eigensystem Realization Algorithm. Because of the dynamic range problems associated with the ERA based spectral modeling technique (see section 3.1.3), and because of the chronology of this application (this application was completed before the investigation of ERA), a new modeling technique will now be introduced. This technique is the Transfer Function Determination Code (TFDC) [4, pp.7-15] [10]. An in depth description of TFDC will not be given in this thesis. In summary, however, TFDC is a least squares curve fitting algorithm that yields a rational function (i.e. a transfer function) whose frequency response approximates the desired curve. Experience has shown that TFDC does not produce

satisfactory models of spectral quantities over the entire frequency continuum. Conversely, TFDC is capable of producing useable models of individual modes over small frequency ranges, and it is in this capacity that TFDC is utilized for this application.

Application of TFDC to the individual modes of the autospectrum of the first output signal, $y_1(t)$, yields the models shown in Figure 4.3. The spectral data points are represented by the symbols "+", and the models are shown as solid lines. Because of symmetrical properties of auto-

Figure 4.3 TFDC Models of Individual Modes.

spectrums, each mode was modeled by a fourth order TFDC model. Recalling from Section 3.2 (refer to Equations 3.19 and 3.21), the poles (or eigenvalues) of a realization of the output signal's autospectrum are a combination of the system poles (i.e. roots of $D_s(z)$), and the excitation poles (i.e. roots of

$D_x(z)$). The z-plane poles of the models, the corresponding natural frequencies, and the corresponding damping ratios are given in the Table 4.1. It may be observed that the natural

Mode #	Z-plane poles	Natural Frequency	Damping Ratio
1	0.9709+0.2397i	0.8068	5.511e-8
1	0.9709-0.2397i	0.8068	5.511e-8
1	0.9715+0.2372i	-0.7983	8.697e-9
1	0.9715-0.2372i	-0.7983	8.697e-9
2	0.9577+0.2958i	-0.9984	7.781e-3
2	0.9577-0.2958i	-0.9984	7.781e-3
2	0.9532+0.2944i	0.9985	7.808e-3
2	0.9532-0.2944i	0.9985	7.808e-3
3	0.9239+0.3826i	1.3087	7.857e-8
3	0.9239-0.3826i	1.3087	7.857e-8
3	0.9250+0.3799i	1.2989	1.701e-8
3	0.9250-0.3799i	1.2989	1.701e-8
4	0.8274+0.5664i	-2.0008	4.475e-3
4	0.8274-0.5664i	-2.0008	4.475e-3
4	0.8230+0.5633i	2.0008	4.477e-3
4	0.8230-0.5633i	2.0008	4.477e-3
5	0.7306+0.6828i	2.5054	9.777e-9
5	0.7306-0.6828i	2.5054	9.777e-9
5	0.7326+0.6806i	2.4955	4.030e-8
5	0.7326-0.6806i	2.4955	4.030e-8

Table 4.1 Location of Z-Plane poles, Natural Frequencies, and Damping Ratios

frequencies of the poles of the models for modes 1 through 4 have both positive and negative values. Specifically, for the

models of these modes two poles have positive natural frequencies and two poles have negative natural frequencies. This trend is caused by the quadrantal symmetry of autospectra poles, where the poles with positive natural frequencies are associated with the polynomials $D(z)$ and the poles with negative natural frequencies are associated with the conjugate polynomials $D(1/z)$. This trend does not hold true for the model of mode 5, and it is believed that this deficiency is caused by the fact that TFDC is not constrained to produce models with poles that exhibit quadrantal symmetry.

Further observation of the data in Table 4.1 reveals that the natural frequencies of the models' poles closely match the desired natural frequencies (i.e. 0.8, 1, 1.3, 2, and 2.5 radians/second for modes 1 through 5, respectively) and that the damping ratios for the system modes (modes 2 and 4) are relatively close (on the same order of magnitude) to the expected value of 0.005. Also, it is interesting to note the dramatic difference between the damping ratios of the system modes and the excitation modes (modes 1, 3, and 5); i.e., the damping ratios for the excitation modes are several orders of magnitude smaller than those of the system modes.

From Section 3.2 (refer to Equations 3.23 and 3.24), the poles of a realization of the ratio of the cross-spectrum of two output signals to the autospectrum of the first output signal are the zeros of one of the system I/O transfer

functions. Hence, the poles of an approximation of the ratio

$$\frac{S_{Y_1 Y_2}(z)}{S_{Y_1 Y_1}(z)} = \frac{K_2 N_{XY_2}(z)}{K_1 N_{XY_1}(z)} \quad (4.4)$$

will yield information about the zeros of the I/O transfer function $H_{XY_1}(z)$. The results of a TFDC application to the ratio defined in Equation 4.4 may be seen in Figure 4.4.

Figure 4.4 2nd Order TFDC Model of the Ratio of Cross-spectrum to Autospectrum.

Given the models of the output autospectrum and ratio of output cross-spectrum to autospectrum, an approximation of the I/O transfer function may now be formed. First it is necessary to examine the poles of the output autospectrum models. Based upon the data in the Table 4.1 and the known characteristics of the system, it is evident that the relative magnitude of damping ratios may be used as a criteria for separating the poles associated with the system from the poles associated with the periodic (sinusoidal) excitations. Hence,

eight poles, i.e. those of modes 2 and 4, may be presumed to be associated with the product $D_s(z)D_s(1/z)$. From the eight identified poles, the roots of the z-domain polynomial $D_s(z)$ may be selected by choosing those that occur within the unit circle of the z-plane. Finally, the approximate system determinant is

$$D_s(z) = z^4 - 3.5534z^3 + 5.1309z^2 - 3.5376z + 0.9912$$

Next, the system zeros, i.e. the roots of $N_{xy_1}(z)$, must be identified. These zeros are simply the poles of the model of the ratio of output cross-spectrum to output autospectrum as defined in Equation 4.4. Also, it is necessary to add a zero at $z=-1$ which does not appear in the ratio because of a cancellation of like terms in $N_{xy_1}(z)$ and $N_{xy_2}(z)$. This zero at $z=-1$ is a result of the sampling process; the zero order hold device (implemented in the m-file C2D) caused each of the numerator polynomials, $N_{xy_1}(z)$ and $N_{xy_2}(z)$, to have a zero at $z=-1$. Finally, the approximate numerator polynomial of the I/O transfer function is

$$N_{xy_1}(z) = z^3 - 0.7417z^2 - 0.7468z + 0.9948 ,$$

and the complete I/O transfer function approximation is

$$H_{xy_1}(z) = \frac{z^3 - 0.7417z^2 - 0.7468z + 0.9948}{z^4 - 3.5534z^3 + 5.1309z^2 - 3.5376z + 0.9912} . \quad (4.7)$$

In order to verify the accuracy of the transfer function approximation shown in Equation 4.7, its frequency response may be compared to the true (actual) frequency response. The true frequency response may be taken from the fourth column of

the P array from the output of SPECTRUM (refer to Section 4.1.4). The results may be seen in Figure 4.5 and 4.6. Examination of these plots reveals that the frequency response of the modeled transfer function is very similar to the true

Figure 4.5 Magnitude [dB] of Approximated and Actual Transfer Functions.

Figure 4.6 Phase [degrees] of Approximated and Actual Transfer Functions.

frequency response of the system. The only obvious difference in the frequency responses is the error in the estimation of

the constant gain K_1 (effects seen in the magnitude plot, Figure 4.5). This error is unavoidable because the modeling procedure, i.e. construction of a transfer function from the poles of spectral models, is only capable of identifying the poles and zeros of the system, and it does not provide any information about the constant K_1 . Hence, the constant K_1 must be determined by other means.

The results of the application presented in this section verify the theory presented in Section 3.2. The spectral modeling technique used, TFDC, does differ from the state-space model producing technique, ERA, discussed in Chapter 3. However, the models produced differ only in mathematical format, as it makes little difference whether one examines the poles of a transfer function model or the eigenvalues of a state-space model. The fact that the modes were modeled individually does not disqualify the conclusion; this was merely a formality caused by the limitations of TFDC.

Chapter 5

Application of Spectral Modeling Techniques to the SSME High Pressure Oxidizer Turbopump Pre-Flight Test Data

In this chapter, spectral modeling techniques will be applied to actual pre-flight test data from the high pressure oxidizer turbopump (HPOT) of the space shuttle's main engine (SSME). This data was taken during tests conducted at the NASA Stennis Space Flight Center and provided to Ohio University by the NASA Marshall Space Flight Center for research purposes under the grant NAG8-167. Section 5.1 briefly describes the system and summarizes relevant details of how the test was conducted. In Section 5.2, the results of an application of the ERA based spectral modeling technique are presented and discussed.

5.1 Description of the Pre-flight SSME Test.

A diagram of the SSME sensor complement may be seen in Figure 5.1. In this figure, the location of the HPOT is indicated by the letter H. Also indicated in Figure 5.1 are the HPOT sensor positions utilized during the pre-flight test. The HPOT is configured with eight strain gauges (SG) and eight accelerometers (AC). These sensors are mounted on the HPOT's housing weld in the 0°, 15°, 45°, and 60° positions as indicated in Figure 5.2. Two strain gauges and two accelerometers are mounted at each position. Because of this sensor configuration, the HPOT test system is very similar to

Figure 5.1 Layout of the SSME Sensor Complement.

Figure 5.2 HPOT Sensor Configuration.

the SIMO system indicated in Figure 2.2, where the sensors measure the system outputs and the excitation is internal to the system and, as such, is unmeasurable.

During the pre-flight test, the SSME was operated at various levels of thrust (or "rated power levels", RPL) ranging from 0 to 109% (see thrust profile, Figure 5.3). Data was collected by the sensors for 513 seconds at a sample rate of 10,240 Hertz. Upon completion of the test, the 513 second

Figure 5.3 Thrust Profile for Pre-Flight Test.

data records were broken into 0.4 second (4096 data points) segments and converted to the frequency domain using a fast

Fourier transform (FFT). The data was stored and provided to Ohio University in FFT form on magnetic tape.

5.2 Application of Modeling to Pre-flight Test Data.

In this section, a description of how the ERA based spectral modeling technique is be applied to the pre-flight test data is presented. First, Section 5.2.1 explains the reasoning used in the data selection process. Procedures for obtaining averaged spectral estimates from the HPOT's sensor measurements are outlined in Section 5.2.2 (these procedures differ slightly from those outlined in Section 4.1.4). In Section 5.2.3, a program called SPEC2MOD is introduced and its usage is described. The results of the application of SPEC2MOD to the SSME data are presented and analyzed in Section 5.2.4. Finally, in Section 5.2.5, two potential criteria that may be used to separate the system and excitation poles in autospectra models are discussed.

5.2.1 Data selection.

Before the modeling process may begin, it is necessary to decide which data points will be used. Because of the high degree of variability from one data record (one FFT block) to the next, it is necessary to conduct averaging in order to smooth the spectral estimates of the system.

Based upon information gained through discussions with NASA personnel, and through the insight of the associated

faculty members and graduate students at Ohio University, it is assumed that many of excitations are vibrations caused by irregularities in the bearings of the HPOT's rotor shaft. As such, these types of excitations will be periodic functions whose periods are a function of the rotor's speed. Because the SSME's fuel demands are directly related to the thrust level, it follows that the periods of the excitations will be related to the thrust, or rated power, level. In a frequency domain plot, a periodic function will exhibit a "spike-like" appearance, looking like a lightly damped mode, at its characteristic frequency. Figure 5.4 is a cascade mapping of the FFT blocks calculated from the test data. Examination of

Figure 5.4 Cascade FFT Mapping from SGO-1.

Figure 5.4 reveals that several of the lightly damped modes do, in fact, vary with the thrust level of the SSME (refer to

the thrust profile shown in Figure 5.3). Hence, in order to conduct averaging without "smearing" these modes across a range of frequencies, it is necessary to select data that is taken during a steady level of thrust. According to the thrust profile, the SSME is operated at a steady 109% rated power level during the time span of 76 to 427 seconds. It is from this time span that data is selected for this application.

Also, when selecting data it is necessary to consider the spatial separation of the sensors from which the data was measured. In order to gain useful information from the ratio of the cross-spectrum of two output signals to the autospectrum of one of the output signals, it is a necessary condition that the signals must neither be too closely related nor too vastly different. For example, if both of the strain gauges at the 0° position were selected, then the cross-spectrum between the two signals would most likely be very similar to the autospectrum of either of the two signals. Hence, the ratio would most likely be useless noise with a mean value of one. Also, if the sensors selected provided measurements of signals that were completely unrelated, i.e. the sensors were too far apart, then the cross-correlation function of the signals would essentially be zero and no useful information may be obtained from the corresponding cross-spectrum. The issue of sensor selection will not be thoroughly investigated in this thesis. It is, however, a

topic for future research for the author. For this application, the measurements of the first strain gauge at the 0° position (SG0-1) and the first strain gauge at the 45° position (SG45-1) will be used.

5.2.2 Obtaining Averaged Spectral Estimates.

The Matlab Signal Processing Toolbox function `SPECTRUM`^[13,p.2-96] can be used to obtain averaged spectral estimates of the data measured by the sensors mounted on the HPOT's housing weld. Recall from the description of `SPECTRUM`'s usage in Section 4.1.4 that `SPECTRUM` requires time domain data as its input. Because the test data provided by NASA is in block FFT form, it is necessary to convert it back into the time domain before it may be used in `SPECTRUM`. This task may be accomplished by using an inverse fast Fourier transform such as Matlab's Signal Processing Toolbox function `IFFT`^[13,p.2-53].

Before using `IFFT`, the data must be arranged into a column vector containing the positive frequency points listed in ascending order (from 0 to $f_s/2$ [Hertz]) and the negative frequency points listed in descending order (excluding the 0 and $-f_s/2$ [Hertz] frequency points). Because the data provided by NASA does not include the Nyquist frequency point, i.e. $f_s/2$, this point is set equal to zero and the corresponding error is accepted. Hence, the vector containing the entire FFT may be formed by implementing the command line

```
fullfft = [fftrec; 0; flipud(conj(fftrec(2:2048)))] ,
```

where the argument "fftrec" is the original FFT block as provided by NASA (2048 points), CONJ^[12,p.3-110] is a Matlab function that calculates the complex conjugate of its argument, and FLIPUD^[12,p.3-75] is a Matlab function that reverses the order of the elements of its column vector argument. IFFT may now be implemented via the command line

```
timerec = ifft(fullfft)
```

to obtain a column vector named "timerec" containing 4096 points of time domain data.

For this application, spectral estimates of the output signals with 100 averages and the original frequency resolution (2.5 Hertz/bin) were obtained by:

1. Selecting 100 FFT blocks corresponding to the time span of 299.6 to 339.6 seconds (region of steady thrust) from both SG0-1 and SG45-1.
2. Converting each individual block back into the time domain using the above mentioned technique.
3. Concatenating the individual time records into two 40 second time records, "sg01" and "sg451", corresponding to the measurements of SG0-1 and SG45-1, respectively.
4. Implementing SPECTRUM (refer to Section 4.1.4) via the command line

```
p = spectrum(sg01, sg451, 4096) ,
```

where 4096 is an integer specifying the size of the Hanning windows to be used in SPECTRUM.

5.2.3 Usage of the M-file SPEC2MOD.

The ERA based spectral modeling technique presented in Section 3.3 has been implemented in a Matlab m-file called SPEC2MOD (see appendix). SPEC2MOD is case specific to the HPOT problem; i.e. it has been designed for input arguments of specific dimensions and it internally assumes the sampling frequency that was used during the pre-flight test.

The command line for SPEC2MOD contains five input arguments and 5 output arguments. This command line is written as follows

$[a, b, c, d, w] = \text{spec2mod}(x, \text{option}, r, s, \text{norder})$,

where the output arguments a, b, c, and d are the respective matrices of the state space model, and the output argument w is a 2048 point frequency vector (0 to $\omega_s/2$ [radians/second]) that may be used when obtaining discrete bode plots. The input arguments are defined as follows:

- x - Column vector (2048x1) containing the spectral quantity that is to be modeled (positive frequencies only, i.e. the 0 to $\omega_s/2$ frequency points).
- option - Integer specifying the type of spectral quantity.
 - 1 = autospectrum
 - 2 = cross-spectrum
 - 3 = ratio (cross-spectrum/autospectrum)
- r,s - Integers specifying the row and column dimensions, respectively, of the Hankel matrices to be used in ERA (see discussion in Section 3.1.1).

norder - Integer specifying the order of the ERA sub-models.
 Actual order of the output state space model will be
 $2 \times \text{norder}$.

5.2.4 Results of Applications to SSME Data.

The results of an application of SPEC2MOD to the averaged autospectrum (obtained in Section 5.2.2) from the sensor SG0-1 may be seen in Figures 5.5 and 5.6. This model is 200th order, and it was generated by setting both the arguments "r" and "s" equal to 200 and setting the argument "norder" equal to 100. The Matlab Control System Toolbox function DBODE was used to obtain the frequency response from the state space model. Also shown in the magnitude plot of Figure 5.5 are the equivalent s-plane damping ratios; they are represented by the symbols "o" and they are plotted in decibels at their

Figure 5.5 Magnitude, 200th Order ERA Based Model (SPEC2MOD) of the Averaged Autospectrum of the Signal Measured by SG0-1.

Figure 5.6 Phase, 200th Order ERA Based Model (SPEC2MOD) of the Averaged Autospectrum of the Signal Measured by SG0-1.

corresponding natural frequencies. It may be noted that the damping ratios are relatively lower at the frequencies where the modes with the "spike-like" appearance occur.

The results of an application of SPEC2MOD to the ratio of the cross-spectrum (from the signal measured at SG0-1 to the signal measured at SG45-1) to the autospectrum of the signal measured at SG0-1 may be seen in Figures 5.7 and 5.8 (following page). Again, the model is 200th order, and it was generated with the same values for the input arguments r , s , and $norder$ as in the previous application.

The results shown in Figures 5.5 through 5.8 reveal that SPEC2MOD has produced models that fit the desired curves reasonably well. Experience has shown that these results are considerably better than the results obtained through applications of the Transfer Function Determination Code

(TFDC) [4, pp. 7-15] [10]. It is also worth noting that the technique used is capable of producing models of very high order.

Figure 5.7 Magnitude, 200th Order ERA Based Model (SPEC2MOD) of the Ratio of Cross-Spectrum to Autospectrum for the Signals Measured by SG0-1 and SG45-1.

Figure 5.8 Phase, 200th Order ERA Based Model (SPEC2MOD) of the Ratio of Cross-Spectrum to Autospectrum for the Signals Measured by SG0-1 and SG45-1.

In order to further evaluate the accuracy of these applications on a quantitative basis, the relative error may

be calculated by using the equation ($\|$ denotes "norm")

$$\epsilon = \frac{\| S_{actual}(e^{j\omega T}) - S_{model}(e^{j\omega T}) \|_2}{\| S_{actual}(e^{j\omega T}) \|_2}, \quad (5.1)$$

where $S_{actual}(e^{j\omega T})$ is the averaged spectral quantity and $S_{model}(e^{j\omega T})$ is the frequency response of the model. The results of this error analysis may be seen in Figures 5.9 and 5.10. In each case, the relative error is plotted in both unitless and decibels as a function of frequency. In these plots, it may be observed that the error decreases as frequency

Figure 5.9 Relative Error of the Autospectrum Model.

Figure 5.10 Relative Error of the Model of the Ratio of Cross-spectrum to Autospectrum.

increases in the model of the averaged autospectrum of the signal measured by SG0-1. In this case, the larger, low frequency error is caused by the limitations of ERA, since computational (and memory space) limitations restrict the allowable size for the Hankel matrices in ERA, thereby reducing the effectiveness at low frequencies (see discussion in Section 3.1.3).

For the model of the ratio, an opposite error characteristic occurs. That is, the error increases as frequency increases. This phenomenon is most likely caused by the large degree of variability in the high frequency range of the ratio of the averaged spectral data (refer to Figure 5.6). Visual examination of this data in the high frequency range will reveal a "noisy" appearing characteristic that is certainly difficult to model.

5.2.5 Separation of System and Excitation Modes.

It was mentioned in Section 3.2 that it is necessary to separate the system and excitation modes in output signal autospectrum models in order to arrive at an I/O transfer function approximation. For the HPOT system, the exact characteristics of the excitation are unknown because they are internal to the system and cannot be directly measured. It may be possible, however, to define a set of separation criteria based upon generalizations about the characteristics of the excitation and the system. Two potential criteria are

relative magnitude of damping ratios and modal fluctuation as a function of thrust.

Many of the excitations for the HPOT system are believed to be caused by irregularities in the bearings of the HPOT's rotor shaft. Such irregularities may cause periodic "pinging" (for example, a flat spot or a rough spot on the bearing) or sinusoidal oscillations along the shaft's axis. In either case, the excitations will be periodic, and they will exhibit the "spike-like" appearance, or lightly damped characteristic, in the autospectral plots. Hence, it may be possible to identify excitations by comparing the magnitudes of the damping ratios on a relative basis. This phenomenon may be observed in the data of Table 4.1 (from the constructed example) and in damping ratio plots of Figure 5.5.

Periodic excitations, like those mentioned above, may also have periods that are a functions of the rotor's speed. Therefore, they will be functions of SSME thrust because of the direct relationship between fuel demands and thrust. It may be possible to identify these types of excitations by tracking modal frequency fluctuation while the thrust of the engine is varied. Such modal frequency fluctuation may be seen in the cascade mapping shown in Figure 5.4 (page 45). Note: If this criteria is used, care must be taken to avoid false identification of system modes that fluctuate in frequency because of non-linearities within the system.

The two separation criteria presented in this section are not intended to be used blindly. They may, however, be used as tools for an individual who is intimately familiar with the system in question. Other such criteria may need to be investigated and utilized.

Chapter 6

Conclusion

This thesis has presented a technique for identifying the internal characteristics of a system through analysis of models of the spectral characteristics of the output signals. Throughout this research, numerous successes and some shortcomings were encountered. The theory necessary to perform system identification via spectral modeling was collected, organized, and derived when necessary. This presentation of theory began with a selection of basic equations used in random signal analysis and ended with a discussion of how the poles and zeros of a system may be identified by examining the spectral functions of the output signals. In addition, the theory behind the Eigensystem Realization Algorithm (ERA) for SISO system modeling was discussed and a technique for modeling spectral quantities which utilizes ERA was presented.

Applications of the spectral modeling theory were conducted using both the Transfer Function Determination Code (TFDC) and the ERA based technique. In an application to a constructed example, TFDC was used to verify the theory that it is possible to form approximations of the I/O transfer functions of a system from models of the spectral characteristics of the output signals provided that a reliable set of separation criteria and estimate of the system order are established. More specifically, TFDC was used to model

the autospectrum of an output signal and to model the ratio of cross-spectrum to autospectrum of two output signals. From the autospectrum model, the poles of the system and excitation were identified. From the ratio model, the zeros of the system were identified. A rational function model of one of the system's I/O transfer functions was constructed from the identified poles and zeros, and its frequency domain characteristic was compared to the known frequency response. The results of this comparison revealed that the constructed model provided a near perfect match in phase and that the magnitude plot differed only by a gain constant. This scaling error was expected because the technique used does not provide any means for estimation of a constant gain. During this application, it was determined that TFDC was not capable of producing satisfactory models of spectral quantities over the frequency continuum (the results of these attempts were not shown in this thesis) but that TFDC could be used to produce satisfactory models of individual modes. It is concluded that this limitation causes TFDC applications to become extremely cumbersome and effectively disqualifies TFDC's usefulness as a spectral modeling algorithm.

The ERA based spectral modeling algorithm was implemented in an m-file called SPEC2MOD and applied to pre-flight test data from the HPOT of the space shuttles's main engine. The results of these applications revealed that the ERA based technique is capable of producing satisfactory models of

spectral quantities of high order. However, when applied to the constructed test system, it was discovered that this technique suffers a limitation in dynamic range of approximately 80 decibels. That is, when this technique is applied to data from a system whose dynamic range exceeds 80 decibels the resulting frequency response (and model) fits only the upper 80 decibels of the plot and the remainder appears as a straight horizontal line (80 decibels down from the maximum value). This dynamic range problem was investigated, but no specific cause could be identified. It was determined, however, that the dynamic range problem was not caused by the implementation of ERA, but rather by the manipulations that precede or follow ERA's usage (see section 3.3). Coincidentally, the data provided by NASA did not exceed this dynamic range.

In response to a request by associated NASA personnel, the m-file SPEC2MOD and other related programs (such as DBODE, IFFT, etc.) are currently being written into code that is usable in the UNIX operating system for utilization at the NASA Marshall Space Flight Center. This task is being performed by graduate students at Ohio University and is expected to be completed by July, 1992.

Future research will be dedicated to the investigation of the sensor correlation issues discussed in Section 5.2.1. From this research, it is hoped that the questions concerning the optimal spatial separation of sensors will be answered.

Also, an investigation of the uses of different types of sensors, i.e. combined accelerometer and strain gauge experiments, will be conducted.

The process of identifying I/O transfer functions of the HPOT system has not been completed. However, two possible criteria for separating system and excitation modes have been investigated and discussed in this thesis. Namely, these are relative damping ratio and modal frequency fluctuation (see Section 5.2.5). More research needs to be dedicated to the investigation of these and other such criteria before a reliable estimate of the system determinant may be formed.

Bibliography

- [1] J. S. Bendat, A. G. Piersol, Engineering Applications of Correlation and Spectral Analysis, John Wiley & Sons, New York, 1980.
- [2] G. R. Cooper, C. D. McGillem, Probabilistic Methods of Signal and System Analysis, Holt, Rinehart, and Winston, Inc., Fort Worth, Texas, 1971.
- [3] C. L. Phillips, H. T. Nagle Jr., Digital Control System Analysis and Design, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1984.
- [4] E. A. Medina B., "Multi-input, Multi-output System Identification for Frequency Response Samples with Applications to the Modeling of Large Space Structures", M.S. Thesis, Ohio University, November 1991.
- [5] J-N. Juang, R. S. Pappa, "An Eigensystem Realization Algorithm for Modal Parameter Identification and Model Reduction", AIAA Journal of Guidance and Control, Vol. 8, No. 5, pp.620-627, Sept-Oct 1985.
- [6] J-N. Juang, R. S. Pappa, "Effect of Noise on Modal Parameters Identified by the Eigensystem Realization Algorithm", AIAA Journal of Guidance and Control, Vol. 9, No. 3, pp. 294-303, May-June 1986.
- [7] R. W. Longman, J-N Juang, "Recursive Form of the Eigensystem Realization Algorithm for System Identification", AIAA Journal of Guidance and Control, Vol. 12, No. 5, pp. 647-652, Sept.-Oct. 1989.
- [8] T. Kailath, Linear Systems, Prentice-Hall, Inc. Englewood Cliffs, New Jersey, 1980.
- [9] G. H. Golub, C. F. Van Loan, Matrix Computations, Second Edition, The Johns Hopkins University Press, Baltimore, 1989.
- [10] R. D. Irwin, "System Identification and Controller Design Using Experimental Frequency Response Data", Final Presentation Materials, 1990 NASA/ASEE Summer Faculty Fellowship Program, Marshall Space Flight Center, Alabama, August, 1990.
- [11] A. Grace, A. J. Laub, J. N. Little, C. Thompson, Control System Toolbox for use with MATLAB, Users's Guide, The MathWorks, Inc., South Natick, MA, October 30, 1990.

- [12] 386-MATLAB[™] User's Guide, The MathWorks, Inc., South Natick, MA, October 15, 1990.
- [13] J. Little, L. Shure, Signal Processing Toolbox for use with MATLAB[™], User's Guide, The MathWorks, Inc., South Natick, MA, August 29, 1988.
- [14] A. V. Oppenheim, R. W. Schaffer, Discrete-Time Signal Processing, Prentice-Hall, Englewood Cliffs, New Jersey, 1989.

Appendix

MATLAB M-Files

The software used during this research is in the form of Matlab m-files. All special Matlab m-files which were written by graduate students at Ohio University and used during this research are listed in this appendix. Many of the m-files referenced in this thesis may be found in the standard Matlab library or in the various specialized toolbox libraries of Matlab; these m-files are not listed in this appendix.

```

function [a,b,c]=era(y,m,r,s,norder)

% An m-file that implements the Eigensystem Realization
% Algorithm for the identification of a realization (a,b,c) of
% a system from its impulse response. The Markov parameters
% of the system are entered as matrix Y, which has p rows
% (p=number of outputs) and nt*m columns (m=number of inputs,
% nt= samples taken) For the r*p x s*m Hankel matrix H(k): r:
% row repetition (number of blocks y( ) in a column). s:
% number of column blocks in a row (number of samples per
% row). If norder is not given, the singular values of the
% Hankel matrix will be displayed (allowing the user to
% estimate the order), and then the user must input norder
% when prompted.
%
% SEE ALSO: HANKERA.M.
%
% Use as [a,b,c]=era(y,m,r,s,norder);
% Enrique A. Medina.
% ECE Department. Ohio University. June, 1991.

[p,dummy]=size(y); % p=number of outputs of the system

% Form Hankel Matrices H(0) and H(1):
hank0=hankera(y,1,m,r,s);
hank1=hankera(y,2,m,r,s);
[mh,nh]=size(hank0);
disp(['Using ',int2str(mh),' by ',int2str(nh),' Hankel
                                     matrices.'])

% Singular Value Decomposition of H(0):
disp('Singular value decomposition of Hankel matrix.')
[pp,dd,qq]=svd(hank0);
hsv=diag(dd);
format short e

if nargin == 4
    disp('      Hankel Singular Values:')
    if length(hsv) > 30
        disp(hsv(1:30))
        pause
        disp(hsv(31:length(hsv)))
    else
        disp(hsv)
    end
    disp('You can look at the singular values of the Hankel
                                     matrix again!')

    keyboard
    kk=input('Please assign the desired order of the
                                     realization >');

% continued on following page

```

```

% era continued

else
    kk=norder;
end

% Construct Realization of the Specified Order:
% Truncate SVD of H(0).
dd=diag(hsv(1:kk));pp=pp(:,1:kk);qq=qq(:,1:kk);
dtohalf=dd^(.5);
dtomhalf=dd^(-.5);
ept=[eye(p) zeros(p,(r-1)*p)];em=[eye(m);zeros((s-1)*m,m)];
a=dtomhalf*pp'*hank1*qq*dtomhalf;
b=dtohalf*qq'*em;
c=ept*pp*dtohalf;

```

```

function h=hankera(y,k,m,r,s)

% An m-file that forms the Hankel matrix H(k) given the
% Markov parameters y(k) of the system in matrix Y [p x
%                                     (nt*m)].
% p: number of outputs.
% m: number of inputs.
% nt: number of samples.
%
% The (r*p x s*m) Hankel matrix is formed as:
%
%
%      | y(k)      y(k+1)    y(k+2) ... y(k+s-1) |
%      | y(k+1)    y(k+2)    y(k+3) ... y(k+s)   |
% H(k) = | .        .        .        .          |
%      | .        .        .        .          |
%      | .        .        .        .          |
%      | y(k+r-1)  y(k+r)    ...      ... y(k+r+s-2) |
%
% s: number of column blocks y( ) in a row (number of
%                                     samples in a row).
% r: number of blocks y( ) in a column (row repetition
%                                     number).
%
% Use as h=hankera(y,k,m,r,s);
%
% Enrique A. Medina. ECE Department.
% Ohio University. June, 1991.

[p,ly]=size(y); % p: number of outputs of the system.

if ly < (s+r-1)*m;
    error('Not enough samples in Y')
end

h=zeros(r*p,s*m);

for ii=1:r
    h((ii-1)*p+1:ii*p,:)=y(:,(k+ii-2)*m+1:(k+ii-2+s)*m);
end

```

```

function [a,b,c,d,w] = spec2mod(x,option,r,s,norder)

% JOSEPH R. BROWN, DAVID L. BARTHOLOMEW
% SSME M-FILE
% 12/11/91
% ECE DEPARTMENT, OHIO UNIVERSITY
%
% This m-file contains the Matlab code necessary to take an
% autospectrum, cross-spectrum, or a ratio of the two and
% obtain a state space model in the form of a realization
% (a,b,c,d). The Matlab m-files used include era.m and
% fullfft.m.
%
% Inputs:
% x - input autospectrum, cross-spectrum, or the ratio of
%     cross-spectrum to autospectrum
% option - is an integer specifying the type of input for x
%           1 = autospectrum
%           2 = cross-spectrum
%           3 = ratio (cross-spectrum / autospectrum)
% r - number of rows in the Hankel matrix formed in
%     hankera.m
% s - is the number of columns in the Hankel matrix
%     norder - order of the model
%
% Outputs:
% a,b,c,d - state space realization of the model
% w - frequency vector corresponding to the model
%
function [a,b,c,d,w] = spec2mod(x,option,r,s,norder)
%
% NOTE: Vector dimensions, frequencies, sampling periods,
% etc. etc. are specialized to the SSME problem.
% Significant modification will be necessary in order to
% convert this m-file to the general case.

w = linspace(0,10240*pi-10240*pi/2048,2048);
ts = 1/10240;
xfull = fullfft(x);
ifftofx = ifft(xfull);
postimex = ifftofx(1:2049);
negtimex = [ifftofx(1);flipud(ifftofx(2049:4096))];

if (option == 1)
    [apos,bpos,cpos,dpos] = era(postimex',1,r,s,norder);
    aneg = inv(apos);
    bneg = aneg*bpos;
    cneg = -cpos*aneg;
    dneg = dpos + cneg*bpos;
    [a,b,c,d] =
        parallel(apos,bpos,cpos,dpos,aneg,bneg,cneg,dneg);
% continued on following page

```

```

% spec2mod continued

elseif ((option == 2) | (option == 3))
    [apos,bpos,cpos,dpos] = era(postimex',1,r,s,norder);
    [aneg,bneg,cneg,dneg] = era(negtimex',1,r,s,norder);
    anegc = inv(aneg);
    bnegc = anegc*bneg;
    cnegc = -cneg*anegc;
    dnegc = dneg + cnegc*bneg;
    [a,b,c,d] =
        parallel(apos,bpos,cpos,dpos,anegc,bnegc,cnegc,dnegc);

else
    disp('The number you entered for parameter number two is
                                                invalid')
    disp('You must enter 1,2, or 3.')
end

```

```

% spec2mod continued

elseif ((option == 2) | (option == 3))
    [apos,bpos,cpos,dpos] = era(postimex',1,r,s,norder);
    [aneg,bneg,cneg,dneg] = era(negtimex',1,r,s,norder);
    anegc = inv(aneg);
    bnegc = anegc*bneg;
    cnegc = -cneg*anegc;
    dnegc = dneg + cnegc*bneg;
    [a,b,c,d] =
        parallel(apos,bpos,cpos,dpos,anegc,bnegc,cnegc,dnegc);

else
    disp('The number you entered for parameter number two is
        invalid')
    disp('You must enter 1,2, or 3.')
end

```



```

function [y] = fullfft(x)

% DAVID L. BARTHOLOMEW
% SSME M-FILE
% 12/2/91
% ECE DEPARTMENT, OHIO UNIVERSITY
%
% This m-file takes SSME frequency data as input in the form
% of a 2048 point complex array and outputs a 4096 point
% array which contains both positive and negative
% frequencies. Note, the half-sampling frequency point was
% omitted in the data received from NASA. It was decided to
% set this point equal to zero and accept the error this
% causes.
%
% Example:
%   Input   x = [x(1);x(2);x(3);x(4)]
%   Output  y =
%   [x(1);x(2);x(3);x(4);0;conj(x(4));conj(x(3));conj(x(2)) ]
%
% function[y] = fullfft(x)

y = [x;0;conj(flipud(x(2:length(x))))];

```