USING ARMA MODELS TO IDENTIFY MODAL PARAMETERS FOR FLUTTER BOUNDARY PREDICTION

By

COREY L. ECKHART

Bachelor of Science

Oklahoma State University

Stillwater, Oklahoma

1993

Submitted to The Faculty of the
Graduated College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
May, 1998

USING ARMA MODELS TO IDENTIFY MODAL

PARAMETERS FOR FLUTTER

BOUNDARY PREDICTION

Thesis Approved:

Thesis Advisor

Prabhakar Pagilla

Dean of Graduate College

ACKNOWLEDGEMENTS

I wish to express my deepest appreciation to my major advisor, Dr. Andrew S. Arena, for his guidance, inspiration, and especially his patience. Without his teachings and patience, I could not have completed my graduate research and studies. I would also like to express my appreciation to my other committee members, Dr. G. E. Young and Dr. P. R. Pagilla.

I would primarily like to thank my wife, Jennifer, for her understanding, support and love especially through the time of my research. She was the driving force for me to complete this work. This graduate thesis is dedicated to her.

TABLE OF CONTENTS

Chapter | Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

NOMENCLATURE

a_0	Coefficient in curve fit equation, the offset or static value
a_i	i^{th} auto-regressive coefficient for AR and ARMA models
a_k	Coefficient in curve fit equation, kth cosine term
ASE	Aeroservoelastic Stability Analysis
b_i	i^{th} moving average coefficient for ARMA models
b_k	Coefficient in curve fit equation, kth sine term
C	Static offset in sinusoidal damped system equation
E	Mean-squared error for curve fitting
f_d	System damping frequency, (Hz)
f_n	System natural frequency at Q = 0, (Hz)
f_s	Sample frequency of input data (Rad/S)
F	Frequency Ratio, sample frequency divided by system frequency of interest (ω_s/ω_n) or (f_s/f_n)
GHV	Generic Hyperspace Vehicle
h	Step size, seconds
i	i^{th} data point in digitized time history or increment in AR and ARMA model
j	Each mode
k	Starting data point to apply AR or ARMA model (k=2*M+1+z) or k^{th} increment
M	Number of system modes
m	Modal index in curve fit equation or Do-loop counter

MOSE	Method of Overdetermined set of Simultaneous Equations
MPCONV	Programming Flag set if regressive convergence occurred
n	Re-sample factor is the number which divides the sample frequency of the input data
N	Total number of points including input.
ON-DLS	On-Line Double Least Squares
ON-LS	On-Line Least Squares
$[P]_k$	Covariance Matrix at time k for on-line formulations
Q	Dynamic Pressure, lb/ft^2
r	Programming lag set from identification of good and bad roots or eigenvalues
STARS	Structural Analysis RoutineS
t	Time, seconds
u_{k-i}	Excitation or forced input at k-i^{th} time increment for ARMA model
W	Number of moving average coefficients in ARMA model
y(t)	Input time history data
y_k, y_{k-i}	Input at k^{th} and k-i^{th} time increment for AR and ARMA models
Y(t)	Curve-fitting expression
z	Last point of excitation in ASE data
$[\alpha]_k$	Vector of data values at k^{th} data points used to update on-line solutions
$[\beta]_k$	Vector of data values at k^{th} data points used to update on-line solutions
$[\Phi]$	Regression Data Matrix (n x m+1 size)

λ	Roots of characteristic equation
$\{\Theta\}$	Auto-Regressive coefficient Vector (m+1 size)
$[\psi]$	Data Vector (n size)
σ	Damping Product ($\zeta\omega_N$)
ς	Damping Factor
ς_k	k^{th} damping factor for curve fit equation
ω_d	System damping frequency, (Rad/s)
ω_n	System natural frequency at Q = 0, (Rad/s)
ω_s	Sample frequency of input data (Rad/S)

CHAPTER 1

INTRODUCTION

1.1 Research Objective and Background

A flutter boundary can be defined as the point at which an area of interest in the model or aircraft begins to inherently have an instability. More specifically, the damping becomes neutral or even worse, unstable. Flutter boundaries can primarily be estimated either from numerical simulations of an aircraft model or analytical examinations of aircraft flutter test data. The goal is to determine the flutter onset speed, being the flutter boundary, by examining areas in the aircraft that have been predicted to have marginal stability at sub-critcal speeds. In aircraft flutter testing, the obtained data must be reduced in a nearly real-time sense to determine the modal parameters (damping frequency, ω_d, and damping factor, ζ) in question. These modal parameters are then used to analyze the closeness to the flutter boundary. This method involves several hours of engineering and flight time, costly instrumentation, noisy data, risky decisions, and most importantly safety issues. Today, anything from classical methods such as fast-fourier transform coupled with power spectral density analysis to more modern methods such as system model identification have been applied to flutter test results at sub-critical speeds to determine such modal parameters.

On the other hand, numerical simulation of an aircraft model can be used to predict the flutter boundary. This method uses an effective and efficient computational simulation for determining the aeroelastic response resulting from an excitation. However, these results are usually backed up by flutter testing, but the flight time, safety, and engineering hours are significantly reduced.

A useable aeroelastic response, being the mode shapes of the model or aircraft, should be determined using an aerodynamic code coupled with a structural dynamics code (Dowell, 1995). The Structural Analysis RoutineS or STARS program and its derivatives, currently implemented at NASA Dryden Flight Research Facility, is a computational method of this kind. STARS is a multidisciplinary program integrating modules from structural, to computational fluid dynamics, to aeroservoelasticity which is capable of performing linear and non-linear modeling and simulation of advanced aerospace vehicles (Gupta & Peterson, 1992). Its non-linear Aeroservoelastic Stability Analysis (ASE) module (ASENL_UNSTEADY code) provides

1) an initial finite element structural modeling and free vibration analysis yielding natural frequencies of all modes and
2) the solution to the generalized equation of motion in the state-space equation form thus yielding "noise-free" response data in the shape of generalized displacements and velocities for each individual mode for the model in question.

The positive aspect of this method is that the ASE Module provides multiple independent mode shapes. Specifically, a single time history of ALL the modes in question is not resulted in as in flight test obtained data, but a single time history for EACH mode is resulted in without any noise.

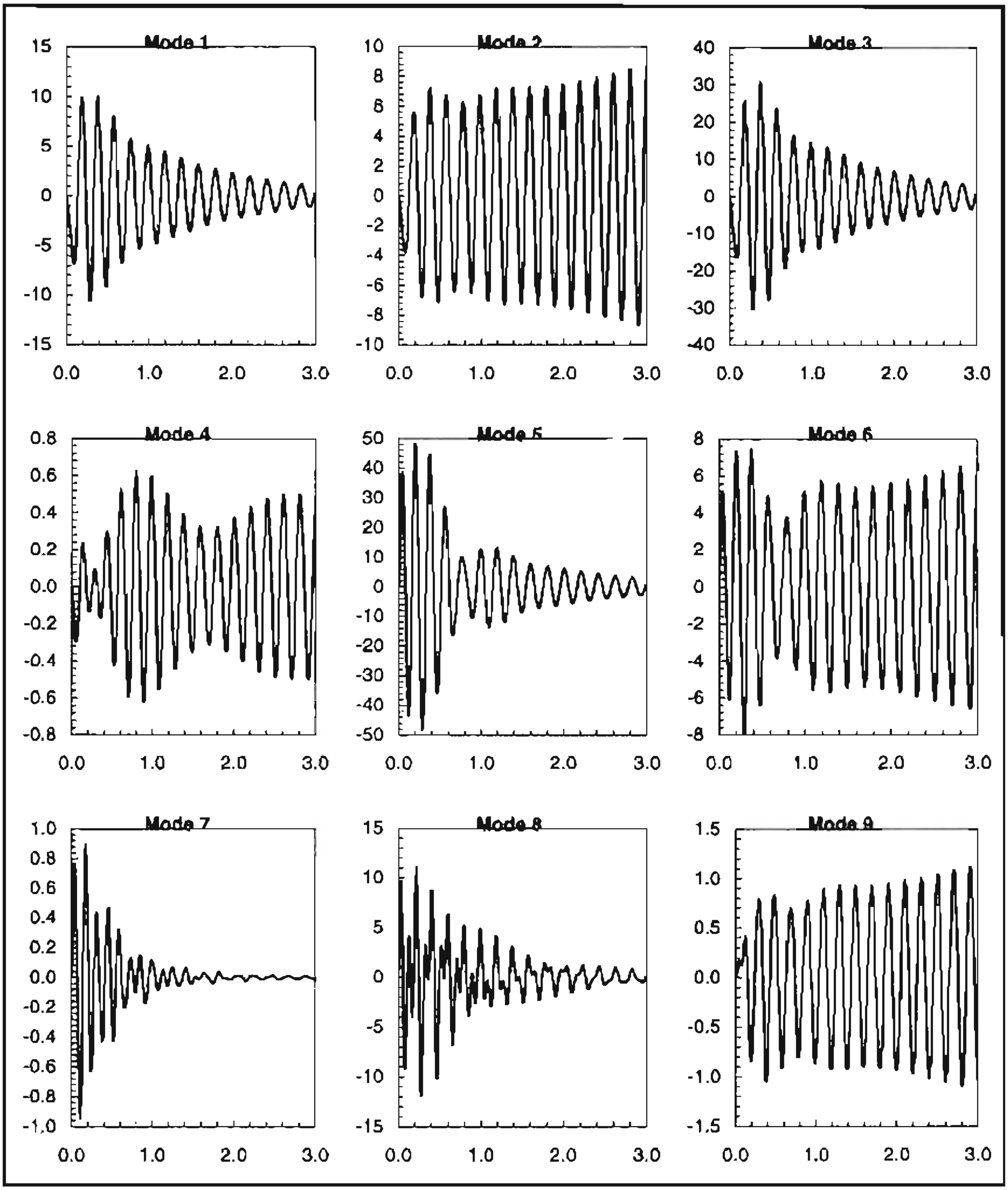


Figure 1. Example of Mode Shape Responses From STARS - Nine Mode GHV Model

Theoretically, each mode shape may contain any or all of the other modes inherent within the system. These mode shapes include an excitation, but once the excitation is complete, the response is basically free from any other structural excitation. However, aerodynamics forces are applied during the response. Figure 1 shows an example of each independent mode shape from the ASENL_Unsteady Program in STARS. These mode shapes were determined using the Generic Hyperspace Vehicle (GHV) model resulting in nine modes. Future references to this type of data will be called ASENL data.

The negative aspect of the time marching approach used in the ASENL_Unsteady Program is that it only provides time history data of each individual mode in terms of generalized displacements and velocities. It does not provide modal parameters or other types of stability parameters. Currently with this ASENL data, the method to identify the flutter onset boundary is to run the program for a model at several specific speeds, pre and post flutter boundary, for several seconds or cycles of response which could take days or even weeks of computational time. The generalized displacements are then graphically plotted, and the responses are visually examined to determine whether the response is converging, diverging, or neutral. Therefore, this method can only result in a flutter boundary prediction with an error of determination due to human judgment.

The objective of this research is to replace this graphical method. A method must be determined which will autonomously assist in predicting the system's flutter boundary from multiple mode time history data, specifically the results from the ASENL_Unsteady Program in STARS. The boundary shall be determined without knowing any information about the magnitudes of the aerodynamic forces or excitation during the responses in the least amount of cycles (data samples) and computational time. Whether using the most common method of plotting the damping factor against the dynamic pressure for each test case or some other method of determining the flutter boundary, this research shall investigate several methods. Once a complete method is determined, it will be developed into a stand-alone, autonomous program which will be used in conjunction with the ASENL_Unsteady Program in STARS to determine the flutter boundaries of any multiple mode system.

1.2 Literature Review

1.2.1 Modal Parameter Identification

In the past, several methods have been employed to determine the damping frequency and the damping factor. When these two parameters are determined, specifically the damping coefficient, the idea is to plot the damping coefficient against a speed or dynamic pressure, and determine the flutter boundary when the damping factor is zero, being neutral damping. In flutter flight testing, subcritical speeds are analyzed and a flutter boundary is determined. It is determined through some sort of extrapolation of the damping factor because the actual flutter point cannot be determined in flight due to safety. Unlike in flutter flight testing, numerical simulation can be accomplished at the flutter boundary and beyond if need to be. Therefore, no extrapolation, a cause for error, needs to occur.

The next sections discuss methods to identify modal parameters which have been used extensively with aircraft flutter test data and numerical simulations.

1.2.1.1 Curve-Fitting

The curve-fitting method extracts frequency, damping, amplitude, and phase information from unforced transient response data (Bennett & Desmarais, 1975). This method is designed to curve fit digitized time history data in a least squares sense using the non-linear exponential function:

$$Y(t) = a_0 + \sum_{k=1}^{M} e^{-\zeta_k \cdot t} \cdot \left[a_k \cdot \cos\left[\left(\omega_d\right)_k \cdot t \right] + b_k \cdot \sin\left[\left(\omega_d\right)_k \cdot t \right] \right]$$

This equation minimizes the squared error difference between the output fit and the input time history for which this error is given by

$$E = \sum_{i=1}^{N} \left(Y\left(t_i\right) - y_i \right)^2$$

This method does require the number of exact modes in the input data to be known and a very good starting guess for all five parameters (a_0, a_k, b_k, ς_k, and $(\omega_d)_k$). From the initial guess and an inputted step size, the data is sequenced through until the error is minimized. This method is sound, however, Bennett and Desmarais only provide a method up to two modes. Recall, more than two modes are feasible to be embedded in each individual mode shape output from the ASENL Program of STARS.

A similar method was applied as a class assignment. This method involved using a least squares non-linear curve fit using the non-linear equations above coupled with Newton's Method (Gerald and Weatley, 1994) to determine the five parameters of interest. The results were obtained for only one mode, but, the initial guess had to be almost equal to the final results for the method to work. Also, several points were needed from the input time history.

The problem with curve-fitting is that it usually is feasible for one mode. With two or more modes, the difficulty of applying the above equation becomes greater and accuracy is degraded due to more calculations (Bennett and Desmarais, 1975). The results of this method have been said to depend strongly on the initial parameters and requires several data points (Pak and Friedman, 1992), which will be shown later in this paper. This method has never been shown for three or more modes, and this method requires the number of modes to be known. In the ASENL data for each mode shape, the number of modes embedded within is not known and may very well be more than three. Even if all the mode shapes were normalized and summed together, the number of modes are known, but, in some cases there are nine modes, thus requiring a very complex curve equation and a very good initial guess. Due to these negative aspects, the curve-fitting routine is not feasible for a solution to this research.

1.2.1.2 Fast Fourier Transform Coupled With Power Spectral Density (FFT/PSD)

FFT has been a very popular method. When used along with Power Spectral Density (PSD) data plots, the damping frequency and damping factor can be estimated. The damping frequency can directly be determined from the PSD plot of local maximas, and the damping coefficient can be determined using the half-power law or some sort of curve fit (Lenz & McKeever, 1975; Dobbs & Hobson, 1979; Kehoe, 1988).

Another similar method is the Moving-Block Analysis (MBA) developed in 1975 (Bousman & Winkler, 1981). The MBA first uses the entire data set to determine one frequency of interest using an FFT and PSD analysis. A block length of time is then selected, usually a ¼ to ½ of the signal length, and the natural log of the so-called moving block function, which is developed from the finite fourier transform of the damped sinusoidal response equation, is obtained. This procedure is repeated for the next block and so on until all the data set has been reduced. These results are plotted against the period of the sample set which results are linear. The slope of this curve results in the damping factor. This method usually involves the filtering of data and hands-on decisions of block sizes and the critical frequencies of interest. This method has only been proven with two modes or less and has difficulty with closely spaced modes.

Some other difficulties with FFT with PSD analysis in is filtering. Determining which type of filter and cutoff frequencies to use to either to reduce the noise in the data (flight test data) or to obtain only the frequencies of interest. This can be complex and is usually a hands-on decision by the engineer. Along with filtering, auto-correlation, zeroing, and data smoothing are also used to assist in determining each mode. With all these factors involved, each one plays an important role in obtaining good accuracy, and it usually determines only one mode of interest (Kehoe, 1988). The ASENL data is "noise-free" so these routines to cancel noise will not be a problem. However, due to the multiple modes contained within each mode shape, it may be difficult to determine all modal frequencies.

Another difficulty with FFT/PSD is closely spaced modes and/or very dominant modes, whether it be two or nine. Figure 2 shows an example of a simple FFT/PSD using MATLAB 4.0 for each mode shape plot from the time history data in Figure 1. From Figure 2, it is obvious that a very dominant mode exists in each mode shape, the other modes are not seen very clearly. Even for a normalized summation of all nine modes presented in Figure 3, it provides difficulties in determining all nine modes without using extensive filtering and knowledge of good cutoff frequencies which must be a hands-on decision.

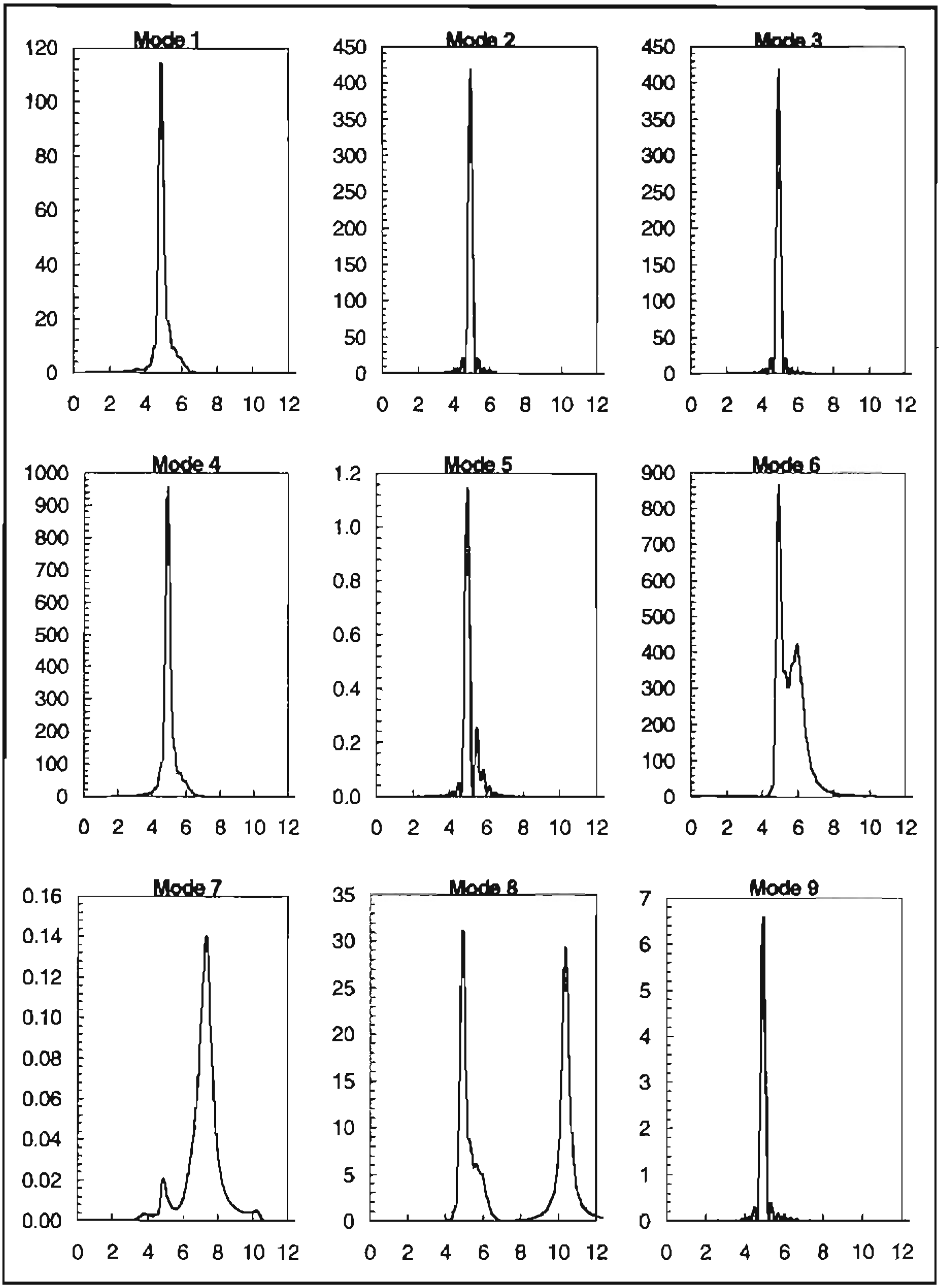


Figure 2. PSD Plots of Each Mode Shape in Figure 1

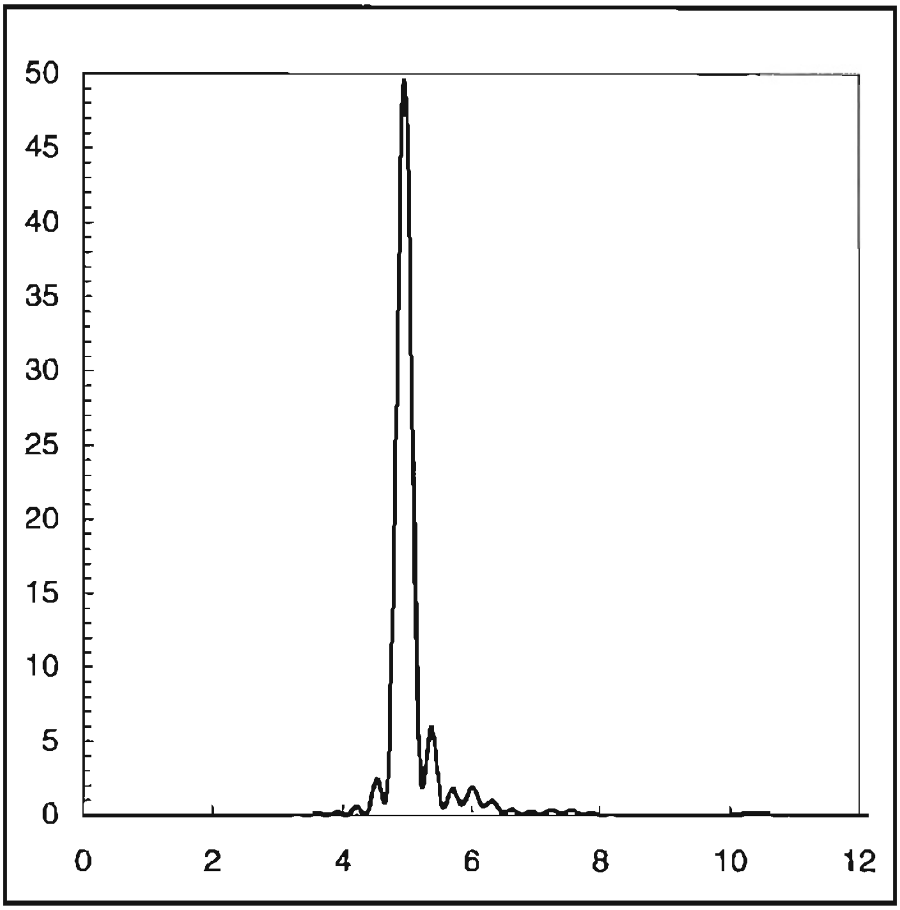


Figure 3. FFT of Normalized Summation of All Mode Shapes in Figure 1

1.2.1.3 Single Input - Single Output (SISO) Auto-Regressive (AR) and Auto-Regressive Moving Average (ARMA) Models

Unlike AR or ARMA models, FFT/PSD methods usually do not determine all modes, and curve-fitting becomes mathematically very complex and require very good initial guesses.

Both AR and ARMA models are system identification methods based upon time difference equations. An AR model models any type of free response, and an ARMA model models any type of forced response. Both models are strictly time domain based unlike FFT/PSD being frequency domain based. These methods have improved capabilities over the frequency based methods because they offer potential advantages when attempting to identify a system with closely spaced modes and multiple modes from multiple outputs (Pinkleman, Batill, and Kehoe, 1995).

ARMA models have been used more extensively to model measured flutter test data or simulated data (Ljung, 1987), while AR models have not. Both the AR and ARMA models (time difference equations) can be coupled with a method to determine the AR and MA coefficients which are usually not known. The AR coefficients are only required to identify the system's modal parameters. The AR coefficients can be determined either by using a transfer function method (Walker & Gupta, 1984; Roy & Walker, 1985), a Recursive Maximum Likelihood (RML) Method (Torii and Matsuzak, 1997; Cooper, 1990), some type of Least Squares Method (Pak and Friedman, 1992; Cooper, 1990; Pinkleman and Batill, 1995), or even a General Instrumental Variables (IV) Method (Cooper, 1990). All of these methods have been proven to be very effective and feasible.

The RML method statistically produces the best estimates for sub-critical flutter points, but this method takes five times the number of calculations as the Least Square methods (Cooper, 1990). Also, the RML method has convergence problems with lightly damped systems. This has only been recommended for sub-critical flutter points. The General IV method performs well in the off-line estimation form. However, it requires twice as many calculations as the Least Squares. The Least Squares methods have been proven to require smaller data samples and are simpler to apply when compared to all others (Cooper, 1990; Pinkleman and Batill, 1995). Due to the negative aspects of the RML and IV methods such as the number of calculations and the convergence problems, these two methods will not be considered for this research.

1.2.2 Flutter Margin and Stability Parameter

Two other methods used to predict flutter boundaries, but not based upon plotting the damping coefficient against a dynamic pressure, are the Flutter Margin and Stability Parameter Estimation Methods. These two methods are based upon plotting the "flutter margin" parameter or the "stability" parameter against the dynamic pressure.

The first method is the Flutter Margin Method (Zimmermann and Wiessenburger, 1964; Applicaation: Hammond & Doggett, 1975). The Flutter Margin is a stability criterion which is based strictly on Routh's Stability criteria applied to the equations of motion from a simple bending/torsion model (two degrees of freedom) with no structural damping. This results in an equation for the flutter margin which is a function of frequency and decay rates (or damping coefficient). Once the flutter margin has been determined from the flutter test results at several dynamic pressures, it can be plotted against the dynamic pressure and is said to be quadratic due to the nature of the characteristic equation from the equation of motion. A quadratic curve can be applied to predict the flutter boundary using the results at very low sub-critical speeds as much as 50% lower than the flutter onset speed.

The problem with this method is that it only involves two degrees of freedom because it is based upon a simple two degree of freedom equation of motion. Therefore, more than three modes cannot be accurately determined. The author even states that for one mode, the method has inaccuracies due to the quadratic curve fitting. This method was further improved to three modes (Price and Lee, 1993), however, the same problems in inaccuracies occurred especially when structural damping was inherent within the data. Because of these negative aspects, this method is not feasible for this research.

Another method, similar to the Flutter Margin, except that it was based upon Jury's Stability Criteria (Jury, 1982), is called the Stability Parameter Method (Torii and Matsuzaki, 1992). This method uses an ARMA model and applies Jury's Stability Criteria to the characteristic polynomial of the difference equation from the ARMA model. The stability parameter becomes a function of the AR coefficients. The AR coefficients are determined by solving the difference equation using the Maximum Likelihood technique. Once the stability parameter has been determined for several dynamic pressures, the stability parameter is plotted against the dynamic pressure and then quadratic curve fitted. The curve is then extrapolated to the flutter boundary (when the stability parameter is zero). Because of the ARMA model, multiple modes with characteristics of closeness and dominating modes can be analyzed (Torii and Matsuzaki, 1992). Therefore, this method is very feasible for the ASENL data and prides itself in situations of explosive flutter because of the use of the time marching ARMA Model. However, this method requires the knowledge of the modal order in the data set which is not the case in the ASENL data as previously explained. And, since STARS is a program providing simulated results, the program can be ran all the way to the flutter boundary. Therefore, no curve fitting, a cause for more inaccuracies, has to be accomplished. This method could be applied to a normalized summation of the data resulting in a known model order (Figure 4), however, it is believed that a much simpler method can be applied. More will be discussed later on the difference between the normalized summation of each mode shape, where the model order is known, and using each independent mode. Therefore, this method will not be further discussed.

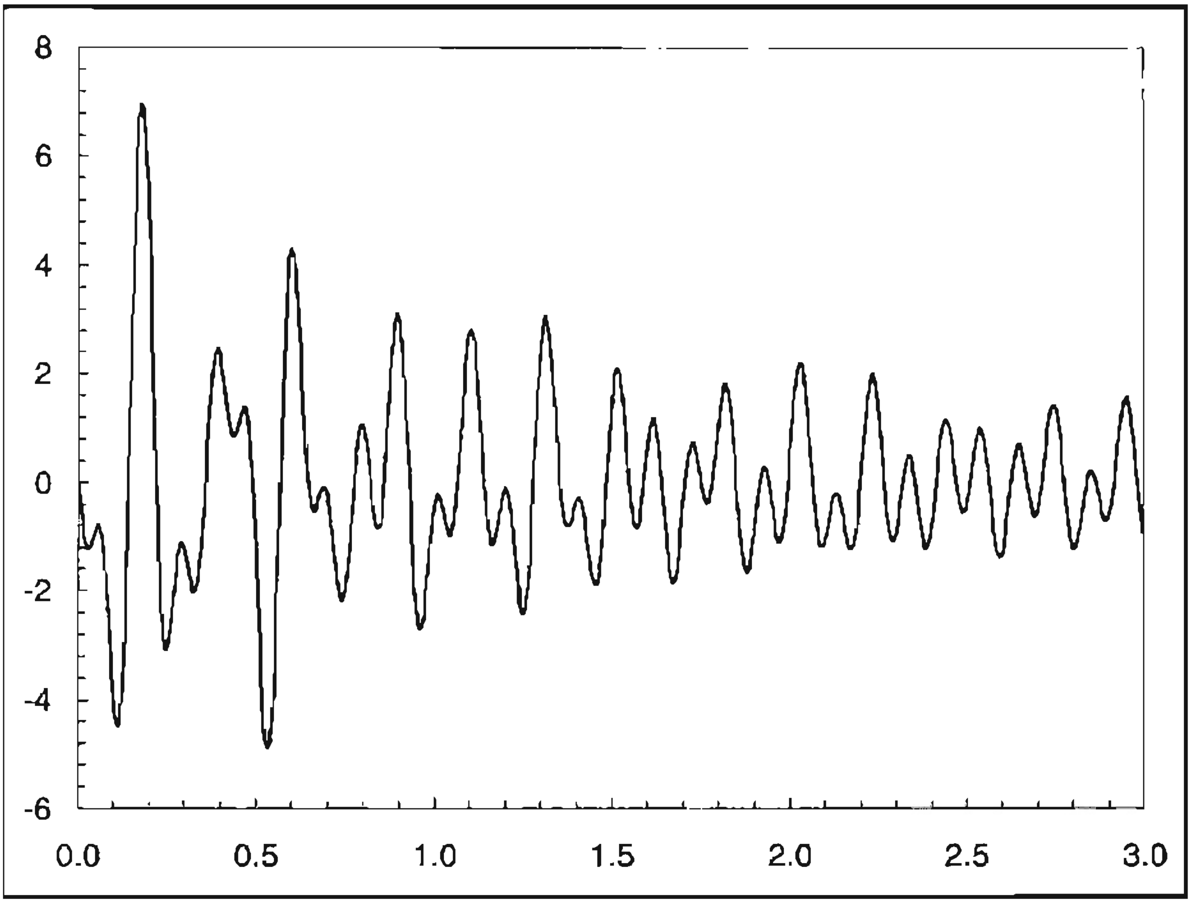


Figure 4. A Typical Normalized Summation of a Nine Mode System

1.2.3 Focus of Research Based Upon Modal Parameter Identification

In summary, the Least Squares Curve Fitting and the Moving Block analysis have been proven to be successful but are very limited to the number of modes (two or lower). They also require substantial amount of data points when compared to AR or

ARMA models, and are strongly dependent upon initial decisions (Pak and Friedman, 1992). Some comparisons will be made to the curve fitting method in this report.

Fast Fourier Transforms requires good knowledge of the modal frequencies and an extensive amount of work in filtering, zeroing, auto-correlation, and data smoothing using hands-on decisions. All of this requires several hours of engineering time. Also, for closely space modes, dominant and multiple modes, FFT/PSD can be very difficult to apply and possibly obtain very inaccurate results. Because of this and because of the example in Figures 2 and 3, FFT/PSD will not be further considered in this research.

AR or ARMA models coupled with a good method, primarily Least Squares, to determine the AR coefficients have been very popular and have been proven to provide accurate results using shorter data samples from single to multiple mode systems even for closely spaced modes. Through this literature review no attempt by any others has used ARMA models for system of five modes or higher.

Due to the more positive aspects of AR and ARMA models, this research will provide further insight upon the application of these models to higher mode systems for one specific pre or post flutter boundary point. Types of Least Squares methods will be the primary focus to identify the AR coefficients from the AR or ARMA models because of the their advantages of using smaller data samples and are simpler algorithms. Once the AR coefficients are identified for one specific flutter point, the modal parameters from these coefficients will be determined. After the damping factor for several pre and post flutter boundary points have been determined, it can plotted against the dynamic pressure for each point thus resulting in flutter boundary prediction.

CHAPTER 2

DEVELOPMENT OF THE MODAL PARAMETER IDENTIFICATION METHODS

2.1 Basis of AR and ARMA Models

A Single Input-Single Output ARMA model can be simplistically represented by an simple time difference equation for a finite number of modes with 2M Auto-Regressive (AR) coefficients, a_i, and W Moving Average (MA) Coefficients, b_i, shown below (Pak and Friedman, 1992).

$$y_k + \sum_{i=1}^{2M} a_i \cdot y_{k-i} = \sum_{i=1}^{W} b_i \cdot u_{k-i} \tag{1}$$

This equation is shown in block form in Figure 5.

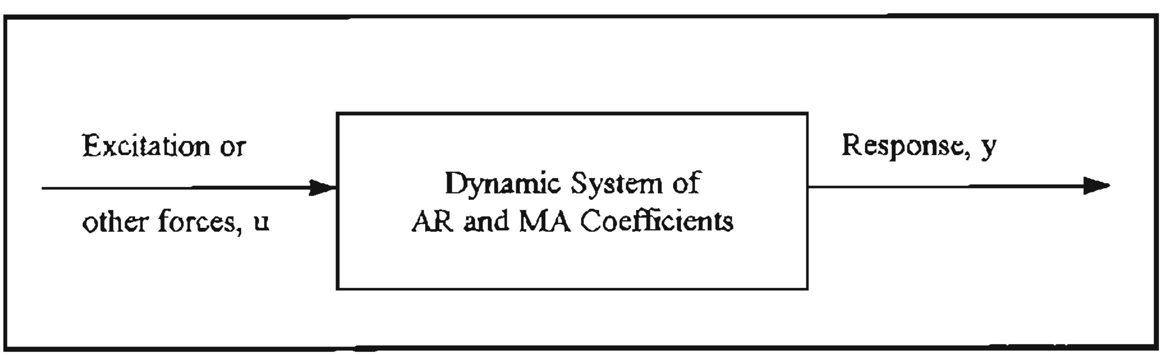


Figure 5. Block Diagram of SISO ARMA Model of Dynamic System

This difference equation can also be written as

$$y_k = -\sum_{i=1}^{2M} a_i \cdot y_{k-i} + \sum_{i=1}^{W} b_i \cdot u_{k-i} \tag{1a}$$

which expresses the response of a dynamic system at any time, y_k, as a function of previous response (regressive) values in the data output, y_{k-i}, and the past data input, u_{k-i}, while knowing a finite set of system AR and MA coefficients, a_i and b_i, respectively. However, in the case of this research these coefficients are not known. Therefore, the measured response and input forces are used to determine these coefficients. As for the AR Model, it is simply Equation 1 except the right hand side is zero.

The order of the model is the number of AR coefficients. The number of AR coefficients should be at least twice the number of modes in the system or 2*M. The number of MA coefficients is usually set to one less than 2*M Auto Regressive coefficients if no prior information is known about the input. If the input is known then Akaike's Information Theoretic Criterion (AIC) can be used to determine the correct number of MA coefficients to include. Since one of the goals of this research was to identify the system modal parameters without knowing the dynamics of the input excitation or forces, this criteria is not useable. However, ASENL data does include an inherent bias or static offset due to the excitation. To account for this bias, instead of using W Moving Average coefficients, as defined earlier, one MA coefficient, b_1, can be used with a fictitious system input, u_1, equal to 1 (Pak and Friedman, 1992). With this, the ARMA model or difference equation reduces to

$$y_k = -a_1 y_{k-1} - a_2 y_{k-2} - \dots - a_{2M} y_{k-2M} + b_1 \tag{2}$$

The bias in the system response is accounted for, but will never be determined due to its value being unimportant in predicting modal parameters. The only thing that is important about Equation 2 is the AR coefficients (a_i). These coefficients are used to determine directly the system modal parameters for a finite number of modes. How these AR coefficients are determined depends upon a type of parameter estimation technique. Section 2.2 discusses three methods that will be compared with each other to determined which method is best to use for this research.

2.2 Techniques To Determine Auto-Regressive (AR) Coefficients

Section 1.2.1.3 stated that Least Squares techniques for determining the AR coefficients required less data samples than all other methods discussed and were easier to implement. The next few sections will focus on the algorithms of the following three types of Least Squares techniques which will be used to determine the AR coefficients.

1. Method of Overdetermined Set of Simultaneous Equations (MOSE)
2. On-Line Least Squares (ON-LS)
3. On-Line Double Least Squares (ON-DLS)

These three methods of Least Square are methods that are very popular and are most suitable for the ASENL data. These three methods will be compared between each other that best determines the AR coefficients from either the AR or ARMA Model (the time difference equations).

2.2.1 Method of Overdetermined Set of Simultaneous Equations (MOSE)

Batill, Pinkleman, and Kehoe proposed to determine the AR coefficients by writing a system of overdetermined set of simultaneous linear algebraic difference equations for the entire data set of N points.

$$\begin{aligned} y_k &= -a_1 y_{k-1} - a_2 y_{k-2} - \dots - a_{2M} y_{k-2M} + b_1 \\ y_{k+1} &= -a_1 y_k - a_2 y_{k-1} - \dots - a_{2M} y_{k-2M+1} + b_1 \\ &\;\;\vdots \\ y_N &= -a_1 y_{N-1} - a_2 y_{N-2} - \dots - a_{2M} y_{N-2M} + b_1 \end{aligned} \tag{3}$$

Usually for a linear solvable set of equations the number of equations should equal the number of unknowns. Here, the unknowns are 2*M AR coefficients and one MA coefficient, therefore making 2*M+1 equations. However, Pinkleman, Batill, and Kehoe suggested to overdetermine the number equations, as shown above, to obtain accurate results in determining the AR coefficients.

The above overdetermined set of simultaneous equations can be written in a linear matrix form being

$$\{\psi\} = [\Phi]\{\Theta\} \tag{4}$$

where

$$\{\Theta\}^T = [-a_1 \;\; -a_2 \;\; -a_3 \;\; \dots \;\; -a_{2M} \;\; b_1] \tag{5}$$

and

$$\{\psi\}^T = [y_k \;\; y_{k+1} \;\; y_{k+2} \;\; \dots \;\; y_N] \tag{6}$$

$$[\Phi] = \begin{bmatrix} y_{k-1} & y_{k-2} & y_{k-3} & \cdot & \cdot & y_{k-2 \cdot M} & 1 \\ y_k & y_{k-1} & y_{k-2} & \cdot & \cdot & y_{k-2M+1} & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ y_{N-1} & y_{N-2} & y_{N-3} & \cdot & \cdot & y_{N-2M} & 1 \end{bmatrix} \tag{7}$$

Again, if $[\Phi]$ was a square matrix, then $\{\Theta\}$ could simple be determined from

$$\{\Theta\} = [\Phi]^{-1}\{\psi\}.$$

However, when there are more equations than unknowns, then the system is said to be overdetermined. If closeness between the right and left hand sides of Equation 4 is defined in the least squares sense, then the overdetermined linear problem reduces to a solvable linear system called the linear least squares problem. Therefore, $\{\Theta\}$ can be solved from (Press, 1992; Pinkleman and Batill, 1995)

$$\{\Theta\} = ([\Phi]^T[\Phi])^{-1} [\Phi]^T \{\psi\} \tag{8}$$

This method becomes the Method of Overdetermined Set of Equations (MOSE) or an Off-Line Least Squares problem of determining the unknown coefficients. This is considered an off-line method because the entire data set from k to N points, resulting in N-k equations is used to determine the AR coefficients. Its solution is the vector, $\{\Theta\}$, of 2*M AR coefficients and one MA coefficients.

As for the initial starting k point for this method, from looking at the matrix $[\Phi]$ at time k = 1, the first row of the matrix becomes

$$[y_0 \;\; y_{-1} \;\; y_{-2} \;\; \ldots \;\; y_{1-2M} \;\; 1]$$

where the data does not exist for these points. Therefore, the initial starting point for this method must be at k = 2M+1 points. When using ASENL data, the initial starting point cannot start until the transient excitation is complete. k must be equal to 2M+1+z where z is the last point of the excitation which is a known parameter.

Instead of using the entire data set, this method can be used to sequence through the data by determining a vector of AR coefficients first using k and k+1 set of equations to determine $\{\Theta\}$, then incrementing by one and determine a new vector of AR coefficients. This is accomplished until N points has been reached or regressive convergence of the AR coefficients can be observed. Once regressive convergence has occurred there is no reason to continue sequencing through the data set, and at this point the AR coefficients are said to be correct if the model order is correct (Pinkleman, Batill, & Kehoe, 1995). However, since the model order is not really defined for each mode shape, improvement to this method must be made. This is further discussed in Section 2.4.

2.2.2 On-Line Least Squares (ON-LS)

On-Line methods are methods primarily used with non-stationary data, when the damping and frequency of a mode changes with time. On-Line methods march through the data thus determining a new vector of AR coefficients at each data point using a "corrector" type of equation instead of an overdetermined set of equations. On-Line methods also allow forgetting of the data, where Off-Line methods or MOSE does not. The Off-Line methods use every past data point in a single matrix while On-Line method updates a new vector of data points at each time marching step. Since the ASENL data is considered stationary, the forgetting factor does not have to be used, therefore, the forgetting factor can be set to one (Cooper, 1990).

The development of the On-Line Least Squares problem for this research begins with Equation 2 of the ARMA model.

$$y_k = -a_1 y_{k-1} - a_2 y_{k-2} - \ldots - a_{2M} y_{k-2M} + b_1 \tag{2}$$

or in matrix form

$$y_k = \{\Theta\}^T{}_k\{\Phi\}_k \tag{9}$$

where $\{\Phi\}_k$ is the data vector at time k (i.e. $\{\Phi\}_k = \{y_{k-1} \quad y_{k-2} \quad \ldots\ldots \quad y_{k-2M}\}$). As mentioned, On-Line methods are "corrector" type methods so the AR coefficient vector is determined as follows which is written for stationary data thus no forgetting (Cooper, 1990)

$$\{\Theta\}_k = \{\Theta\}_{k-1} + [P]_{k-1}\{\psi\}_k\left[\{\psi\}_k{}^T[P]_{k-1}\{\psi\}_k + 1\right]^{-1}\left[\{\psi\}_k{}^T\{\Theta\}_{k-1} - y_k\right] \tag{10}$$

where:

$$[P]_k = [P]_{k-1} - [P]_{k-1}\{\psi\}_k\left[\{\psi\}_k{}^T[P]_{k-1}\{\psi\}_k + 1\right]^{-1}\{\psi\}_k{}^T[P]_{k-1} \tag{11}$$

and

$$\{\psi\}_k{}^T = [\, y_{k-1} \quad y_{k-2} \quad \ldots\ldots \quad y_{k-2M+1} \quad 1\,] \tag{12}$$

The data sequencing must begin at k=2M+1+z for the reasons provided earlier, and everything is known at this point except the initial conditions for $[P]_{k-1}$ and $\{\Theta\}_{k-1}$. The initial condition for $[P]_{2M+z}$ ($k = 2*M+1+z$), is $\alpha*[I]$, where $[I]$ is the identity matrix and α is a large number. The larger α is the quicker the convergence while marching through the data. Here, α will be set to 10^{30} (Cooper, 1990; Pak and Friedman, 1992). The initial condition for $\{\Theta\}_{2M+z}$ can be anything from zeros to the full solution from all data points (Cooper, 1990; Pak and Friedman, 1992). For this research $\{\Theta\}_{2M+z}$ will be a vector of zero's, because this initial condition had very little effect upon the final results.

2.2.3 On-Line Double Least Squares (ON-DLS)

On-Line Double Least Squares was developed to reduce the bias from noise corruption in flight test data since damping coefficients are very sensitive to noise (Cooper, 1990). This method averages two solutions in which the damping estimated has a positive and negative bias and the bias is hoped to cancel out. This method really should only be applied to noisy flight test data. Even though the ASENL data is "noise-free", this method will be applied in this research to show some comparisons with the previous two methods.

Mathematically, the only difference between On-Line Least Squares and On-Line Double Least Squares is the vector $\{\psi\}_k$ which is represented by two vectors, $\{\alpha\}_k$ and $\{\beta\}_k$ being

$$\{\alpha\}_k^T = [\, y_k + y_{k-1} \;\; y_{k-1} + y_{k-2} \;\; \ldots \;\; y_{k-2M} + y_{k-2M+1} \;\; 2\,] \tag{13}$$

and

$$\{\beta\}_k^T = [\, y_{k-1} \;\; y_{k-2} \;\; \ldots \;\; y_{k-2M+1} \;\; 1\,] \tag{14}$$

where the equation for $\{\Theta\}_k$ and $[P]_k$ are now

$$\{\Theta\}_k = \{\Theta\}_{k-1} + [P]_{k-1}\{\alpha\}_k\,[\{\beta\}_k^T[P]_{k-1}\{\alpha\}_k + 1]^{-1}[\{\beta\}_k^T\{\Theta\}_{k-1} - y_k]$$

$$[P]_k = [P]_{k-1} - [P]_{k-1}\{\alpha\}_k\,[\{\beta\}_k^T[P]_{k-1}\{\alpha\}_k + 1]^{-1}\{\beta\}_k^T[P]_{k-1}$$

This method is said to provide more accurate results than the On-Line Least Squares method, but does take more calculations (Cooper, 1990).

2.3 Extraction of Modal Parameters from AR Coefficients

With $\{\Theta\}^T = [-a_1 \;\; -a_2 \;\; -a_3 \;\; \ldots \;\; -a_{2M} \;\; b_1]$ now determined from any of the three methods discussed, the modal parameters can be determined using one of two following methods.

The first method involves finding the roots from the following characteristic 2*M order polynomial which represents the AR part of the ARMA model.

$$\lambda^{2M} + a_1\lambda^{2M-1} + a_2\lambda^{2M-2} + \ldots\ldots + a_{2M} \qquad (15)$$

Only the complex roots, which determine modal parameters, to this equation becomes the modal parameters of the system. All other type of roots, called calculated roots, are discarded.

Many simple root finding methods exist for polynomials such as Bisection, False Position, Newton's, and Bairstow's Method which are only a few. The common factor between all of these methods is an initial guess is required. The initial guess usually strongly affects the results as in Bisection and False Position (Gerald and Wheatly, 1994). Many of these methods can be excluded due to the fact that they due not work well with complex numbers. Newton's method can be used, but it does require a complex initial guess and special complex arithmetic. Only Bairstow's Method for Quadratic Factoring is best when working with complex numbers because special complex arithmetic is not required. The negative aspect to all of these methods is that all of them have difficulty with repeated roots being identical modes (Press, 1992).

The second method for determining the modal parameters from the AR coefficients involves finding eigenvalues from the following matrix (Pak and Friedman, 1992).

$$\begin{bmatrix} -a_1 & 1 & 0 & \dots & 0 \\ -a_2 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ -a_{2M-1} & 0 & 0 & 0 & 1 \\ -a_{2M} & 0 & 0 & 0 & 0 \end{bmatrix} \tag{16}$$

This general real matrix is determined from the state-space form of the difference equation (Eq. 2) (Pak and Freidman, 1962). When finding the eigenvalues of this matrix, it usually requires one more step than finding the roots of a polynomial when the following equation is applied.

$$\det(\text{Matrix} - \lambda I) \tag{17}$$

The results of this equation is a polynomial similar to Equation 15. Therefore, why introduce more calculations which could cause more round off error thus affecting the final results. Both the Bairstow's method and Pack & Friedman method will be examined for final application.

Once the roots of the polynomial or eigenvalues, usually several pairs of complex conjugates $(u \pm iv)_j$, have been determined from either method, the modal parameters can be determined using the equations below.

$$\omega_{Dj} = \frac{1}{h} \operatorname{atan}\left(\frac{-u_j}{v_j}\right). \tag{18}$$

$$\zeta_j = \frac{\sigma_j}{\sqrt{\omega_{Dj}^{\ 2} + \sigma_j^{\ 2}}} \tag{19}$$

$$\text{where } \sigma_j = -\frac{1}{2h} \ln(u_j^2 + v_j^2) \tag{20}$$

2.4 Noise in Input Data and Determination of Model Order

Usually in most cases of flutter data, two factors are inherent, noise in the data and the modal order is not known. With the ASENL data being "noise-free", this reduces the complexity of the method, and the method also require more data points. Therefore, this will not be a problem and not addressed any further.

Theoretically, each individual mode may contain any or all other modes. Because of this, the finite modal order is not known, therefore, two methods were approached. The first was to normalize all modes then sum them together (i.e. Figure 4), then apply each method discussed with the finite known model order. Several days of analysis were accomplished, and the final conclusion was that several points were required to obtain all system modes. Therefore, the more modes that exist or trying to identify using a single data stream the more points required for regressive convergence, especially for large mode systems such as the GHV model. Chapter 3 will provide more reasons with normalizing and summing each mode shape is not feasible.

The second method uses model overspecification to obtain the modal parameters desired (Pinkleman and Batill, 1995). Model overspecification is not required if the model order is known (Cooper, 1990; Torii & Matsuzaki, 1997). For example, if one known mode exists in Mode 1 then the model order may be overspecified with an order of two or higher. The problem with this is that both additional unwanted parameters, called calculated parameters, and the actual system modal parameters are obtained. One method of avoiding the calculated modes is to input the response data in reverse order called the Reduced Backward Method (RBM) (Pinkleman and Batill, 1995). When using the RBM, the stable system parameters are driven unstable and the stable calculated parameters are forced stable. Therefore, the resulting unstable parameters are considered the system parameters then the sign is changed on the damping coefficient making it stable and the calculated parameters are discarded. This method only works with stable systems because unstable systems are stable in the backwards sense in which both the calculated and system modal parameters are driven stable when RBM is applied. Therefore, the unstable system parameters will never be identified among the clutter of the calculated parameters. Recall, one of the desires of this research is to identify unstable modes, therefore RBM cannot be applied for this research.

Another method of only finding system modal parameters when using model order overspecification is to compare the results of two or even three overspecified models of different model order. This idea came from Pinkleman and Batill, when they were showing that accuracy of the damping coefficients was increased for higher overspecified model orders. Similar modes existing between the two results are usually the system modes after eliminating unreasonable calculated modes usually having negative or zero frequencies. This method will be used in this research to make the complete algorithm of a very direct method of determining modal parameters.

Model order overspecification is usually not applied to On-Line methods due to convergence problems (Cooper, 1990). This may only be true if noise is inherent in the data. Model overspecification will be adapted with the on-line methods to see if convergence is achieved.

2.5 Re-Sampling of Input Data

Most response data, if obtained or developed correctly, usually has a sample frequency much greater than the Nyquist frequency by orders of magnitude from 5 to 500. In most of the literature, when these system identification methods were applied, especially to complex high mode systems and highly sample systems, the input data or data trying to be modeled was re-sampled closer to the Nyquist frequency (Cooper, 1992; Pak & Friedman, 1995; and Pinkleman and Batill, 1995). More often, the data was usually re-sampled 2.5 times the Nyquist frequency, which itself is defined as two times the frequency of interest. Thus, the re-sample frequency should be at least five times the frequency of interest. No explanation was every seen on why the original or high sample frequencies from the input were never used. Different re-sampling frequencies will be examined in Chapter 3.

2.6 Complete Algorithms of The Three Modal Parameter Identification Methods

2.6.1 Method of Overdetermined Set of Simultaneous Equations (MOSE)

The complete algorithm of the Method of Overdetermined Set of Simultaneous Equations (MOSE) for any model order is shown in Figure 6. Model overspecification is not included in this algorithm because a study will be done using different model order for each system analyzed.

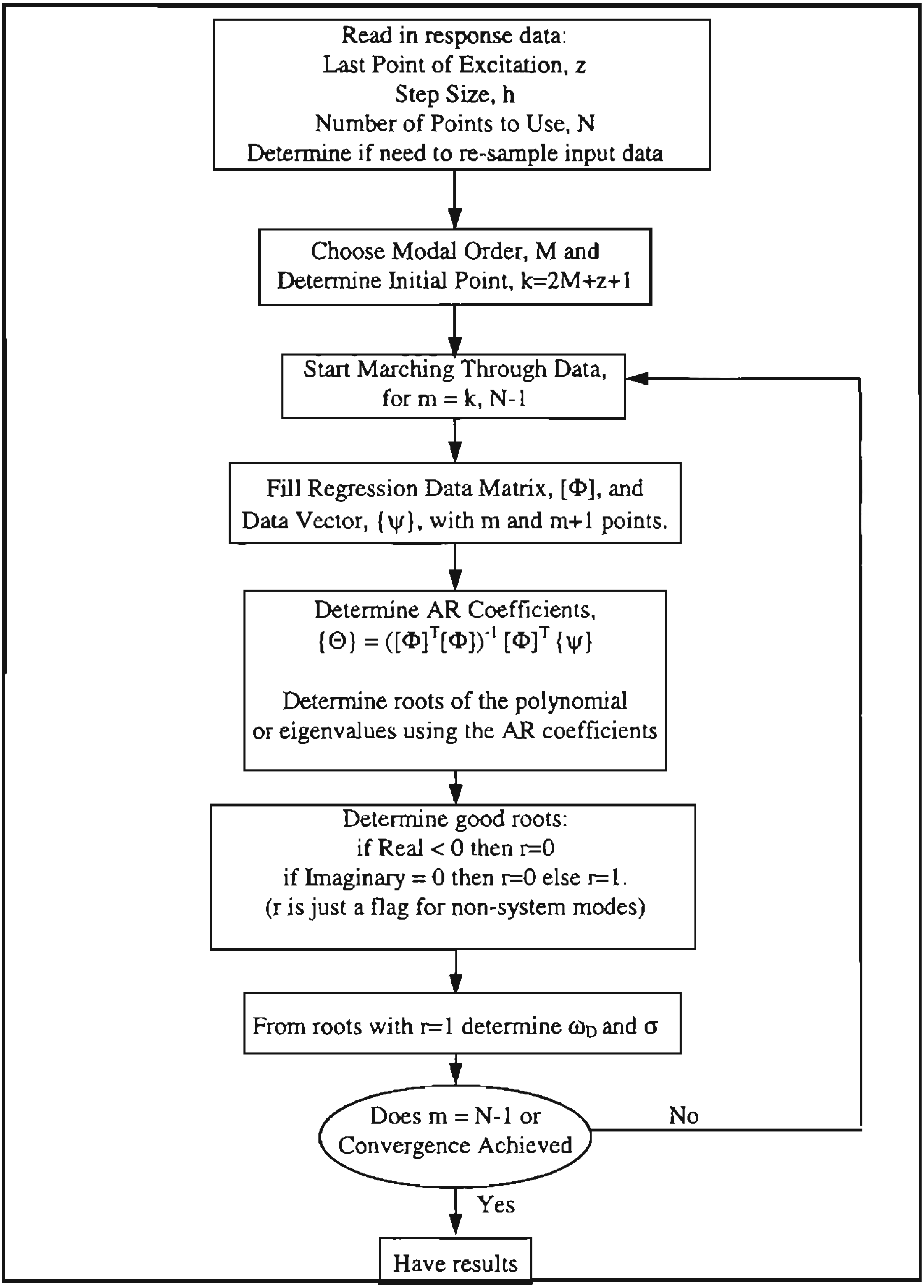


Figure 6. Algorithm for Method of Overdetermined Set of Equations For Any Modal Order (NOT including Model Overspecification).

2.6.2 On-Line Least and Double Least Squares

The complete algorithm, again not including model overspecification, of both On-Line and Double Least Squares is shown in Figure 7. The only difference between double and basic least squares is the determination of $\{\alpha\}_k$ and $\{\beta\}_k$ instead of $\{\psi\}_k$, which was discussed earlier.

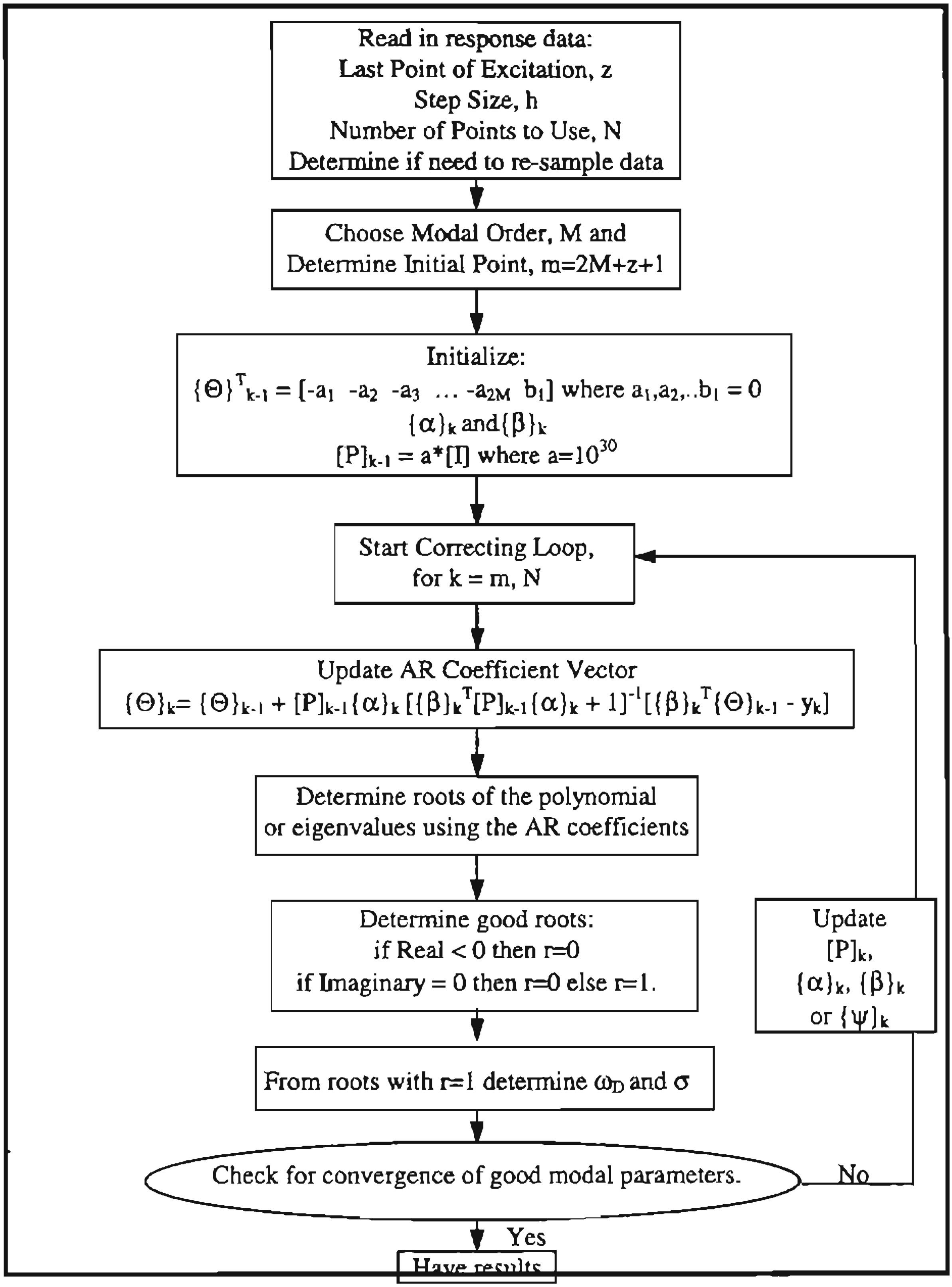


Figure 7. Algorithm of On-Line Least Squares and Double Least Squares For Any Modal Order. (NOT including Model Overspecification)

2.7 How These Three Modal Parameter Identification Methods Will Be Compared

To determine the best of these three complete system identification methods, each method will be compared by using both simulated and actual data sets (ASENL data) at one specific pre or post flutter point as input. For comparison only, MATHCAD v6.0 will be used as a tool to apply each method. A MATHCAD example of each method, which basically follows the algorithms in past two sections, is provided in Appendix A.

When applying all three methods, the entire data set will be used to analyze the characteristics of the regressive convergence and the final results. Once each method begins marching though each data point, starting from k, the modal parameters are stored in a matrix in the order as they are determined. For example, if a model of order three is used with a data stream of 500 points, then the final matrix size for each modal parameter will be a 500-k by M matrix (M is the number of modes in the system). Once the entire data set is used, each column will be plotted for each modal parameter. From this plot the regressive convergence of the model parameters can be analyzed.

The criteria for selecting the best of the three methods will be based upon the following two criteria.

1. Examining the convergence of each modal parameter when marching though the data stream, which will be called regressive convergence. This examination will mainly determine which method requires the least number of points to provide good results. Obviously, the best accuracy for regressive convergence is 0.0% error, and in most references the best accuracies obtained

have been as low as 5.0% (Cooper, 1990 and Pinkleman and Batill, 1992). An accuracy limit for this research will be at least 5% error, however, the best accuracy will try to be achieved. This is only a goal for this research. If this goal is obtained, the results are no worse than past research accomplished by others.

2. Examining the convergence when applying model overspecification will be called mode convergence. This will further justify the number of points required and the accuracy of the modal parameter. In most of these systems being analyzed in the next few sections, the exact number of modes to determine the model order are not known. Therefore, model overspecification has to be applied.

In some of the simpler systems being analyzed, these identified modal parameters will be used with a sinusoidal damped equation and plotted against the inputted ASENL data to determine if they are indeed accurate.

Recall the ASENL data includes both the generalized displacements, q, and velocities, qdot, for each independent mode which may or may not include some or all of the other modes. This is very different than in flight test data where a single data stream includes all modes of interest. The best of the three modal parameter identification methods will be based upon the earlier criteria, and they will determine

1. which ASENL data set to use, the generalize displacements or velocities;
2. whether a normalized summation of all the independent mode shapes will be used since the modal order will be known is better, or to apply the methods to

each individual mode shape, and

3. if re-sampling needs to occur.

CHAPTER 3

RESULTS FROM COMPARING ALL THREE MODAL PARAMETER IDENTIFICATION METHODS USING DIFFERENT SYSTEMS

3.1 Using Simulated Single Mode System.

A very common, method proving system is a simple single mode sinusoidal damped system. This system, being of the form

$$y(t) = C + e^{-\sigma t}\cos(\omega_D t)$$

$$\text{where:} \quad \sigma = 5,\ \omega_D = 30,\ C = 1,$$

was first used in the basic curve-fitting application (Bennett and Desmarais, 1975). A plot of this system's time history is shown in Figure 8. The frequency ratio, F, in this figure, is defined as the sample frequency over the system frequency. This data set will be used by applying all three methods with and without the static offset term (one MA term) in the ARMA model. This will show why an ARMA over an AR model must be used and prove why only one MA coefficient is required.

The input data contains 64 data points with a step size of 0.0098663 seconds, or a sample frequency of 101.36 Hz (636.83 rad/s) giving a frequency ratio, F, of 21.2. The starting data point is $k = 2M+z+1$ or 3 since only one mode ($M = 1$) and no excitation ($z = 0$) exists. The results of all three methods using these ARMA models are shown

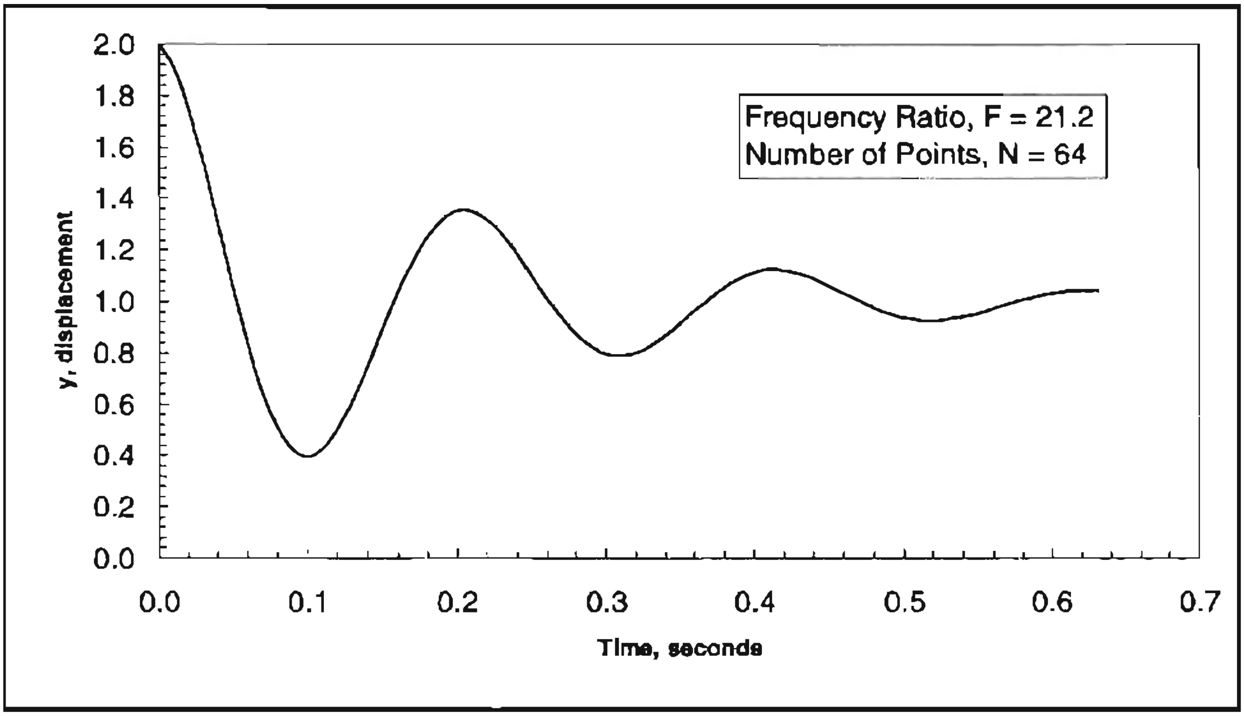


Figure 8. Time History of Simulated Single Mode System

Method	σ (rad/s)	ω_D (rad/s)	Points needed for convergence
LSCFM **	5.0	30.0	64
Method of Overdetermined Equations	5.0	30.0	4
On-Line Least Squares	5.0	30.0	8
On-Line Double Least Squares	5.0	30.0	10
Method of Overspecification*	3.095	6.873	After 63 Points
On-Line Least Squares*	-	-	Method Failed because r = 0
On-Line Double Least Squares*	-	-	Method Failed because r = 0

* AR model (no account for the static offset, C)

** Least Squares Curve Fitting Method (LSCFM) (Bennett and Desmarais, 1975)

Table 1. Modal Parameter Results From Application of Several Methods Using A Simulated Single Mode System

directly from MATHCAD 6.0 in Appendix A and are summarized in Table 1. Table 1 also provides comparisons with the Least Squares Curve Fitting Method (Bennett and Desmarais, 1975).

It is apparent from Table 1, the ARMA model with the static offset term (one MA coefficient) for any of the three system identification methods must be used. The AR model, no MA coefficients, provided inaccurate results or regressive convergence never occurred. The LSCFM results identified the modal parameters, but it required several data points compared to the other methods. The ARMA model with MOSE determined a solution after only 4 points while the other two ARMA model ON-LINE methods required slightly more points.

Re-sampling at a lower frequency ratio had no affect on the accuracy or regressive convergence of this data, however, this is a very simple system. Also, no model overspecification was used because the model order was known.

3.2 Using ASENL Data

3.2.1 Two Mode System (AGARD)

3.2.1.1 Description of Data

This two mode system using the AGARD Wing configuration, which is a standard aeroelastic test case experimentally investigated in the Langley Transonic Dynamics Tunnel, is a result from the ASENL Program in STARS. This geometry is shown in Figure 9. This module applies a transient structural excitation resulting in a response where one of the two modes is unstable. Table 2 provides the properties of the input data for the AGARD data set. This data set contains two independent modes of generalized displacements, q, and velocities, qdot, which is plotted in Figures 10 and 11 for both modes.

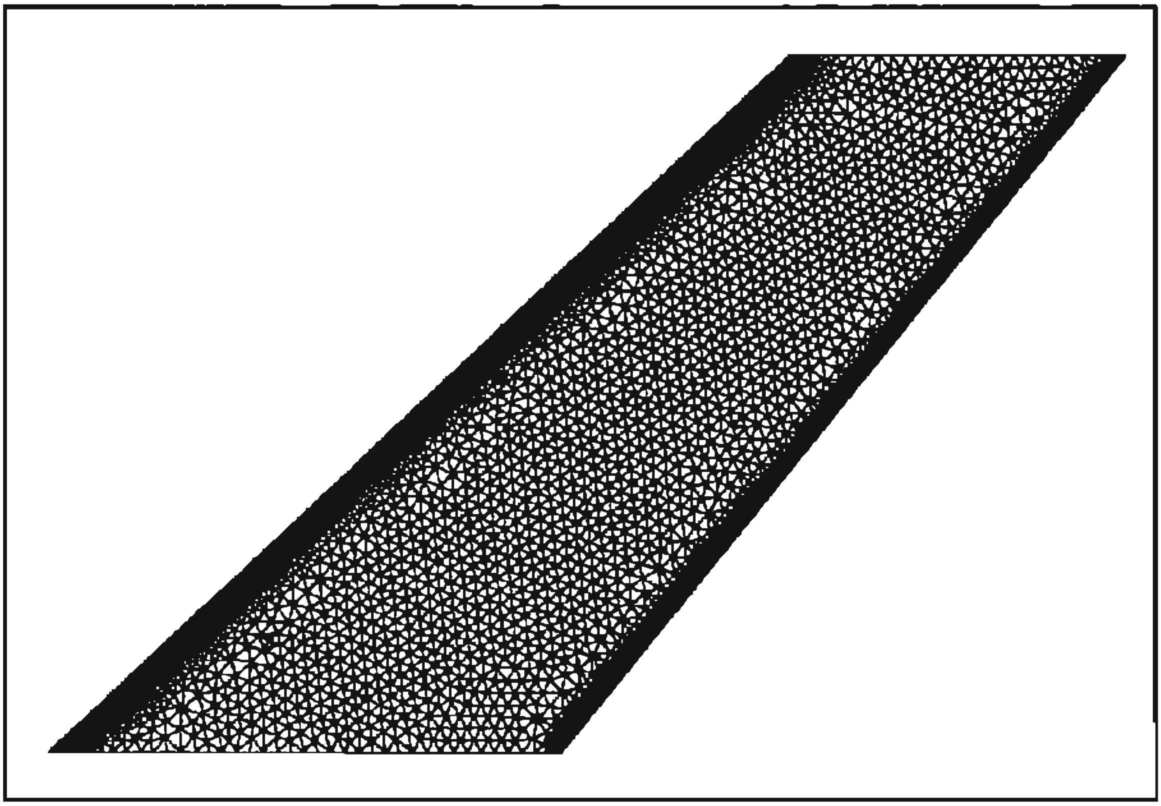


Figure 9. Planform of AGARD Wing Configuration

Item		Value
Number of Total Modes		2
Sample Frequency, ω_s, (Rad/s)		4,829.51
Number of Points, N		1,000
Last Point of Excitation, z		4
Natural Frequencies @ Q = 0, (Rad/s)		
	Mode 1	60.312
	Mode 2	239.798
Frequency Ratio, $F = \omega_s/\omega_n$:		
	Mode 1	80.1
	Mode 2	20.1

Table 2. Properties of Input Response Data From AGARD

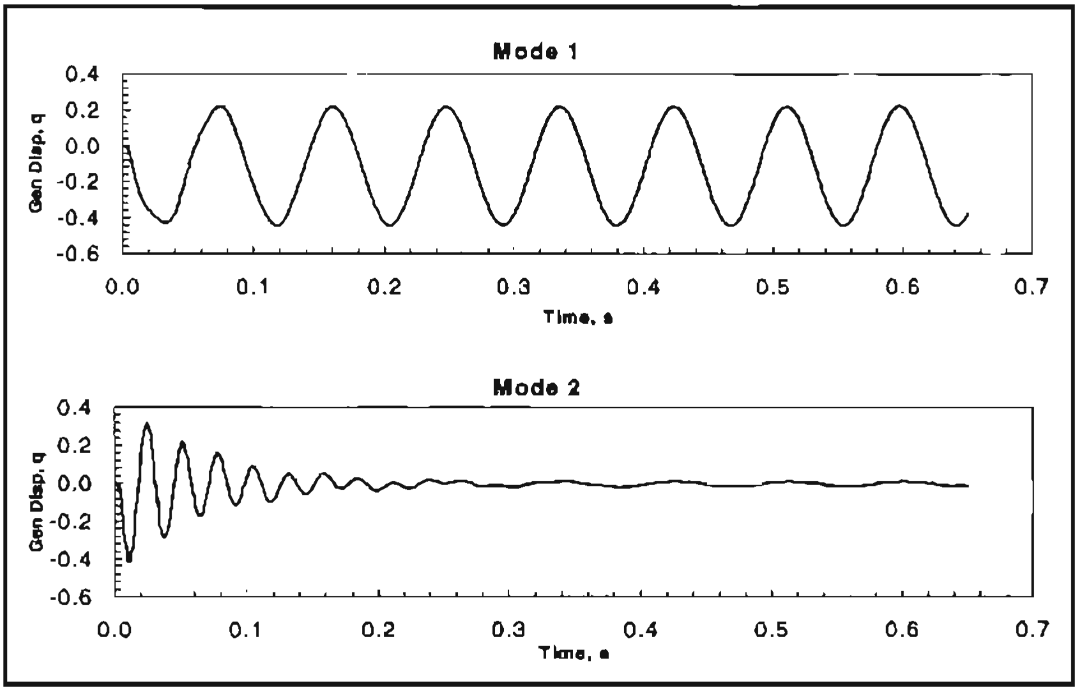


Figure 10. Time Histories of Generalized Displacements For Each Independent Mode Shape From AGARD

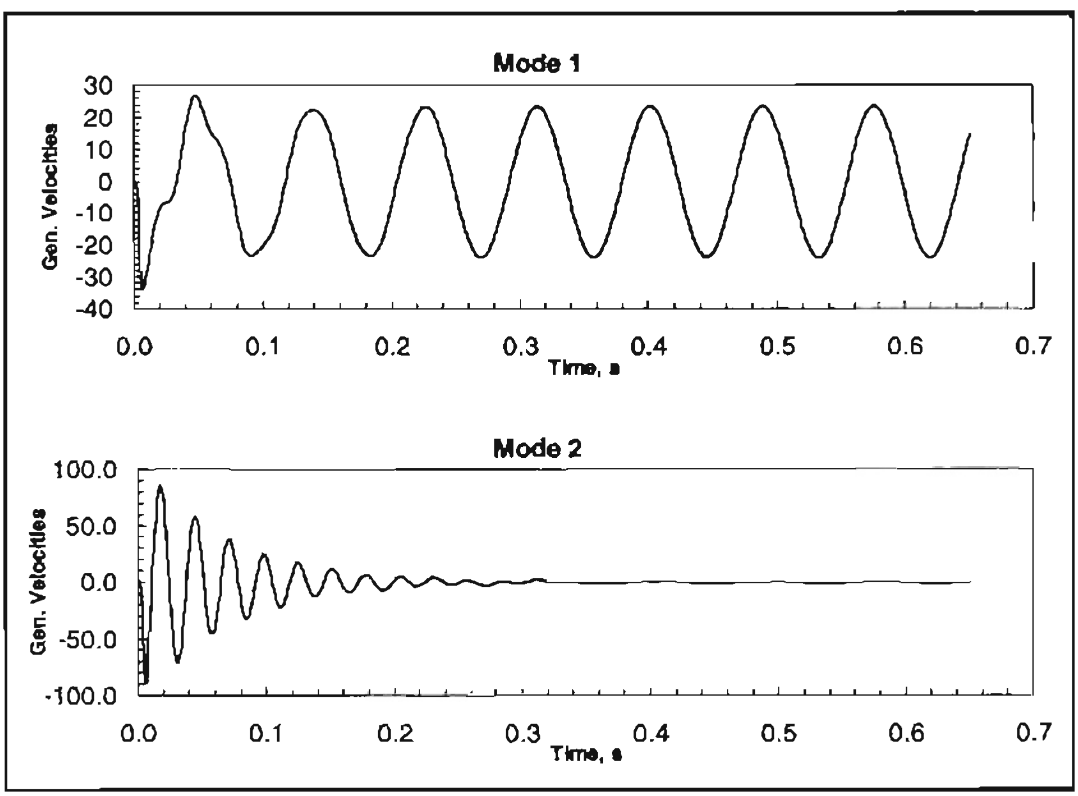


Figure 11. Time Histories of Generalized Velocities For Each Independent Mode Shape From AGARD

3.2.1.2 Results of Re-Sampling the Data and Model Overspecification

When all three methods were applied to both the displacement and the velocity data using higher model orders at the original frequency ratios, F, regressive convergence of the modal parameters was never obtained. This was due to noisy outputs of the modal parameters while regressing through the data stream. This occurred primarily with MOSE. An example of this is shown in Figure 12 being the results from applying MOSE to the generalized displacements of Mode 2 with a model order of six (M=6). The dashed lines represent the 5% error band for regressive convergence.

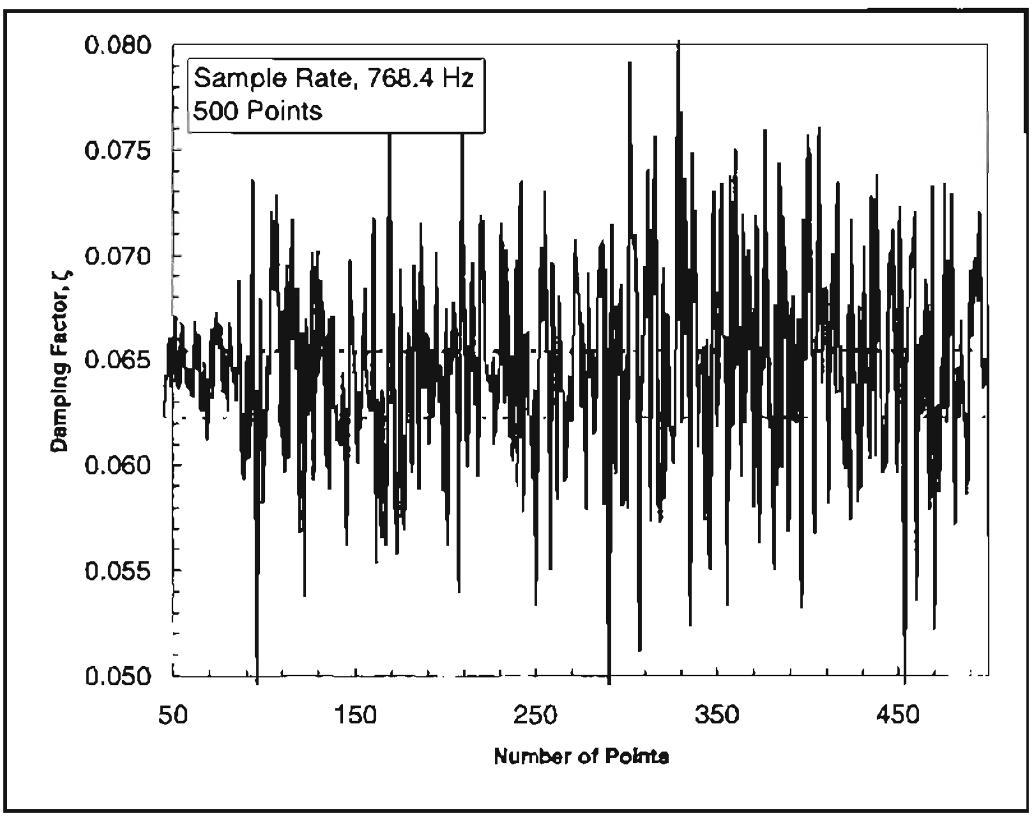


Figure 12. Example of Applying MOSE Using the Original Frequency Ratio (Frequency Ratio, F = 20.1)

This problem can be solved by reducing the sample frequency of the input data as explained in Section 2.5. Recall, the data was recommended to be re-sampled for input five times the frequency of interest or in other words, at a re-sample frequency ratio, F, equal to at least five.

Before a specific re-sample frequency of 5 is chosen as fact, several re-sample frequencies were applied using overspecified models for each of the three modal parameter identification methods. Before these results are shown, the term re-sample factor, n, must be defined. The re-sample factor is a number which divides the original sample frequency which in turns defines the re-sample frequency. For example, if n = 1, the re-sample frequency is equal the original sample frequency given. If n = 4, then the re-sample frequency is four times lower than the original re-sample frequency.

Mode 1, again, of AGARD was used to analyze the effect of re-sampling at lower sample frequencies, or higher re-sample factors. This effect using all three methods for several model orders are shown in Figures 13 through 15. The damped frequency of this mode is not plotted because it was determined with any modal order at almost all re-sample factors. It is only the damping product that had difficulty in regressively converging upon a good result. This is seen in the results provided in Appendix B.

The first observation from Figures 13 through 15 is as the re-sample factor and the model order is increased the convergence of the damping product improves for any method. The best results for all methods were obtained when the re-sample factor was greater than two ($F < 33.6$) for model orders greater than two. A model order of eight provided the best results for any method. Noisy results were apparent at lower re-sample factors especially for n = 1 for the MOSE method.

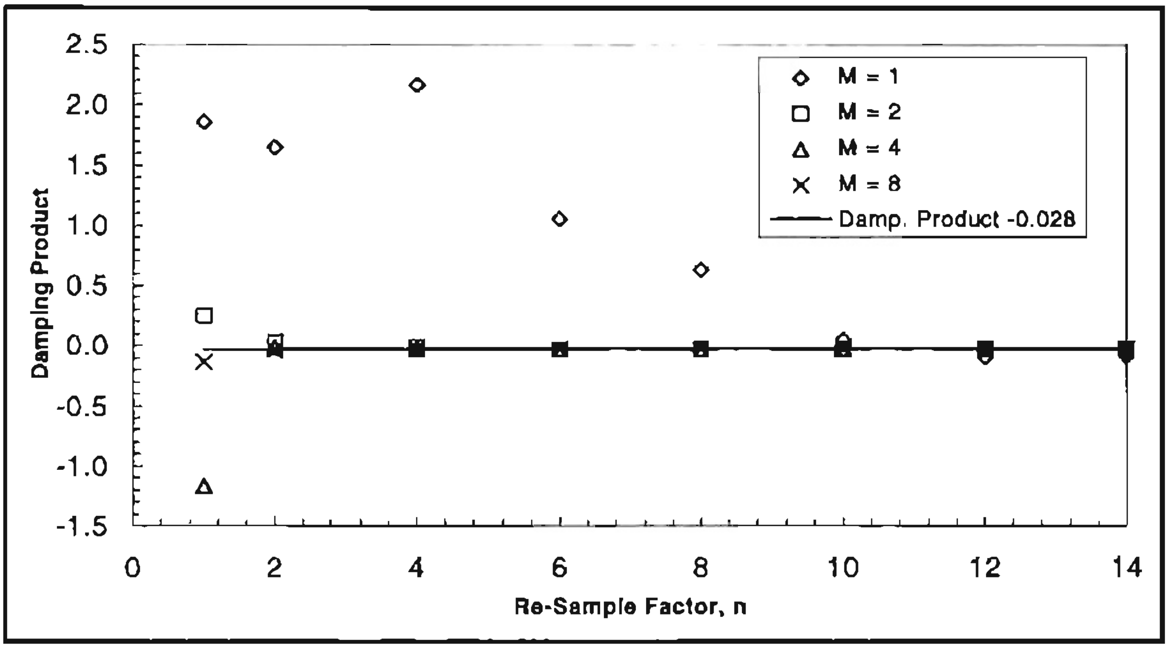


Figure 13. Damping Product, σ, Versus Re-Sample Factor, n, Using MOSE at Various Model Orders Applied to Mode 1 of AGARD

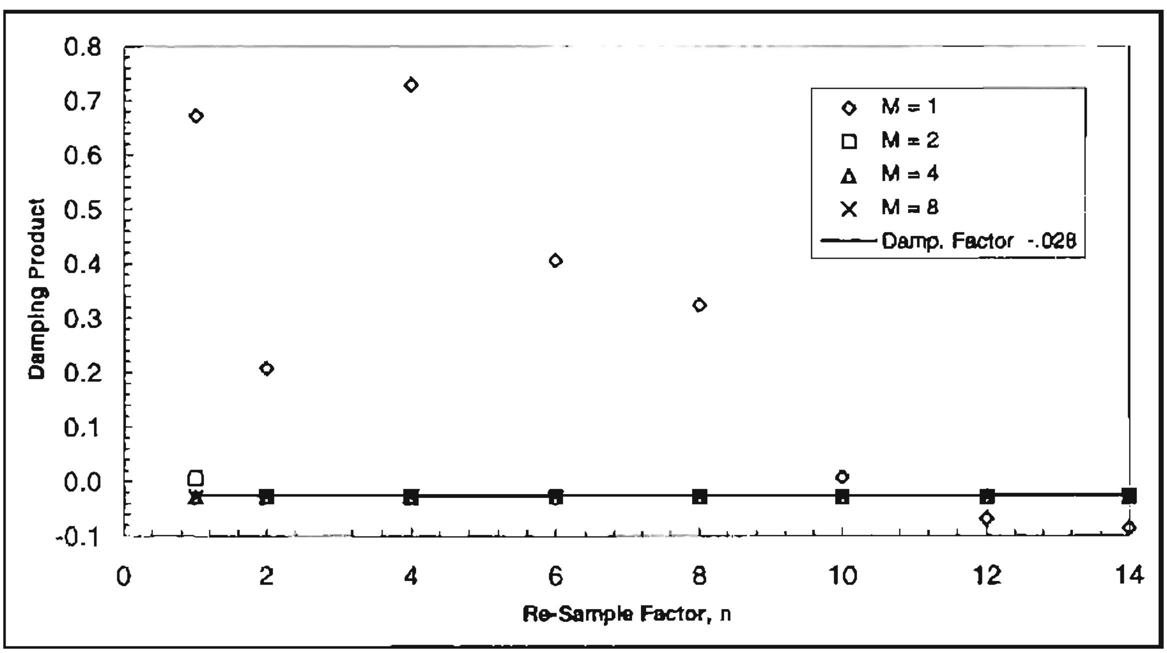


Figure 14. Damping Product, σ, Versus Re-Sample Factor, n, Using On-Line Least Squares at Various Model Orders Applied to Mode 1 of AGARD

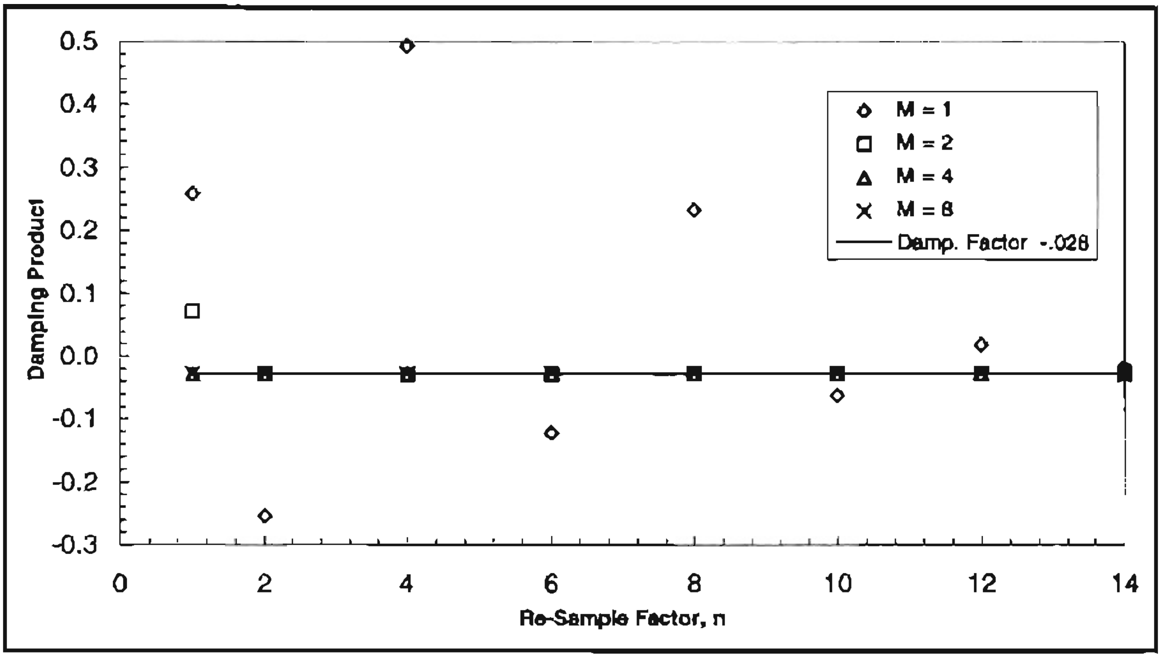


Figure 15. Damping Product, σ, Versus Re-Sample Factor, n, Using On-Line Double Least Squares at Various Model Orders Applied to Mode 1 of AGARD

It has been shown the convergence or say the accuracy of the damping product was improved with a greater re-sample factor and higher model orders. What about the advantage between each method? Figure 16 provides the damping product plotted against the model order at a specific re-sample factor of n = 8 (F = 8.4) as a direct comparison between all three methods. This figure shows very good results for model orders two or greater for all methods, but here no advantage between any method is really apparent.

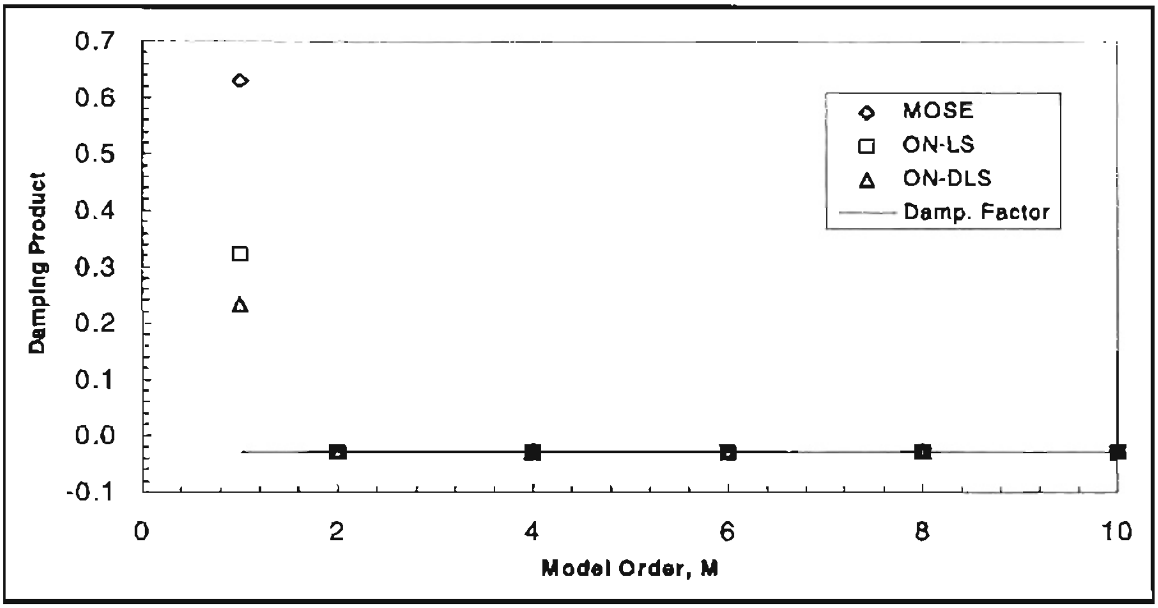


Figure 16. Damping Product, σ, Versus Model Order, M, Comparing All Three Methods at a Re-Sample Factor of 8 Using Mode 1 of AGARD

Now that the accuracy of the damping product has been shown to increase with increasing re-sample factor and model order, how many points does it take for the regressive convergence of the results shown in Figures 13 though 16? Figures 17 through 19 provides the number of points to converge upon 5% of the damping product provided in Figures 13 through 15. Results for a re-sample factor of n = 1 are not shown because of the inaccurate results of the damping product shown in Figures 13 through 15. Also, for these figures when points are shown plotted at 1,000, this means that the damping product after 1,000 points did not regressively converge upon 5% of the damping product primarily due to noisy results.

In terms of number of points for regressive convergence, re-sampling the data for input had the greatest effect on the MOSE method. With n = 2, regressive convergence

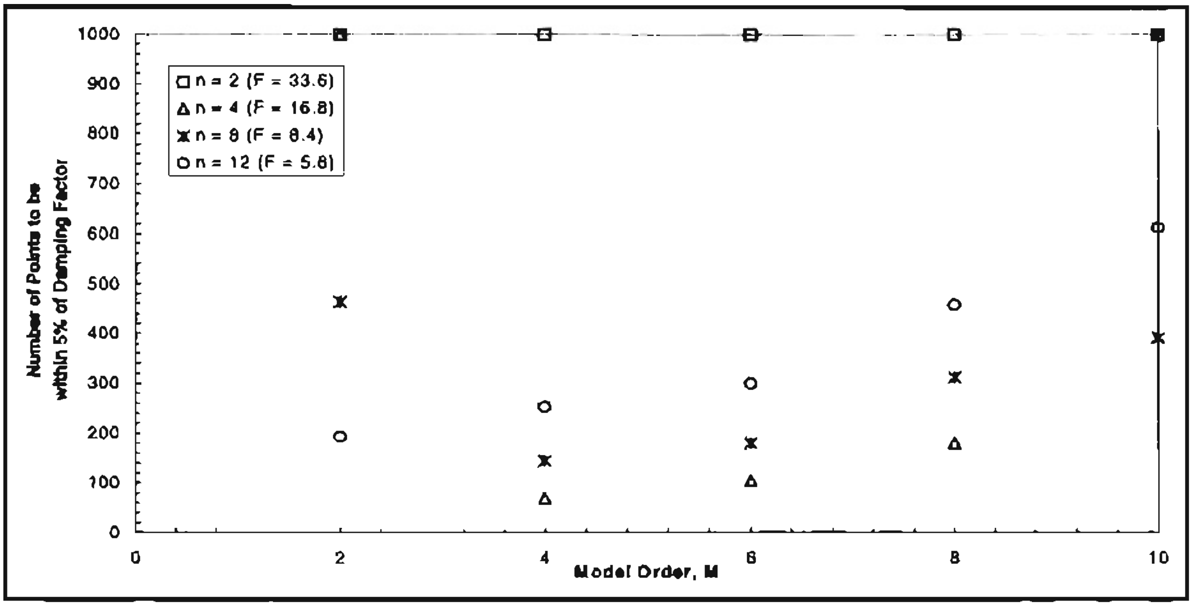


Figure 17. Number of Points to be Within 5% of the Damping Product Versus Model Order Using MOSE for Various Re-Sample Factors, n, Applied to Mode 1 of AGARD

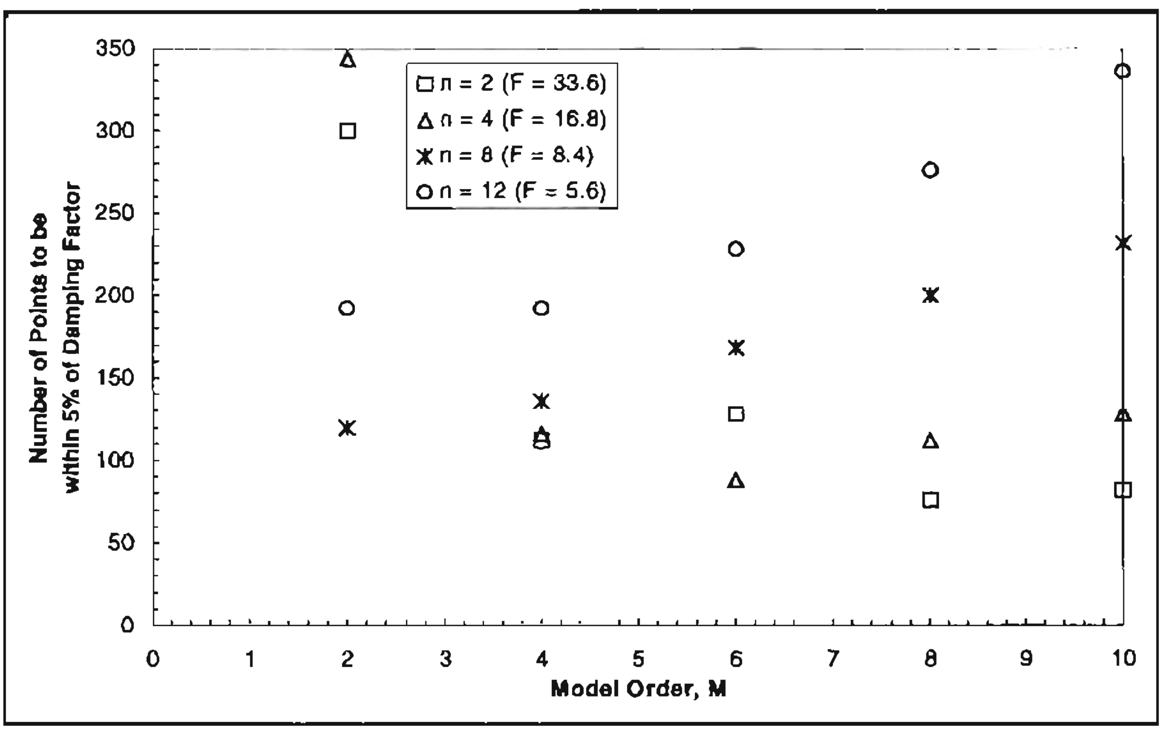


Figure 18. Number of Points to be Within 5% of the Damping Product Versus the Model Orders Using ON-LS for Various Re-Sample Factors Applied to Mode 1 of AGARD

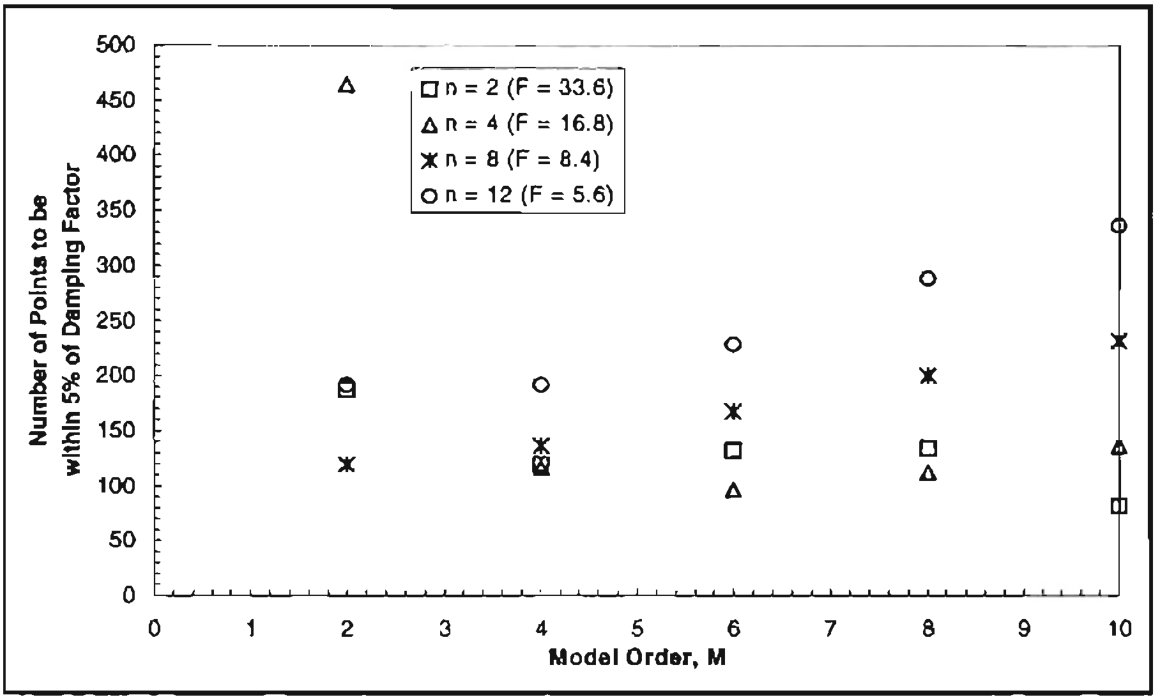


Figure 19. Number of Points to be Within 5% of the Damping Product Versus The Model Orders Using ON-DLS for Various Re-Sample Factors Applied to Mode 1 of AGARD

was never obtained, however, with an increase in n from 2 to 4, the number of points for regressive convergence were much lower.

Re-sampling the data at a lower frequency, higher re-sample factor (n > 6 or F < 12.6), only causes more total points to be used from the input response data due to larger step size. This is why re-sampling the data at a frequency ratio of five did not provide the best results in terms of number of points for regressive convergence. Therefore, a limit on the re-sample factor must be specified.

During this research, while analyzing several highly sampled systems at given frequency ratios, F, from 5 to 500, typical results as shown in Figure 20 occurred. This figure provides what occurs to the damping factor and the number of points for regressive convergence while increasing the re-sample factor, n, or decreasing the new frequency ratio, F. Above a certain new frequency ratio, usually F > 25, the accuracy of the damping factor decreased due to noisy results (the step size was too small for the time difference equation), and thus no regressive convergence. Below a certain new frequency ratio, almost always (F < 5), the accuracy of the damping factor decreased due to the method is aliasing the input data. Therefore, this analysis concluded that for given frequency ratios below 225, good results were obtained when the re-sample factor, n, was eight or less (giving new frequency ratios of 28 and less, but never less than 5), and for given frequency ratios greater than 225, the new frequency ratio was set at 12 instead of 5. These set limits are provided to obtain the data in less amount of points and still maintain good accuracy.

For re-sample factors of 4 and 8 the different methods are compared in Figures 21 and 22 to determine which of the three methods are better. From these two plots and plots of the damping product previously, the ON-LINE methods produced the best results in the least amount of points for all model order. The MOSE method for a model order of 2 was poor, however, for model orders near 4 the results between all methods were similar. When the model order was increased beyond 6, the MOSE method required the most points while the ON-LINE methods required almost the exact number of points between model orders. All methods provided very similar results in terms of the damping product for these high model orders. This is why the damping product is not plotted.

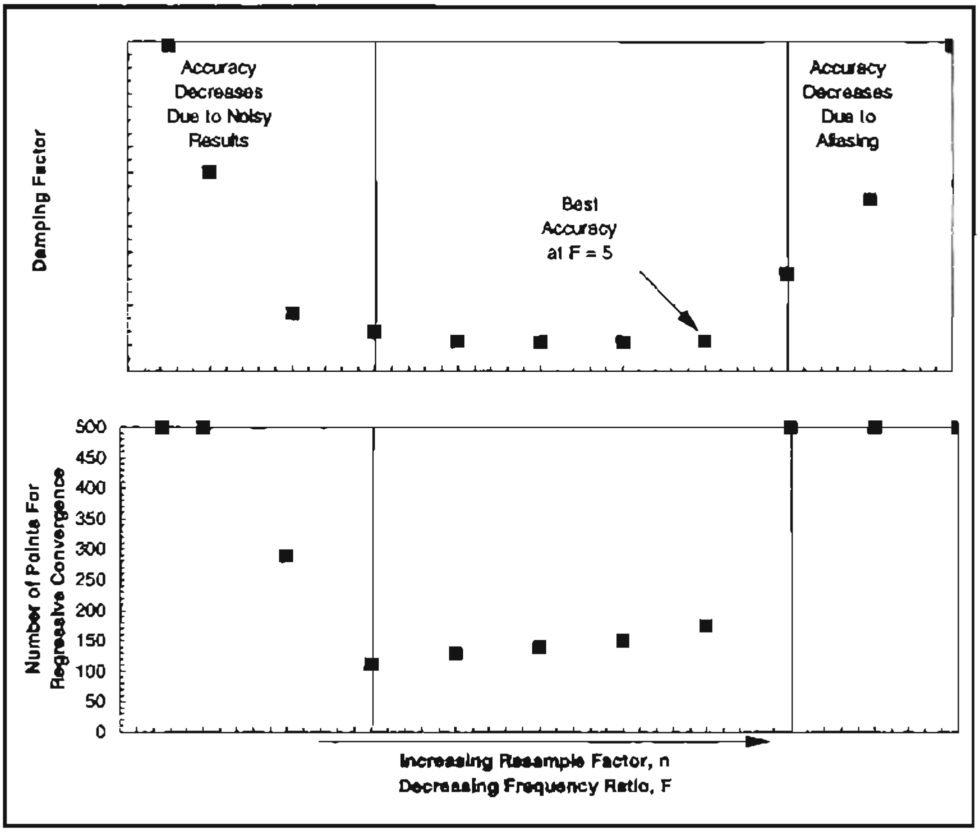


Figure 20. Typical Results Concerning The Accuracy of The Damping Factor and The Regressive Convergence Using the Same Model Order

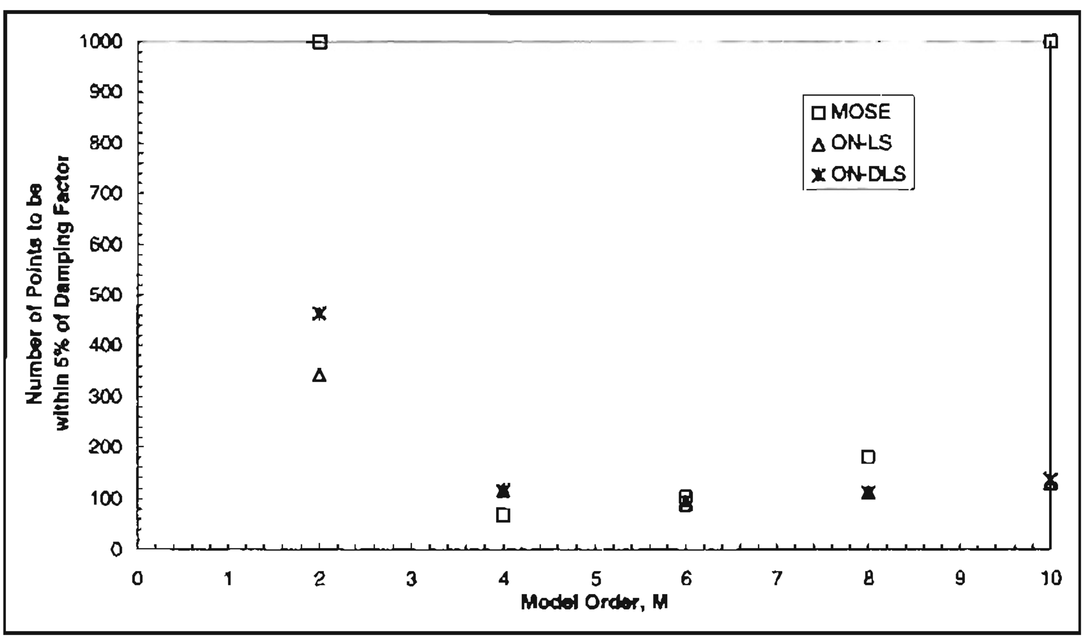


Figure 21. Number of Points to be Within 5% of the Damping Product Versus Model Order Using All Three Methods at a Re-Sample Factor of n = 4 Applied to Mode 1 of AGARD

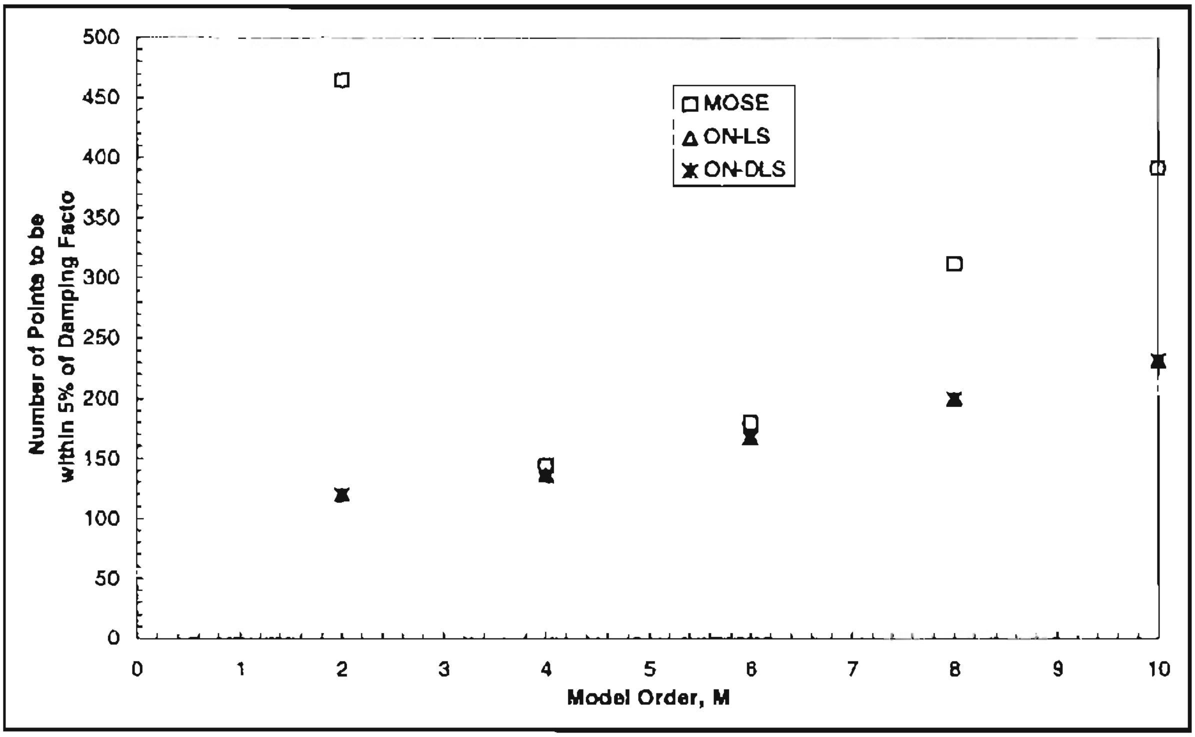


Figure 22. Number of Points to be Within 5% of the Damping Product Versus Model Order Using All Three Methods at a Re-Sample Factor of n = 8 Applied to Mode 1 of AGARD

3.2.1.3 Methods Applied to A Normalized Summation of All Modes or To Each Independent Mode Shapes?

The normalized summation of all modes is accomplished by normalizing each independent mode shape and summing these results together. The advantage of this is the model order is known, thus model overspecification is really not required (Cooper, 1990; Torii & Matsuzaki, 1997). However, for this unique set of independent mode shapes and for high mode systems such as the system presented in Figure 1 (GHV), normalized summation of the data can cause inaccurate results and several data points required for convergence. This was discussed briefly in Chapter 2.4. Another reason why not to use a normalized summation of all independent modes is based upon the results of the last section. A good re-sample factor cannot be chosen to be appropriate for every mode. If the lowest re-sample factor was chosen based upon the highest modal frequency, then the lowest modal frequency will be difficult to obtain and will take several points for regressive convergence. The lowest mode will have a very high frequency ratio compared to the highest mode. Section 3.4 provides two other reasons why a normalized summation of all modes shapes is not feasible. Because of these reasons all analysis will be applying each method to each independent mode using model overspecification to determine accurate modal parameters.

3.2.1.4 On Using Generalized Displacements or Velocities

Appendix B provides tables of all results from applying each method to both the generalized displacements and velocities at the properly determined re-sample factors of n = 8 (F = 8.4) for Mode 1 and n = 4 (F = 5.6) for Mode 2 of the AGARD system. Recall, one of the objectives for this system was to determine which set of data to use. All past data for this system has been using the generalized velocities. As shown below in Figure 23 and in Appendix B, the Generalized Displacement data for both Mode 1 and 2 produce poor results for convergence due to regression and model overspecification. This primarily occurs for the MOSE method, and was more prominent in Mode 1 than in Mode 2.

For the ON-LINE methods, the difference between using the generalized displacements or velocities based upon the number of points for regressive convergence and the accuracy of the damping factor was very small compared with the MOSE method. To determine the best method using further analysis and for the best comparisons between each of the three methods, the generalized velocities will be used as the input response based upon the results in Figure 23.

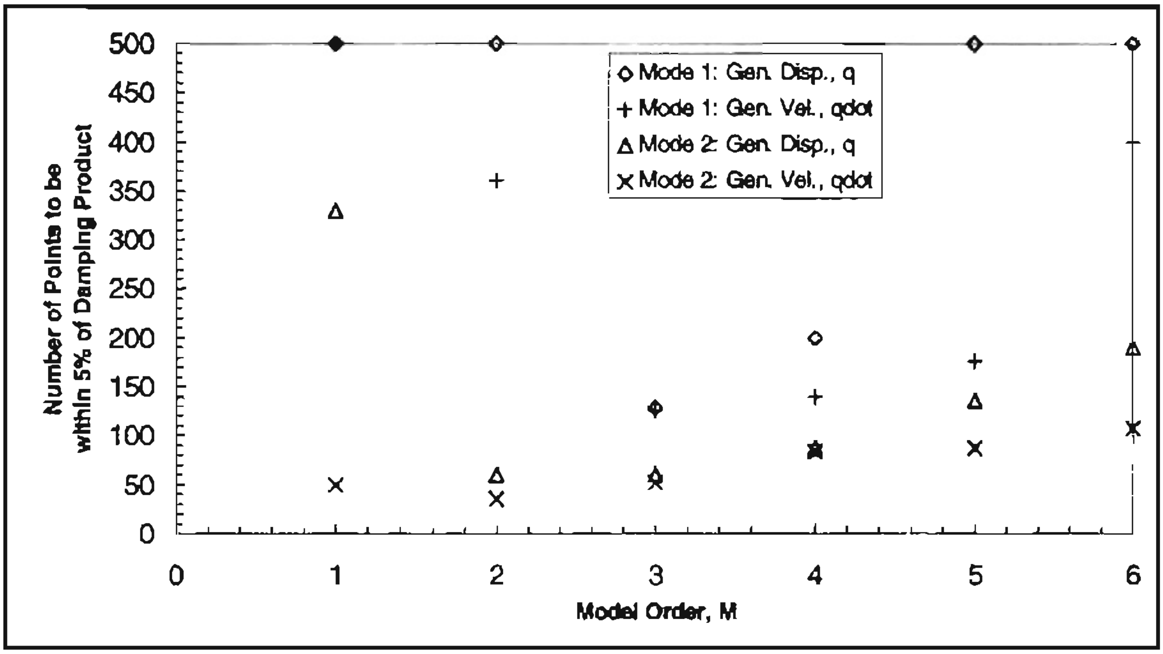


Figure 23. Number of Points to be Within 5% of the Damping Product Versus the Model Order Using MOSE with Proper Re-Sampling Factor Comparing The Number of Points for Regressive Convergence Between the Generalized Displacements and Velocities

3.2.1.5 Results For Both Modes

Using the previous findings in the past few sections both modes can be specifically analyzed using all three methods. To obtain the model parameters for each mode, these results in Appendix B for generalized velocities are compared between each model order. Modes that compare very well between different model orders are usually the system modal parameters. The damping products for the most common damping frequencies for Mode 1 and 2 from applying all three methods are provided in Table 3 for more finite model orders from M = 2 to 6. The damping frequencies are provided in Appendix B.

Mode	Model Order	MOSE	ON-LS	ON-DLS
1				
	2	-0.024	-0.028	-0.028
	3	-0.028	-0.028	-0.028
	4	-0.027	-0.028	-0.028
	5	-0.028	-0.028	-0.028
	6	-0.027	-0.028	-0.028
2				
	2	15.111	15.205	15.205
	3	15.207	15.206	15.206
	4	15.204	15.206	15.205
	5	15.202	15.206	15.205
	6	15.206	15.205	15.206

Table 3. Most Common Damping Products From Each Independent Mode from AGARD

The damping products for all methods were very similar, but based upon accuracy, the ON-LINE Methods provided the best results for all model orders.

Figures 24 and 25 provide plots examining the number of points for regressive convergence for each mode to obtain the results in Table 3. Figure 24 for Mode 1 just re-iterates some of the findings already, however, it does provide more details of a more feasible model order range. In both Figures 24 and 25, generally, the MOSE method converged in less amount of points than the other two methods for model orders of three or less, but more points were required for model orders greater than 4. Overall, the difference in the amount of points was very small, and the number of points for the ON-LINE methods were almost identical. The number of points were less for Mode 2, compared to Mode 1, because the frequency ratio was closer to five and not exceeding the re-sample factor limit of 8 (for a given F < 250) like Mode 1.

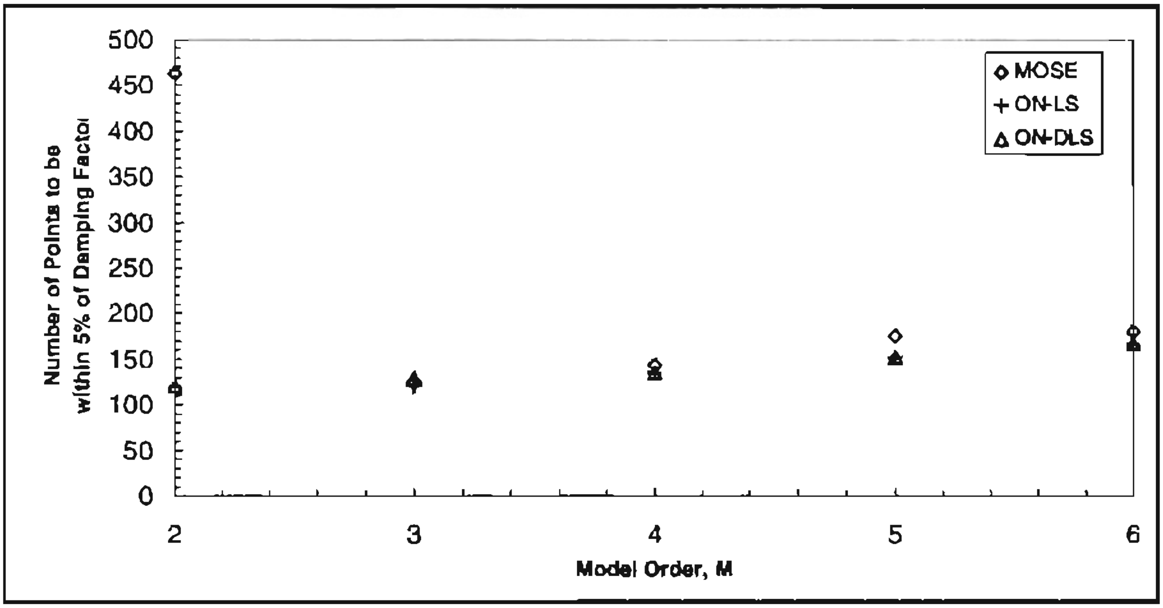


Figure 24. Number of Points to be Within 5% of the Damping Product, -0.028, Versus the Model Order For the Independent Mode 1 of AGARD Using All Three Methods at a Re-Sample Factor of n = 8 (F = 8.4)

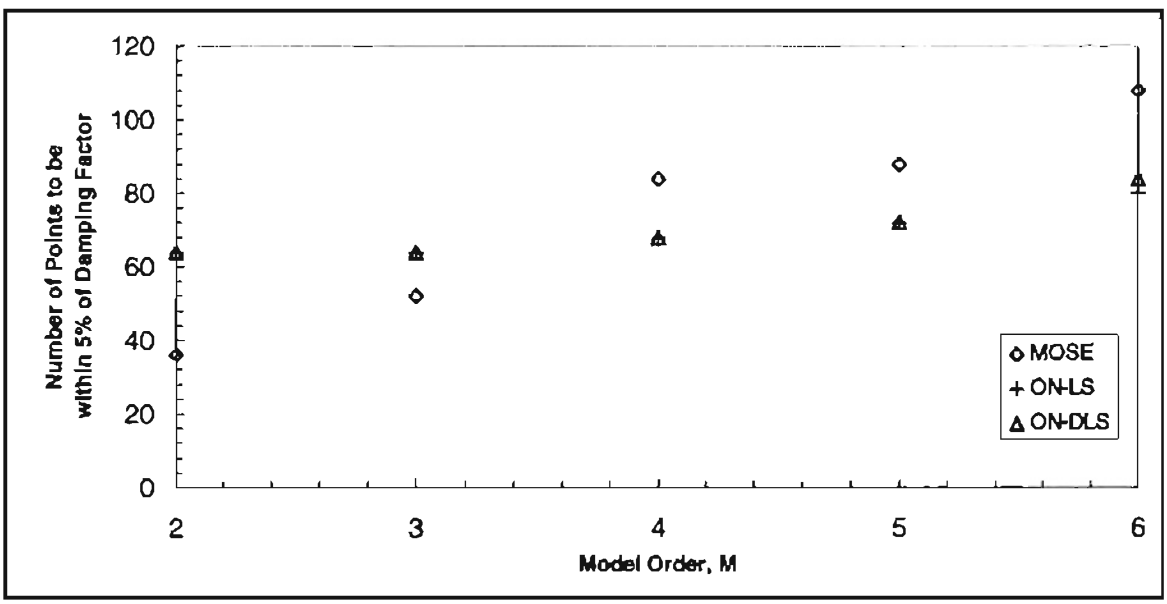


Figure 25. Number of Points to be Within 5% of the Damping Product, 15.206, Versus the Model Order For the Independent Mode 2 of AGARD Using All Three Methods at a Re-Sample Factor of n = 4 (F = 5.1)

3.2.2 Six Mode System (Flat Plate)

3.2.2.1 Description of Data

This set of response data is the results from a exciting six modes of a simple flat plate model, which the an isometric view is shown in Figure 26.

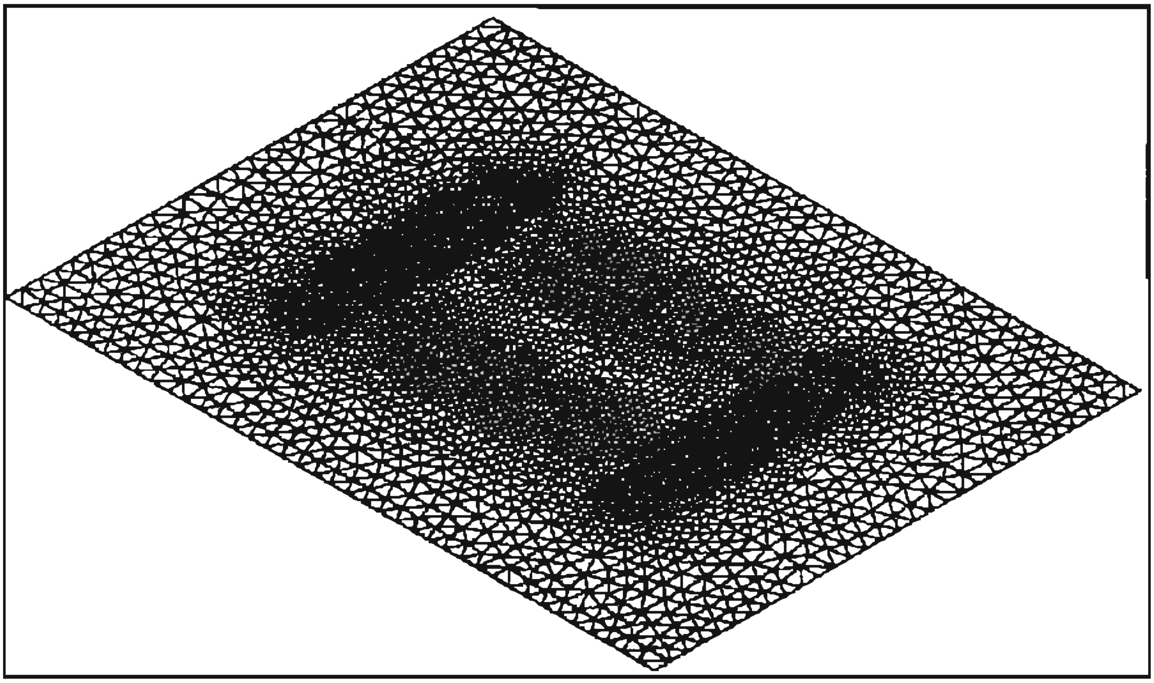


Figure 26. Isometric View of the Flat Plate

This particular data set includes six independent modes at a sub-critical flutter condition thus resulting in six very stable modes. The aspects that were learned when analyzing the two-mode system in Section 3.2.1 will be applied here, however, some these aspects will be briefly presented to re-iterate why they are used.

Figure 27 provides the time history of each independent mode and Table 4 provides the properties required to set up each method including the required re-sample factor resulting in new re-sample frequency ratios.

Item		Value
Number of Modes		6
Sample Frequency, ω_s, (Rad/s)		3.717.86
Number of Points, N		500
Last Point of Excitation, z		8
Natural Frequencies @ Q = 0, (Rad/s)		
Mode 1		18.591
Mode 2		76.179
Mode 3		137.802
Mode 4		183.732
Mode 5		277.414
Mode 6		371.785
Frequency Ratio, $F = \omega_s/\omega_n$:		
Mode 1		200.0
Mode 2		48.8
Mode 3		27.0
Mode 4		20.2
Mode 5		13.4
Mode 6		10.0
Re-sample Factor with New Frequency Ratio :	n	F
Mode 1	8	25
Mode 2	8	6.1
Mode 3	5	5.0
Mode 4	4	5.0
Mode 5	3	5.0
Mode 6	2	5.0

Table 4. Properties of Input Response Data From Flat Plate Model

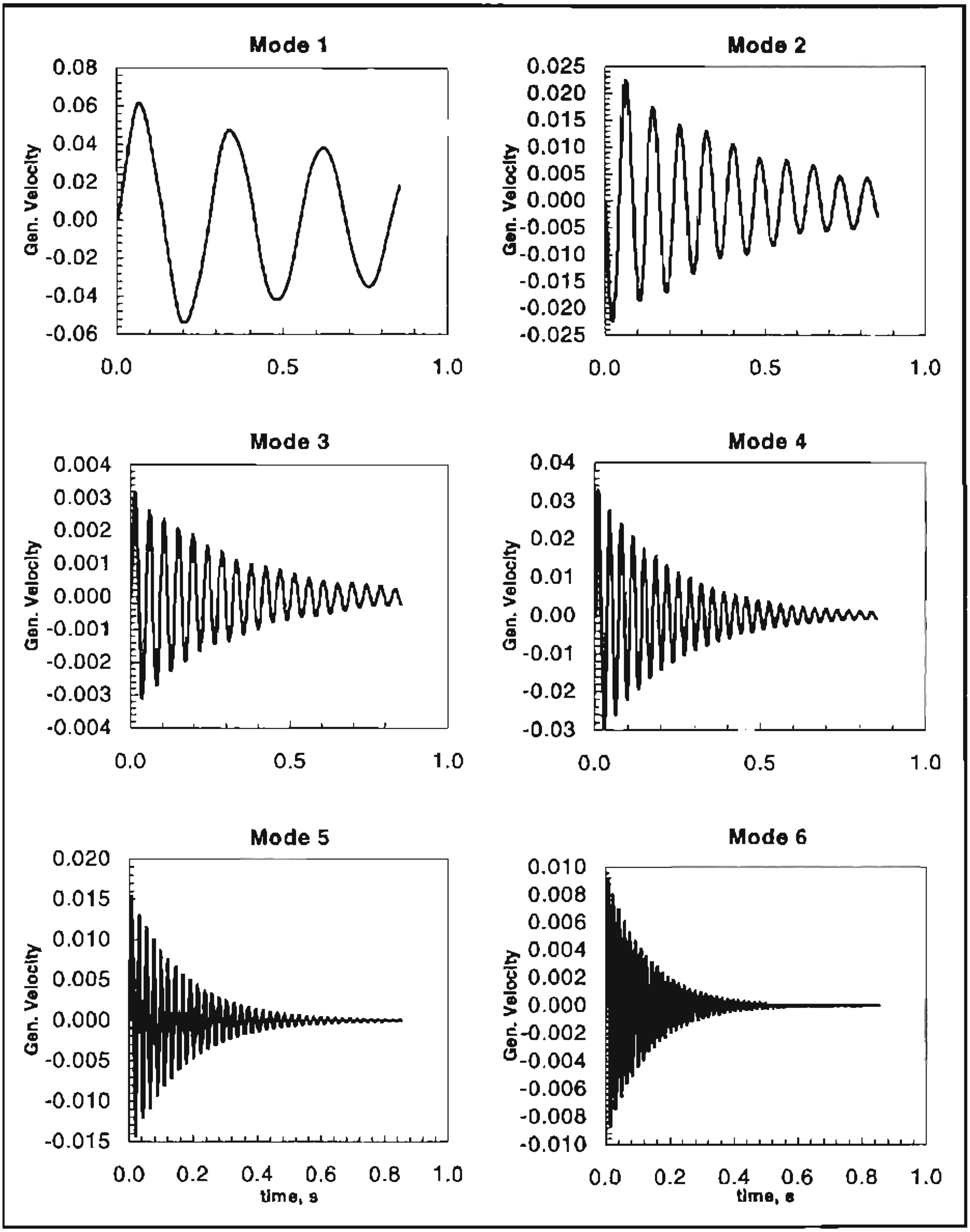


Figure 27. Six Independent Mode Shapes from Generalized Velocities of Flat Plate Model

3.2.2.2 Re-Sampling of Data and Model Overspecification

For these modes, the results of re-sampling the data at a lower sample frequency (lower frequency ratio) were very comparable to the results for the two mode system in Section 3.2.2. As the re-sampling factor increased (re-sample frequency decreased), the damping product's accuracy improved. Also, model overspecification increased the accuracy of the damping product. Figures 28 and 29 provides an example of these results of the damping product and the number of points for convergence from using both the MOSE and ON-LS methods on Mode 4 for this system.

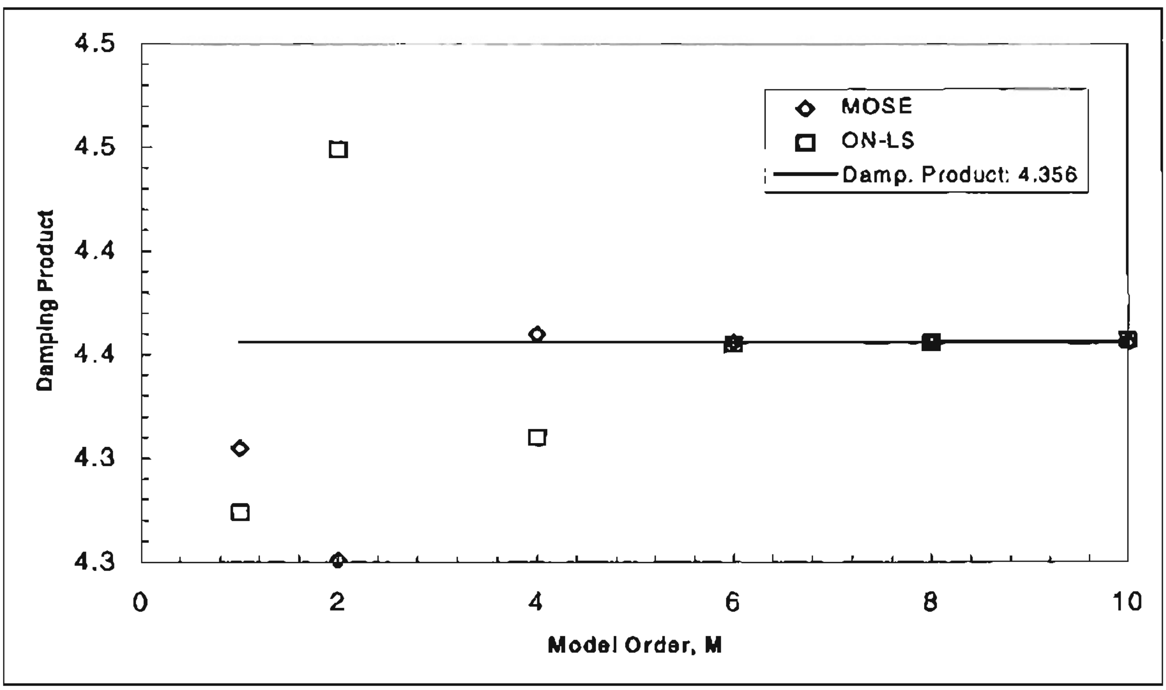


Figure 28. Damping Product, σ, Versus Model Order, M, Using Both MOSE and ON-LS Methods on Mode 4 of the Flat Plate System at a Re-Sample Factor of $n = 4$ ($F = 5.1$)

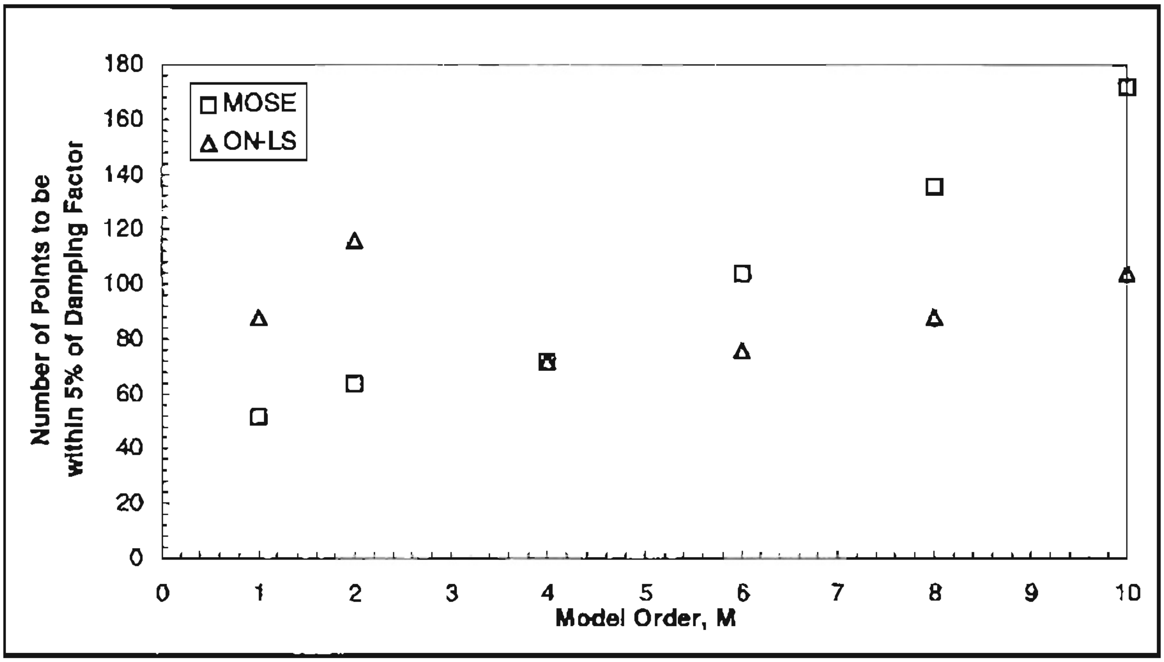


Figure 29. Number of Points to be Within 5% of the Damping Factor, 4.356, Versus the Model Order, M, Using Both MOSE and ON-LS Methods On Mode 4 of the Flat Plate System at a Re-Sampling Factor n = 4 (F = 5.1)

From these two figures for model orders from 4 and below, the number of points for regressive convergence were lower using MOSE and the accuracy of the damping product was generally better. For model orders greater than four, the accuracy for both methods were very similar, but, the number of points for regressive convergence increased more rapidly for MOSE method than with the ON-LS method. The exact same results was seen the two mode AGARD system.

3.2.2.3 Results For All Modes

Similar to the AGARD system, the modal parameters are identified by finding the most common modal parameters between models of different order. Other modes may

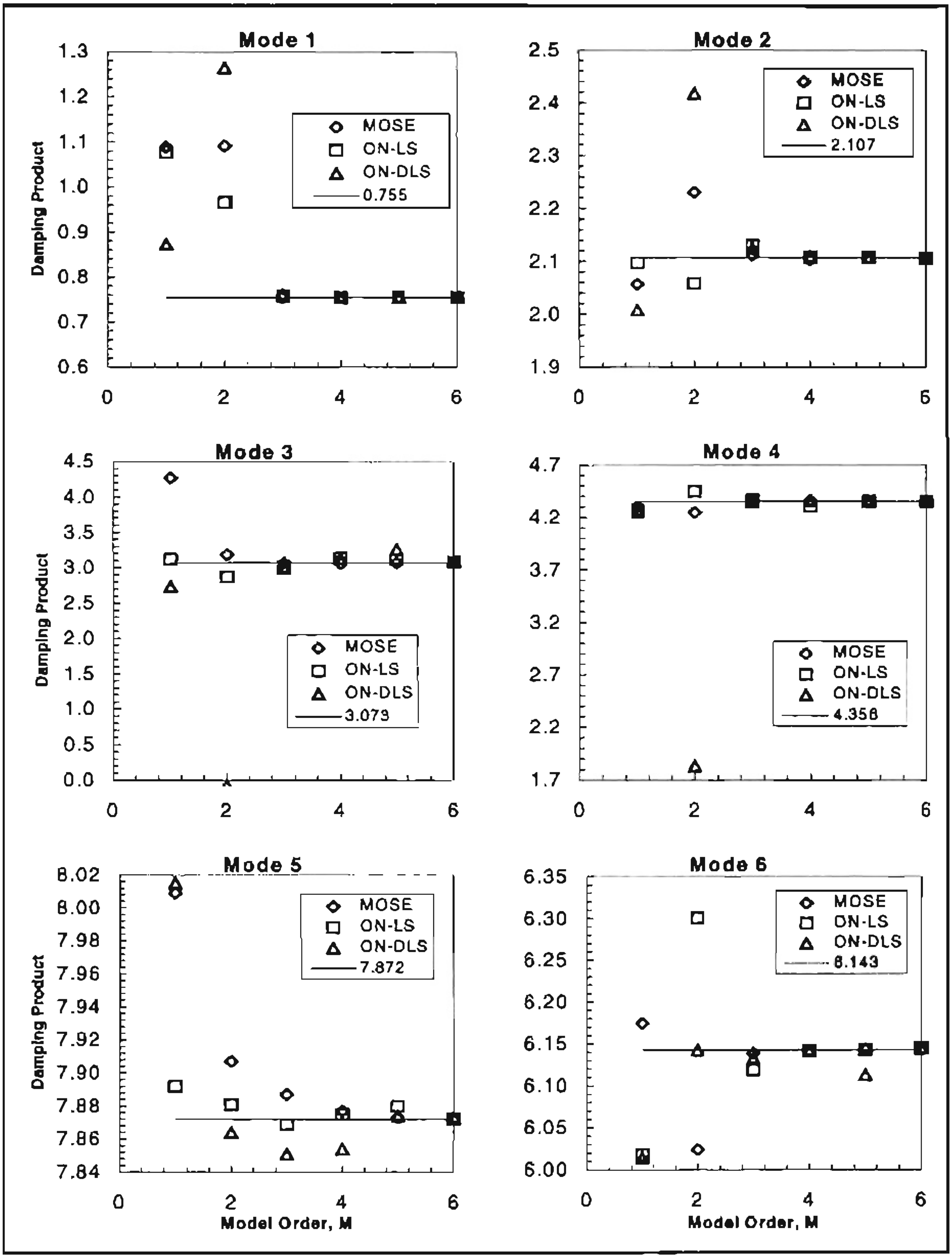


Figure 30. Common Damping Products From Each Mode For All Methods using the Flat Plate System

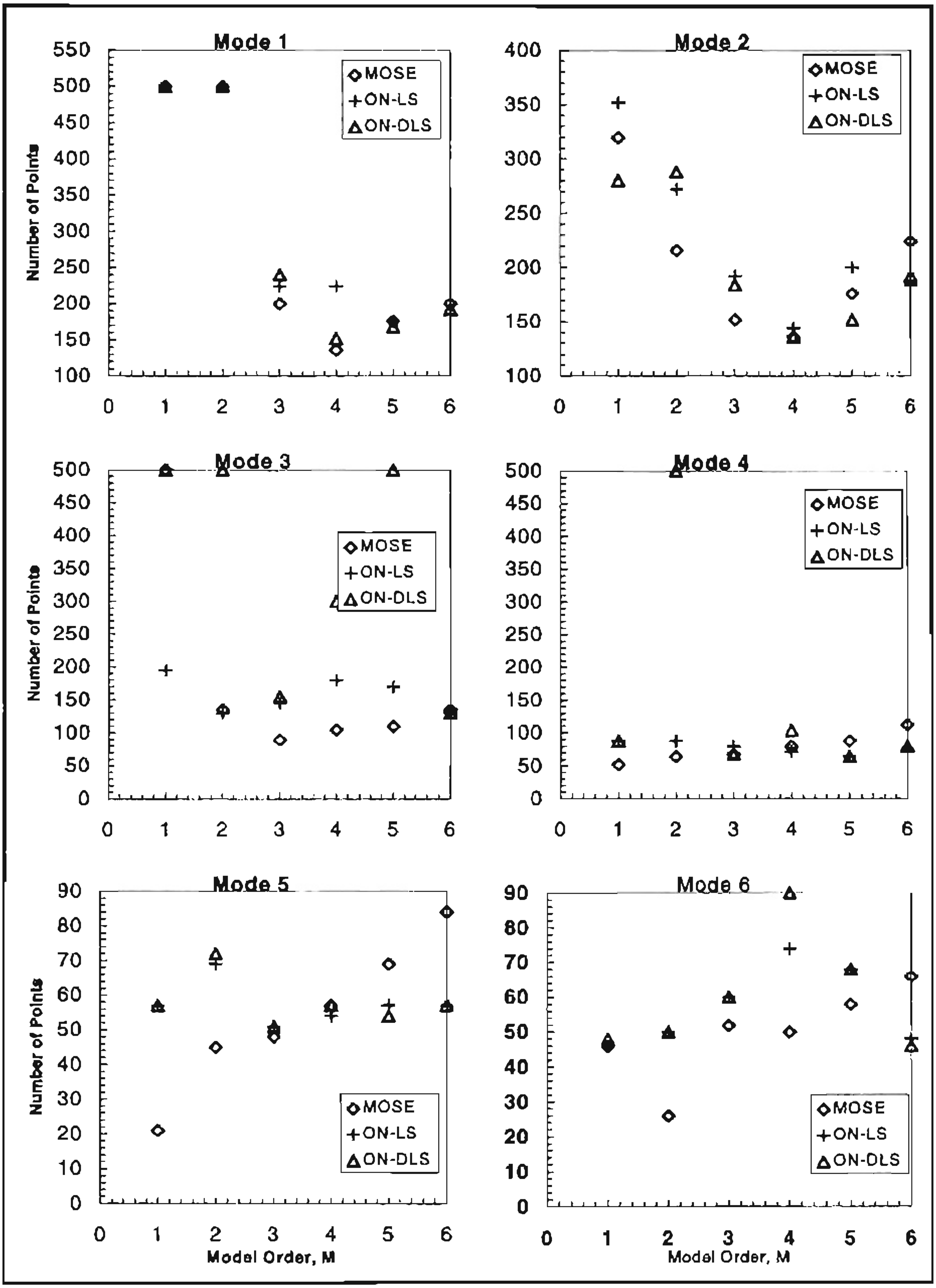


Figure 31. Number of Points To Be Within 5% of Actual Damping Product
For Each Mode For All Methods Using of the Flat Plate System

be present and the modal parameter for that independent mode response may not be the most common. However, if starting from Mode 1 and finding the most common, Mode 1 is identified. Then when moving on the Mode 2, if Mode 1 is present and it being the most common then Mode 1 can be ignored and then the next most common may be identified being most likely Mode 2. This criteria is used with all subsequent modes. Figures 30 and 31 provide the results of only the identified damping products and the number of points for convergence for these damping products comparing all three methods for each mode.

The accuracy of the damping products for this system was generally good for all model orders greater than three. For model orders of one and two the results were varying. For model orders greater than three the MOSE method did provide the best results in the accuracy because the two ON-LINE methods sometimes would produce an inaccurate result (Mode 3 through 6). As for the number of points for regressive convergence in Figure 31, all results were basically similar in the amount of points except the MOSE method generally converged in a less amount of points for all model orders and still providing good accuracy of the damping factor. Based upon these results, these findings proved that MOSE did provide overall better results for all mode orders below six which was enough model overspecification to obtain accurate damping products. This conclusion is similar to the findings of the last two systems in terms of low model orders.

To verify the correct modal parameters were obtained and being that these modes were obtained at a sub-critical flutter state, most likely each mode shape can be represented using a single mode sinusoidal damped equation. However, this assumption will not be entirely accurate because other modes were found to exist. Table 5 provides the modal parameters for each mode using the MOSE method of model order of six.

Mode	ω_D (rad/s)	σ (rad/s)
1	22.90	0.755
2	74.98	2.110
3	138.33	3.090
4	182.65	4.350
5	277.10	6.145
6	371.79	7.872

Table 5. Modal Parameters For Flat Plate System

Figure 32 provides the normalized input response for each mode shape from the ASENL data plotted with the single mode sinusoidal damped representation using the appropriate modal parameters from Table 5. The single mode model almost modeled every mode exactly except maybe Mode 1. The point of regressive convergence is based upon a model order of 6 using the MOSE method.

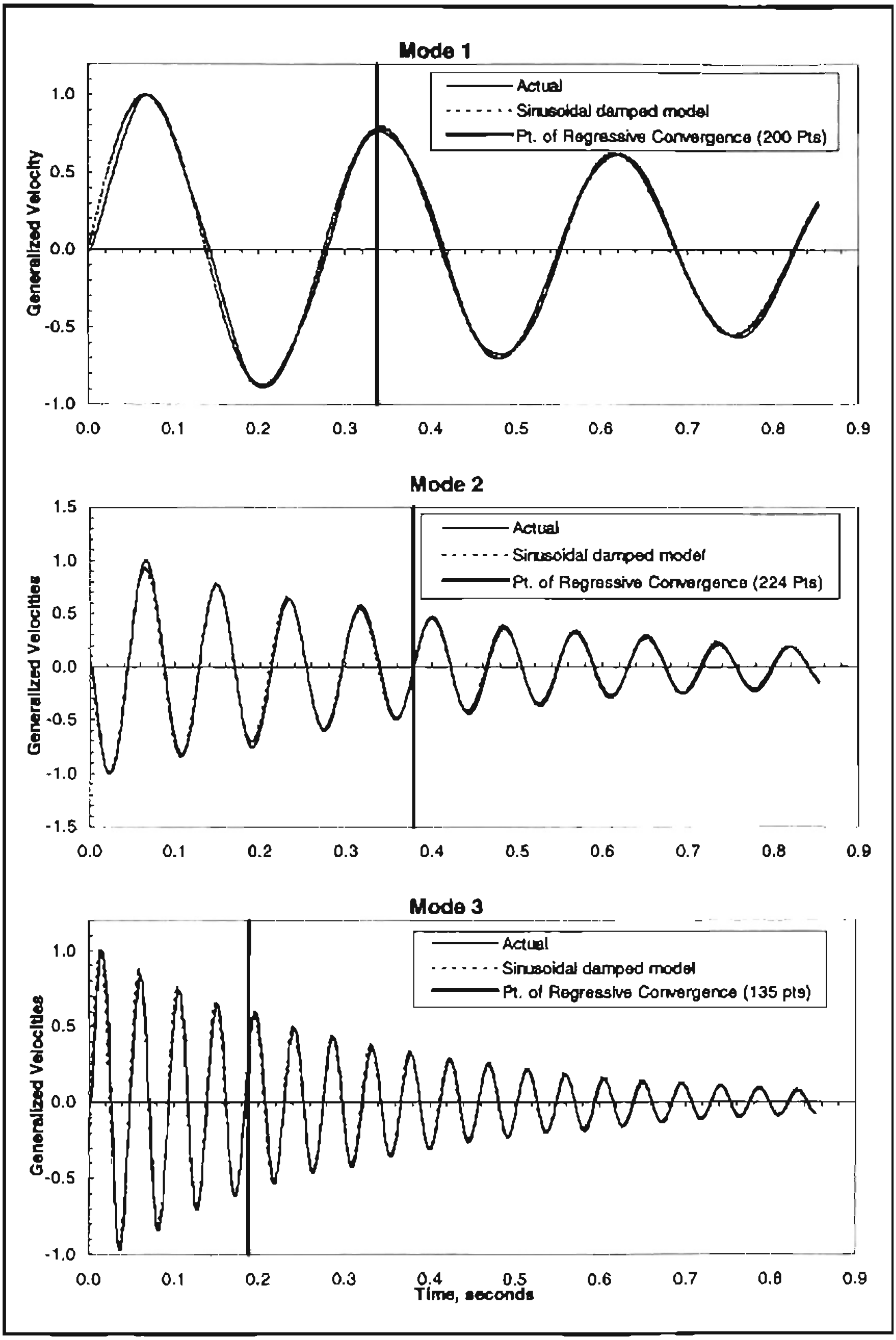


Figure 32. Modeling of Original Normalized Input Response Data From the Flat Plate System Using Identified Modal Parameters

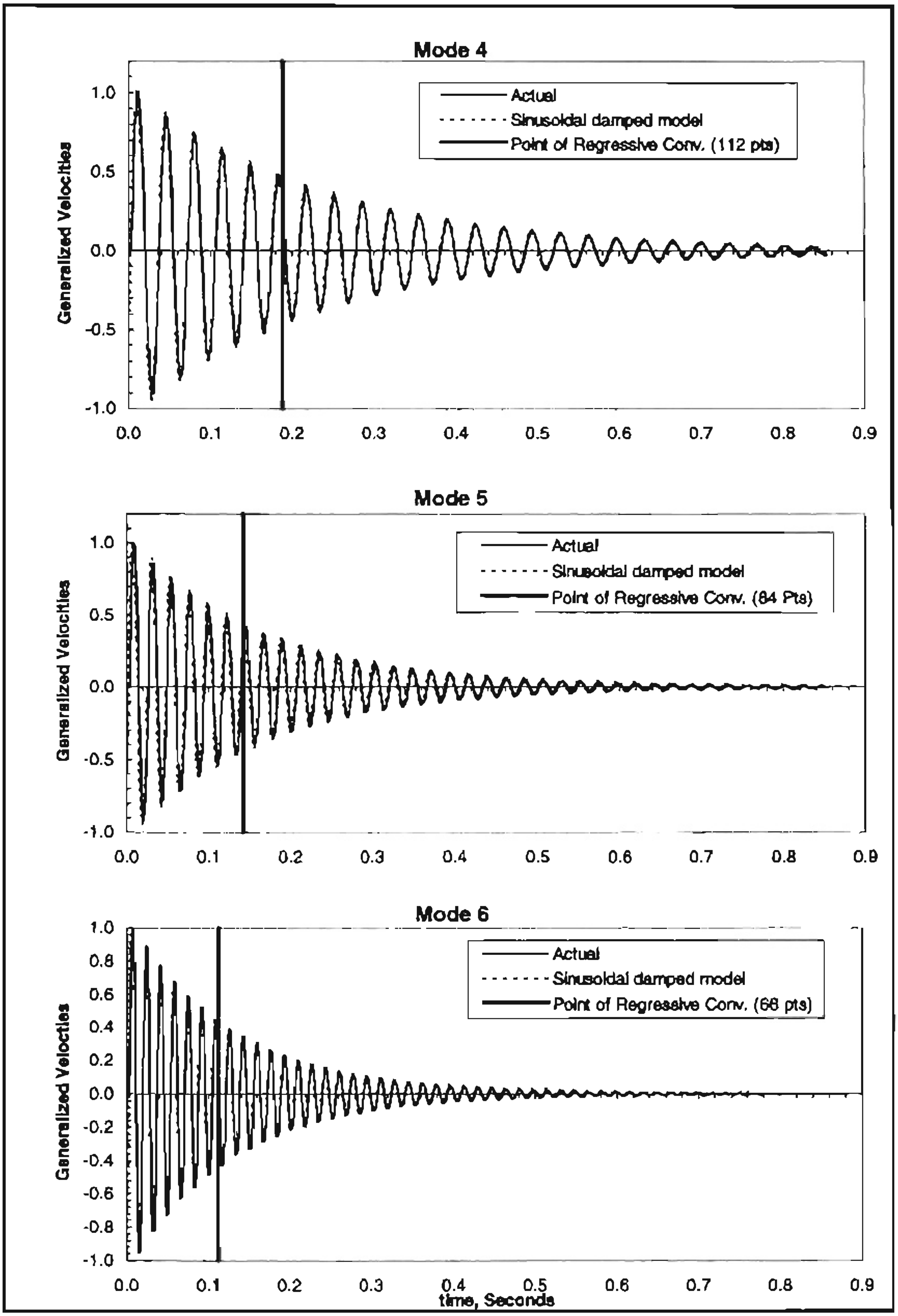


Figure 32 Continued. Modeling of Original Normalized Input Response Data From the Flat Plate System Using Identified Modal Parameters

3.2.3 Nine Mode System (GHV)

3.2.3.1 Description of Data

This data set is the result of exciting a nine mode system of a model of the Generic Hyperspace Vehicle (GHV), shown in Figure 33.

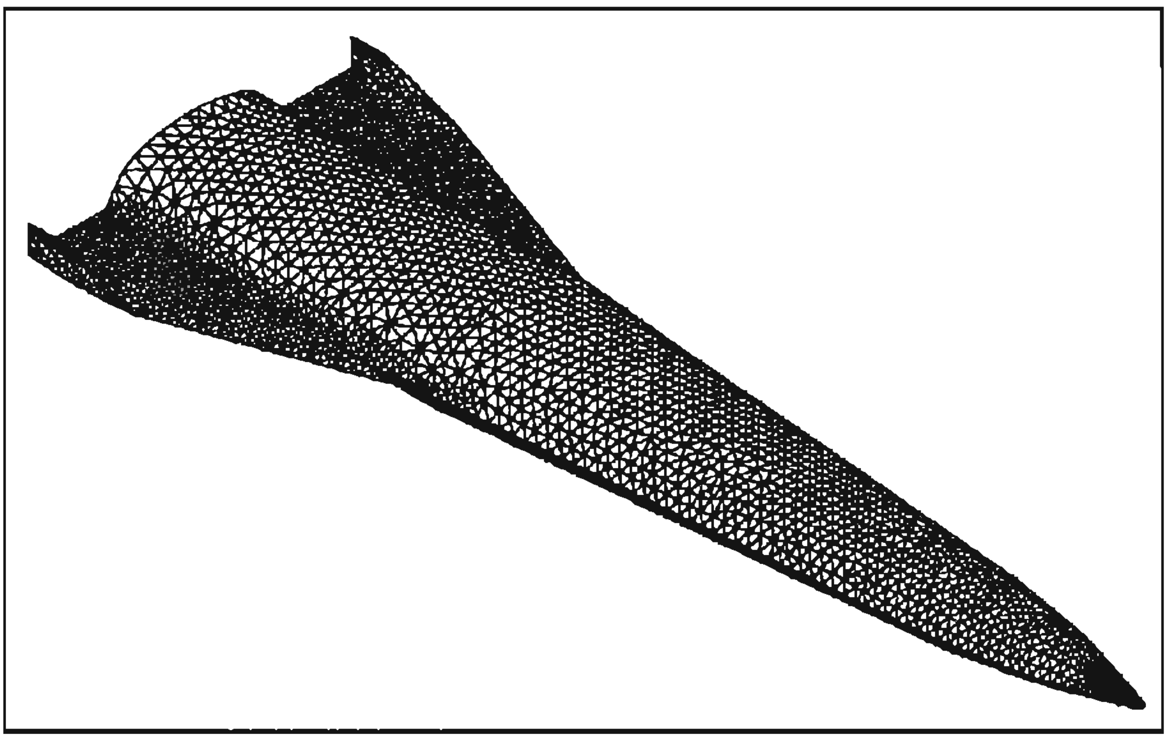


Figure 33. Geometry of GHV Model

This system is at a condition just beyond the flutter boundary resulting in three of the nine modes being usually unstable as shown in the Figure 1. This set of data provided the greatest difficulty in identifying the system modal parameters. This system not only contains nine modes, which is the largest system analyzed, but, it also contains three sets of two modes that are very closely spaced. This can be seen from the natural frequencies in Table 6. Table 6 also provides the necessary properties of the system and the pre-determined new frequency ratios for re-sampling the input data for each method. Not only does this system have closely spaced modes, but the damping products are also very close as will be shown later. Therefore, this system is very complicated.

Each of the three methods were applied to each independent mode using model overspecification and the appropriate re-sample factor shown in Table 6. Several higher model orders were used on several of the modes. This was accomplished primarily with the closely space modes because convergence, due to model overspecification, of the damping product was sporadic as the model order increased. This issue will be discussed in the next few sections.

Item		Value
Number of Modes		9
Sample Frequency, ω_s, (Rad/s)		943.42
Number of Points, N		500
Last Point of Excitation, z		7
Natural Frequencies, ω_n @ Q = 0, (Rad/s)		
Mode 1		19.993
Mode 2		25.312
Mode 3		25.462
Mode 4		34.918
Mode 5		43.435
Mode 6		43.801
Mode 7		45.657
Mode 8		59.069
Mode 9		59.286
Frequency Ratio, $F = \omega_s/\omega_n$:		
Mode 1		47.2
Mode 2		37.3
Mode 3		37.1
Mode 4		27.0
Mode 5		21.7
Mode 6		21.5
Mode 7		20.7
Mode 8		16.0
Mode 9		15.9
Re-sample Factor with New Frequency Ratio :	n	F
Mode 1	8	5.9
Mode 2	7	5.3
Mode 3	7	5.3
Mode 4	5	5.4
Mode 5	4	5.4
Mode 6	4	5.4
Mode 7	4	5.2
Mode 8	3	5.3
Mode 9	3	5.3

Table 6. Properties of Input Response Data From GHV

3.2.3.2 Re-Sampling of Data and Model Overspecification

Re-sampling this data at a lower frequency ratio and using model overspecification had very similar results to the previous two systems. Above a model order of four, accuracy of the damping product improved for both methods as shown in Figure 34 using Mode 4. The number of points for regressive convergence for modes that were not so closely spaced to other modes was nearly the same for both the MOSE and ON-LS method for all modes. An example of this is shown in Figure 35 for Mode 4. For closely space modes the performance of each method varied. This will be discussed in more detail in the next section.

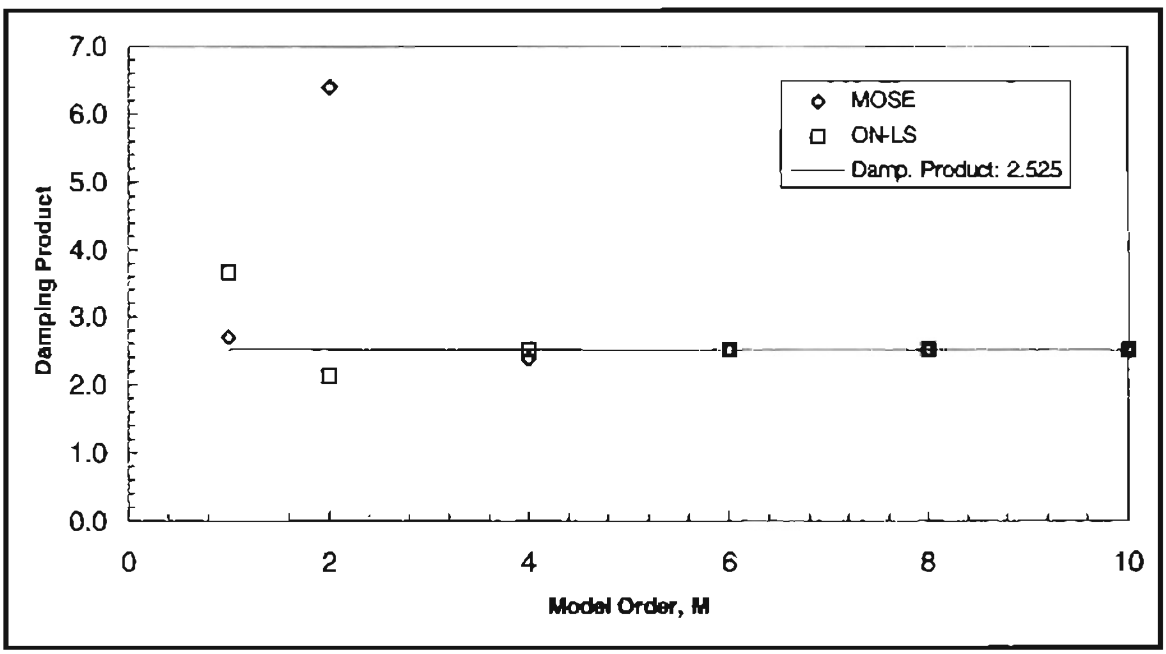


Figure 34. Damping Product, σ, Versus Model Order, M, Comparing Both MOSE and ON-LS Methods Using Mode 4 of the GHV System At A Re-sample Factor n = 4 (F = 5.2)

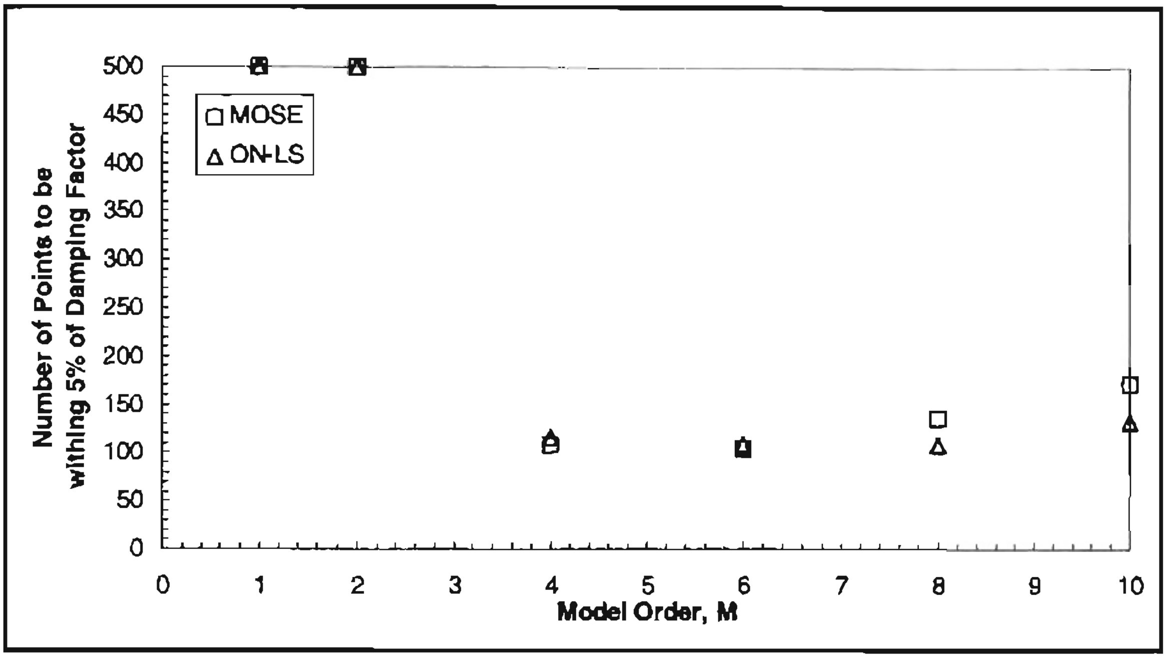


Figure 35. Number of Points to be Within 5% of the Damping Factor, 2.525, Versus the Model Using Comparing Both MOSE and ON-LS Methods Using Mode 4 of the GHV System At a Re-sample Factor of n = 4 (F = 5.2)

3.2.3.3 Results For All Modes

To identify the modal parameters between different model orders, the same criteria as in the six mode system was used. For this complex nine mode system, the system modal parameters were more difficult to identify due to the closeness of several modes and the number of modes existing. The damping product generally varied sporadically with increasing model order for closely spaced modes. Sometimes the other closely spaced mode was approached as in Mode 2 for a model order above five. Being that the damping product and frequency are similar between the closely spaced modes, this criteria was even more difficult. However, if enough higher model orders were used and each mode is carefully examined then each mode can be identified. Usually in most cases, the modal parameters for the mode of interest in that mode shape between the closely spaced modes had better characteristics of regressive convergence with higher model orders.

Figures 36 and 37 graphically provide the results of the identified modes from each independent mode shape in terms of the damping product and the number of points required to regressively converge upon an accurate damping product. If no values are seen for a particular model order then regressive convergence failed.

The main observations for all three methods is that with increasing model order, the accuracy of damping factor did improve. The most significant improvement was made for model orders greater than three to five depending upon the mode. Sometimes the MOSE method was not as good as compared to the other methods as shown for Mode 2, however, the MOSE method was better than the other methods as in Mode 6. For Mode 6 at a model order of 8 the common damping product was never obtained for the ON-LINE methods. As for the number of points for convergence, the MOSE method was generally better than the other two methods for model orders below six, and for higher model orders the ON-LINE methods were not much improvement.

After examining all methods applied to the complex nine mode system, the MOSE method provided results that were not as accurate as the ON-LINE methods, however, did produce the results in less amount of points at lower model orders. The results of the ON-LINE methods in obtaining the damping factor were not much improvement over the results from the MOSE method until higher modes were reached. Usually with these higher model orders, more calculated and system modes are obtained, therefore, the difficulty of identifying common modal parameters becomes greater especially for closely space modes. For example, when examining a mode shape for a mode that is closely

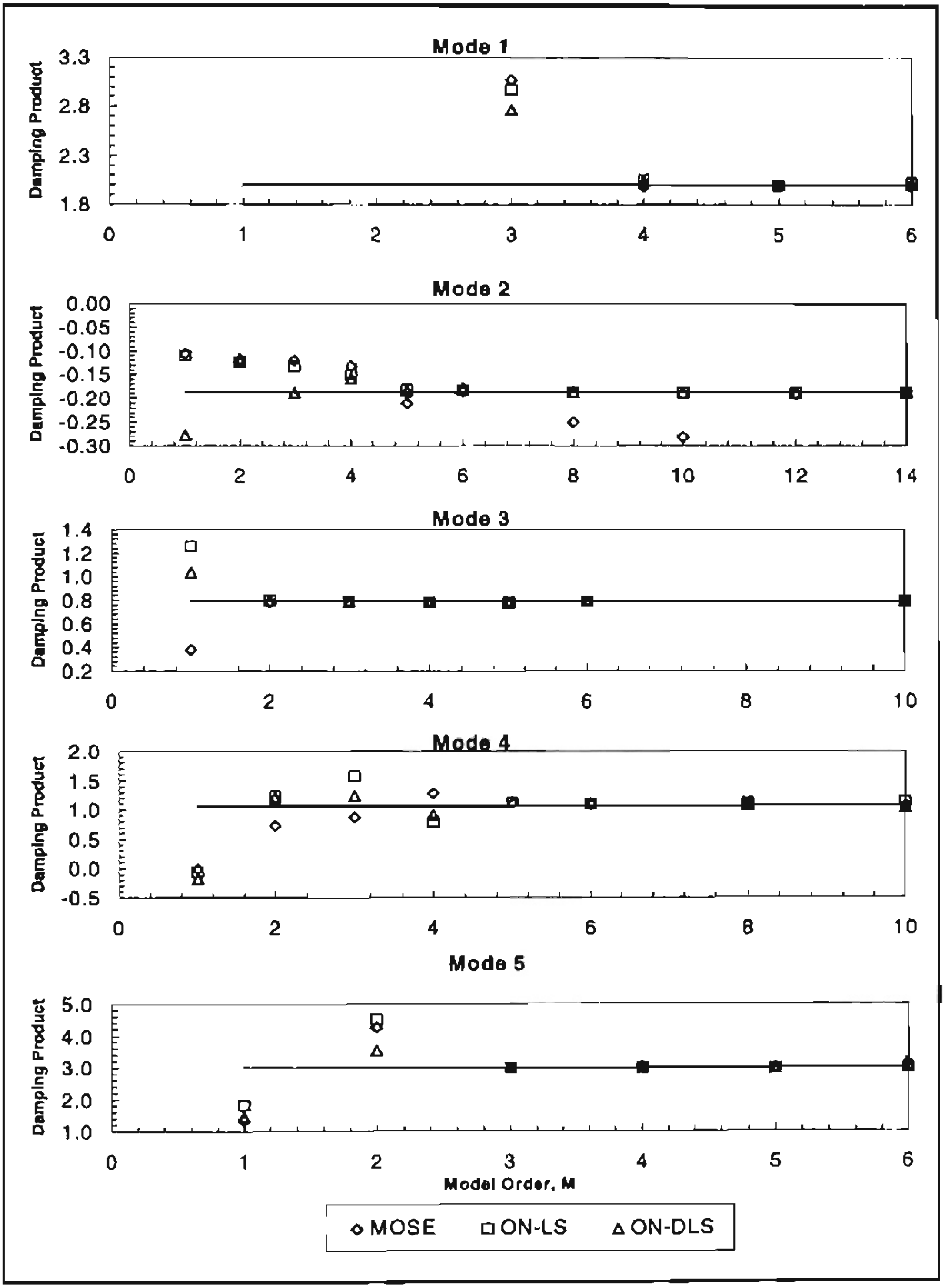


Figure 36. Most Common Damping Product From Each Mode For the GHV System Comparing All Three Methods

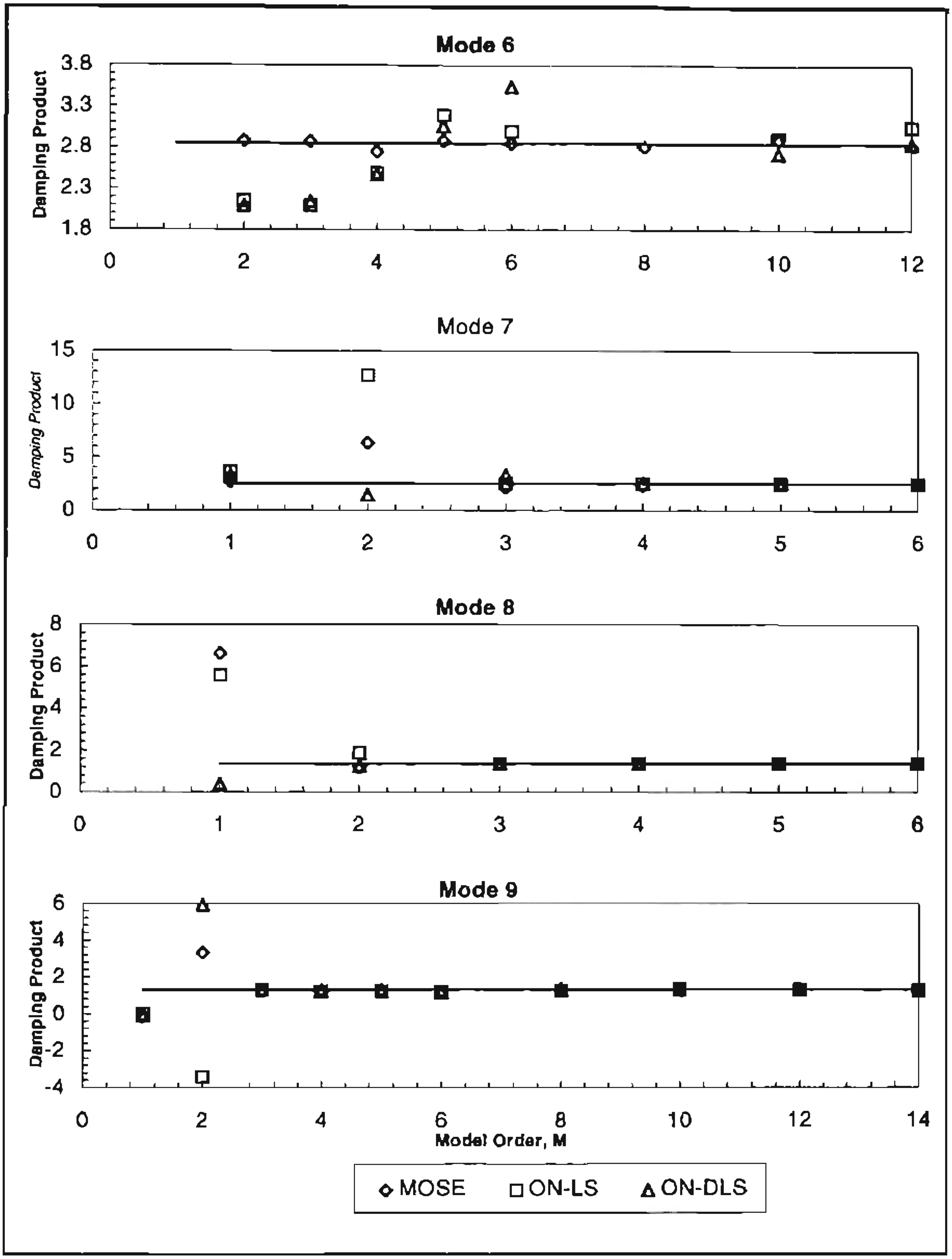


Figure 36 Continued. Most Common Damping Product From Each Mode For The GHV System Comparing All Three Methods

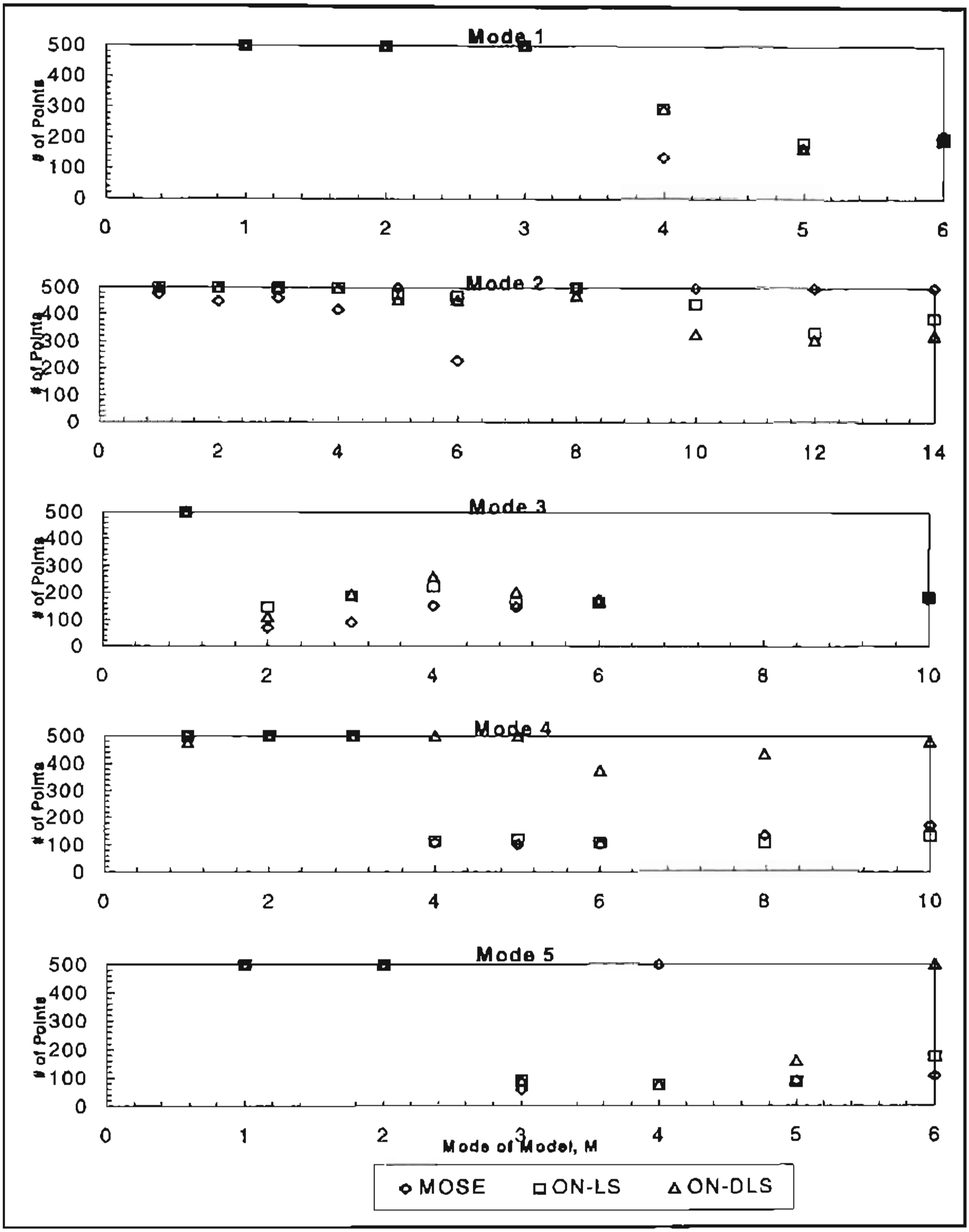


Figure 37. Number of Points to be Within 5% of the Damping Product for Each Mode of GHV Comparing All Three Methods

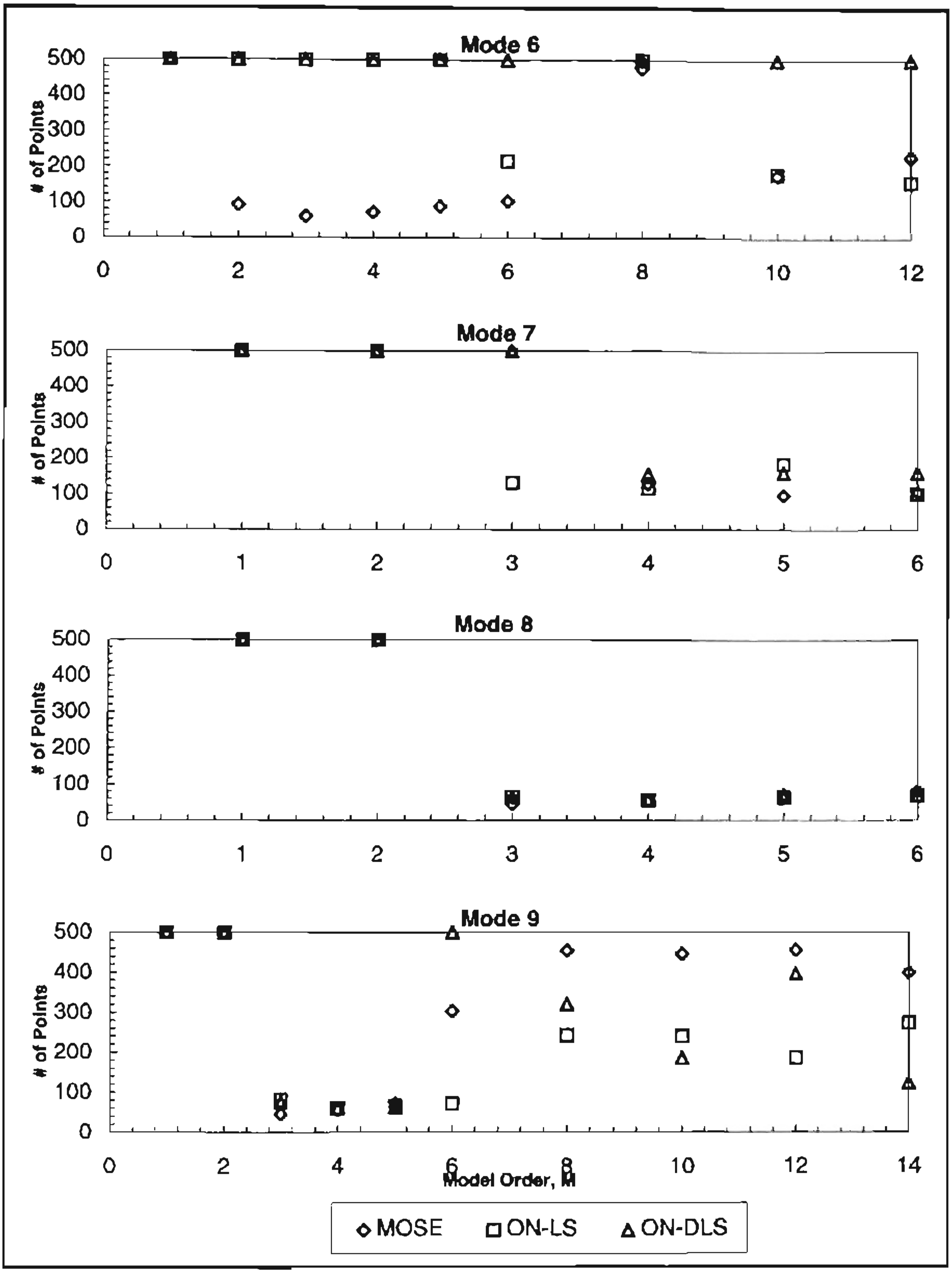


Figure 37 Continued. Number of Points to be Within 5% of the Damping Product for Each Mode of GHV Comparing All Three Methods

spaced with another mode, the other mode usually does not get identified until model orders greater than six. Therefore, for low model orders, if results are soundly obtained, then the less amount of the unwanted calculated modes and other system modes would make it easier in identifying the mode of interest for that particular mode shape. The MOSE method does this for model orders near four and five. Again, this method may not provide the most accurate results for this particular system, but it results are still within 5% of the actual result. This statement can only be made with this nine mode system of closely space modes, because for the other three systems analyzed the MOSE method did provide very good accuracy for lower model orders.

The final modal parameters for this system using the results of the MOSE method are shown in Table 7.

Mode	σ (rad/s)	ω_D (rad/s)
1	2.005	22.986
2	-.190	31.078
3	0.793	30.740
4	1.069	34.818
5	3.016	38.188
6	2.813	38.413
7	2.521	46.226
8	1.361	65.206
9	1.337	64.217

Table 7. Final Modal Parameters of GHV Model Taken From the MOSE Method

3.3 Stability of Regressive Convergence Upon Modal Parameters

Before a method is actually decided upon, one more aspect about these three methods, that has not been discussed, is the instability of the regressive convergence. For all methods, the instability of regressive convergence increases with higher model orders. The instability for the ON-LINE methods usually lasts longer than the MOSE method. Figure 38 provides an example of the characteristics of this phenomenon comparing the regressive convergence between all three methods. This data was obtained from the GHV system for Mode 8 and a model order of six. In this figure, the instability is stated as "switching". After examining the instability closer, the eigenvalues or roots obtained from the AR coefficients, are not necessarily obtained in the same order while marching through the data. Thus, the modal parameters are ordered differently during the march. Pinkleman and Batill realized for higher model orders that sorting of the eigenvalues must be done for each point so that actual regressive convergence can be seen. Sorting the eigenvalues this way can be difficult for closed space modes.

In Appendix B, showing the results of the all methods applied to the AGARD wing previously discussed, the last two columns provide when the modal parameters were first encountered to be within 5% of the actual damping product and the point at which the instability quit and regressive convergence of the modal parameters was obtained. It is shown here that the ON-LINE methods provide poor results, in terms of instability of regressive convergence compared to the MOSE method. In these tables if a dash is seen in the last column this means the instability never quit up until the last point of the data was used. This usually occurred more frequently for the ON-LINE methods. Therefore, the MOSE method provides better results in terms of the instability of regressive convergence.

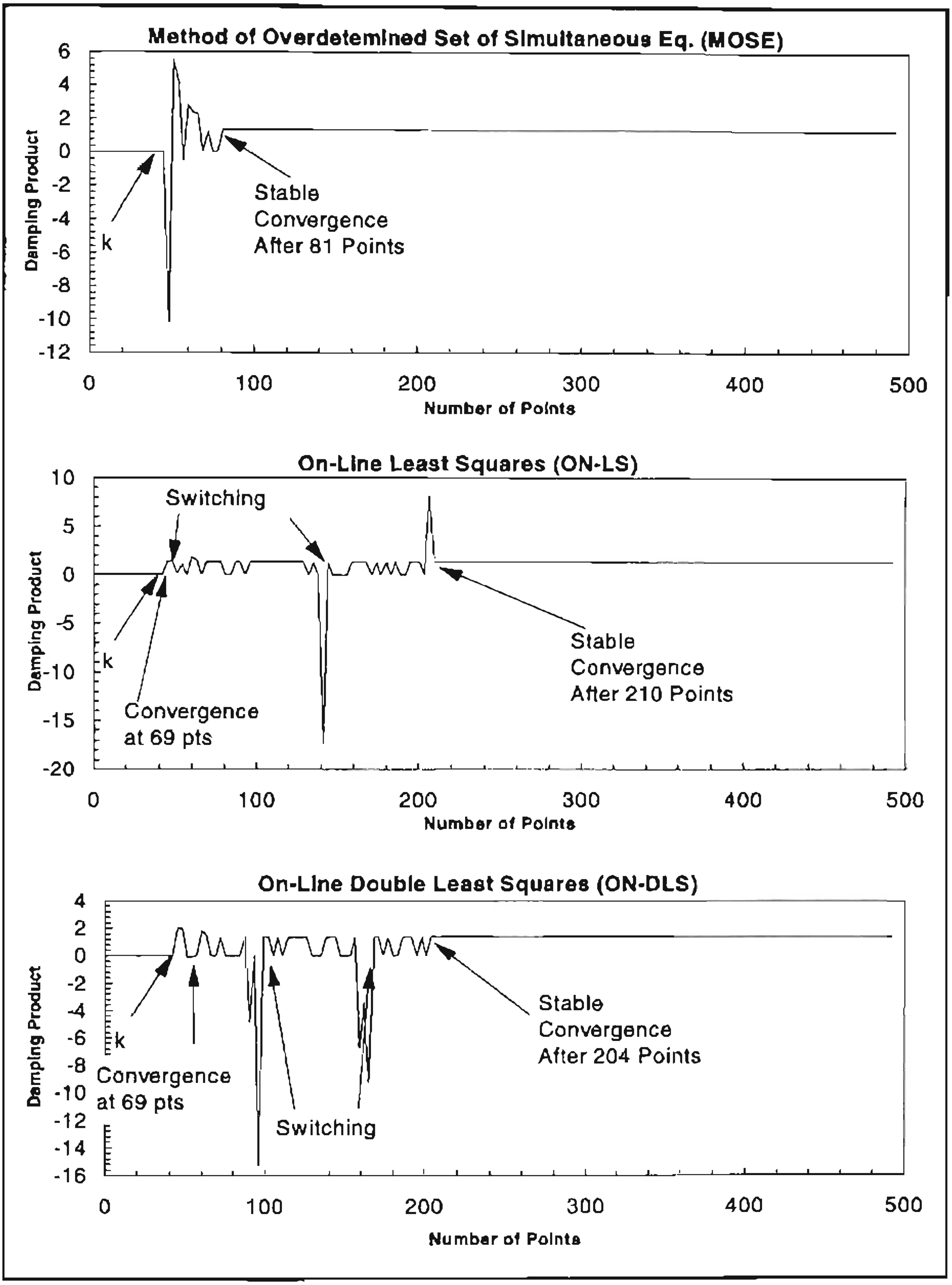


Figure 38. An Example of Characteristics of Convergence Between Each Method

3.4 Normalizing of ASENL Data

As previously mentioned, re-sampling a normalized summation of all mode shapes at a lower frequency ratio based upon the highest mode may cause accuracy and convergence problems of the lower modes. It was also discussed, the number of modes trying to identify in the single data stream requires more points for regressive convergence which is also true for higher mode systems. This was why only individual modes were analyzed.

Two other reasons can be used to support why normalizing each mode and summing them all together is not feasible as input data for these methods. The primary reason was developed from the results of the nine mode GHV system. By looking at the results from Table 7. The damping frequency from Mode 4 was less than the damping frequency from Mode 3. The same can be said about Mode 8 and 9. From looking at the natural frequencies given in Table 6 for these modes, Modes 4 and 9 have decreased so much as to be less than Mode 3 and 8, respectively. When using a normalized summation of the input response data, this phenomenon could not be recognized. When all model parameters were identified using a high model order, at least 9, with any method the modes would then be sorted based upon frequency. Thus Mode 3 and 8 would be less than Mode 4 and Mode 9, respectively, which is not actually the case. Therefore, this closely space nine mode model would not be analyzed correctly using a normalized summation as the input data.

The second reason for not using a normalized summation is based upon the occurrence of "switching" occurring for higher model orders. For these models it would be more difficult to identify all modes correctly and efficiently due to regressive convergence.

3.5 Method of Choice

Primarily based upon the results of analyzing the nine mode system, the ARMA model using the MOSE method will be developed into a stand-alone, autonomous, FORTRAN 77 program to assist in estimating flutter boundaries. For the MOSE method, the modal parameters were determined in less points and for lower model orders for any mode system. With lower model orders, the model parameters can be identified more easily because less system and calculated parameters exist, and the chance of another closely space mode to be inherent is lower. Also, both criteria of convergence based upon regression and model overspecification can be used more easily to assist in identifying all model parameters.

CHAPTER 4

A PROGRAM CALLED MOSE

4.1 FORTRAN 77 Source Code

The source code for this program, called MOSE, is presented in Appendix D and three necessary input files for this program are shown in Appendix C. These three input files (*.arrays, *.scalars, and xn.dat) must be used and in the same directory path as the MOSE program. The 'x' is the project name the user inputs. The circled areas in Appendix C for each file is the only information required by MOSE. The user also is given the chance to pipe the data to the screen or to a file called *.txt. If the user inputs in the project name followed by ' .' (i.e. 'ghv .') then the program pipes all results to the file, else everything is outputted to the screen.

The flow chart for MOSE is shown in Figure 39. When developing this program two aspects of the application of the ARMA model with the MOSE method were changed. The first aspect is how many model orders to use to determine common modal parameters for each mode shape. From the results of all analyzed systems, model orders from two through four for two or less mode systems, and model orders from three and six for higher mode systems provides the best results using the MOSE method. This range of model orders was high enough to obtain an accurate damping product and good regressive convergence. This range was also low enough so that the number of points for regressive convergence was low, and instabilities during this convergence does not occur. Therefore, this range of model orders was used in the program.

The second aspect is how regressive convergence was handled for each mode

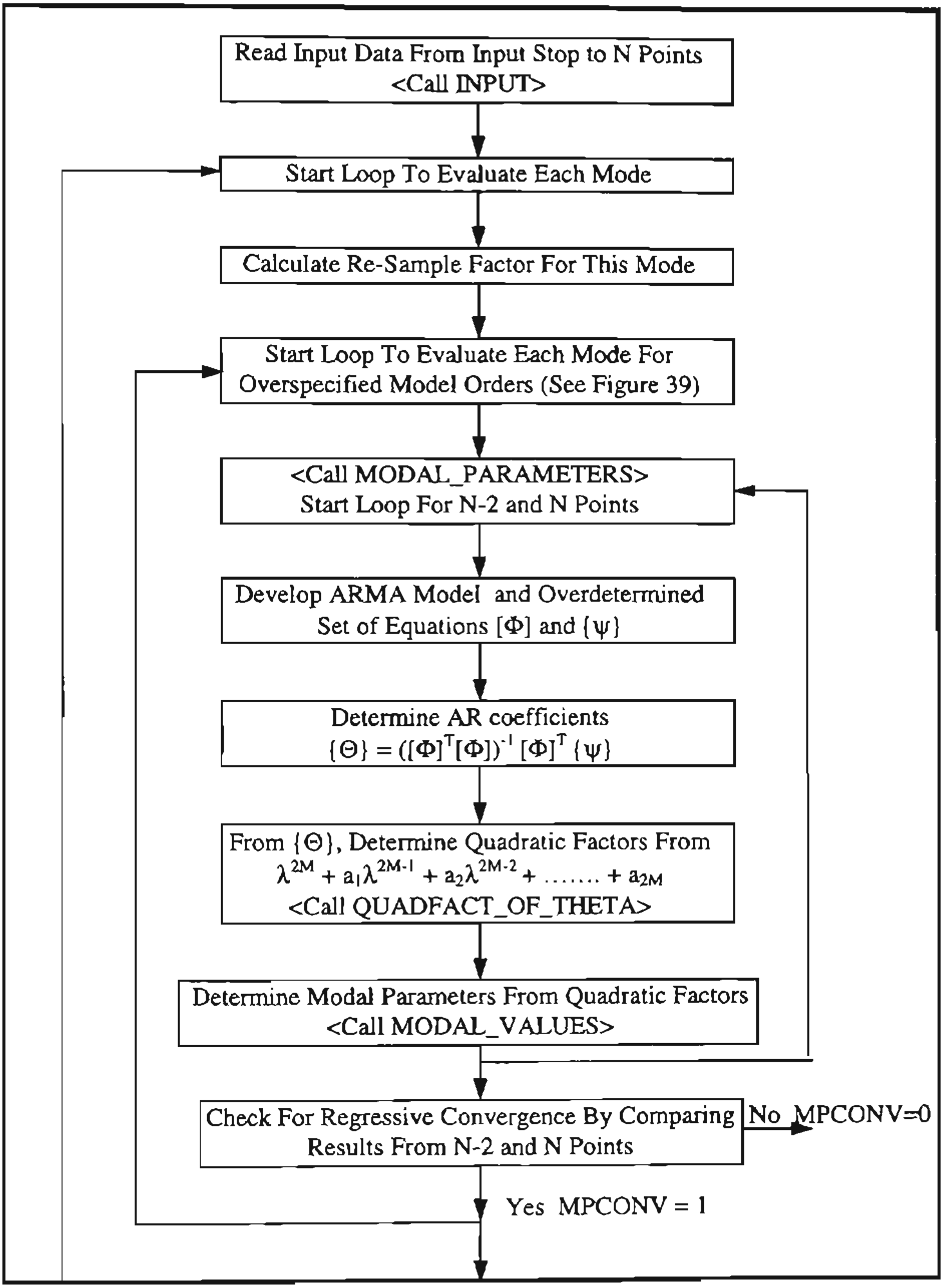


Figure 39. MOSE Flow Chart

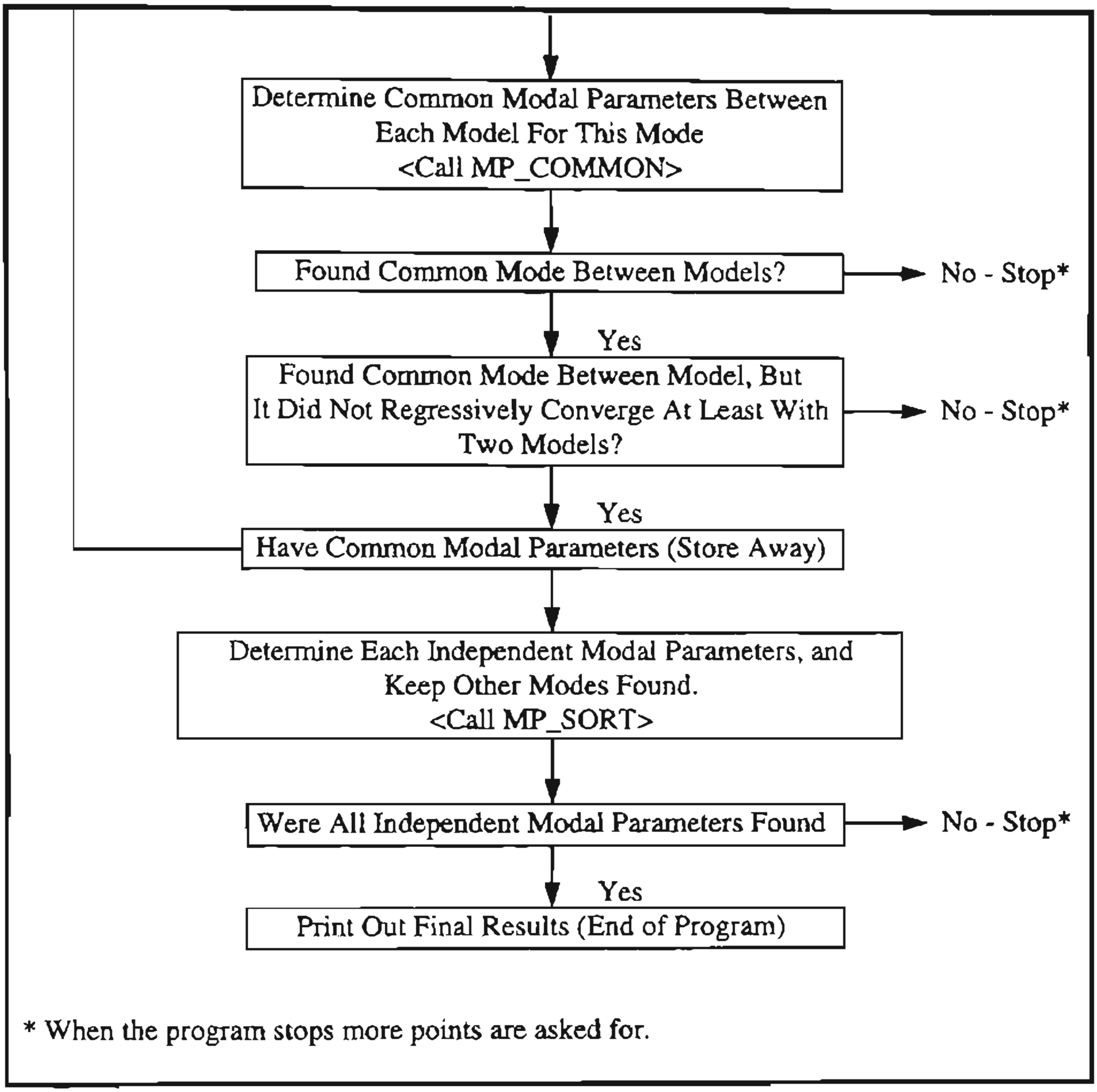


Figure 39 Continued. MOSE Flow Chart

shape. Instead of determining when regressive convergence occurs (i.e. the number of points for convergence), the program will read in ALL points for each mode shape included in the xn.dat file. The model order to use for each mode shape will depend upon the system being analyzed (Figure 40). The decision for this algorithm was based upon analyzing all systems.

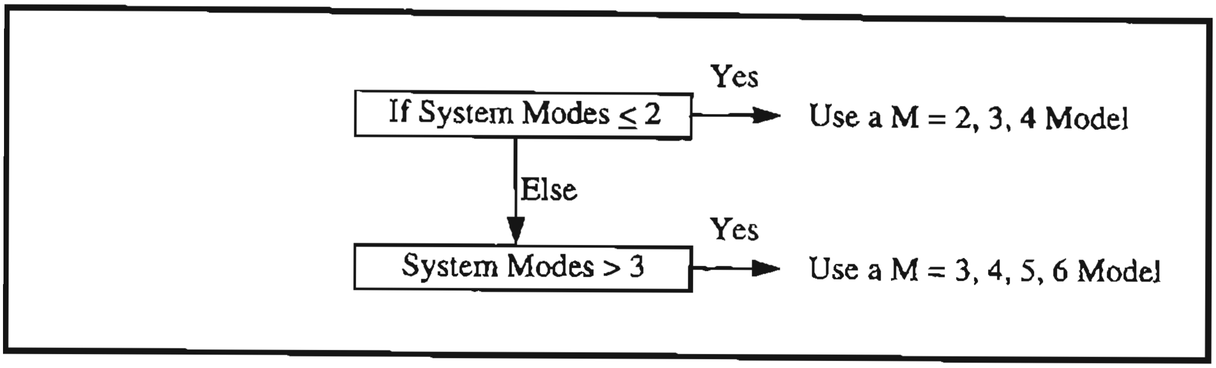


Figure 40. Algorithm for Model Order Determination

For each mode shape, the program will determine if regressive convergence has occurred by comparing the modal parameter results between N and N-2 points for each model order for all mode shapes. If regressive convergence has not occurred for that model order, a flag called MPCONV is set to 0. After the results from all three model orders have been calculated, common model parameters are determined between these models. Several sets of common modal parameters may be determined which may include system and calculated modal parameters for each independent mode shape. If common modal parameters are determined and if MPCONV is 0 at least for two different model orders, the program stops and asks for more points. If MPCONV is 1, a good result has been obtained for this mode shape and uses the results from the highest model order. All subsequent modes are handle the same way. If the program pauses and tells the user that more points may be required, therefore, the ASENL_Unsteady Code of STARS must be ran again for more data points. To avoid repeated runs for more points, it is recommended to run the ASENL_Unsteady code to develop a minimum of 4 data cycles for systems of six modes and less and a minimum of 10 data cycles for systems of more than six modes. The is based upon the lowest frequency being Mode 1.

After all common modal parameters have been determined for each mode shape, each independent mode is identified by starting from Mode 1 and determining its lowest damping frequency from all resulting frequencies. This frequency along with its damping factor becomes the modal parameter set for Mode 1. All other modes become other existing modes for that mode shape. Mode 2 is then analyzed keeping in mind the results from Mode 1. The lowest frequency of the results for Mode 2 are first determined. If this damping frequency and factor are the same as Mode 1, then it is discarded, then the next lowest frequency is determined. If this set of modal parameters does not compare to the previous mode then this set becomes the modal parameters for Mode 2 or so on until this mode has been correctly identified. Once Mode 2 has been identified, then all subsequent modes are identified in a similar matter. This method can handle cases such as the GHV system previously analyzed. For example, the frequency for Mode 9 was less than Mode 8. It can handle this because this comparison method compares both damping frequency and factor. However, two problems do occur and are discussed in Section 4.2.

After all independent modes are identified, the damping frequency in Hz and the damping factor are placed in an output file called *.txt in a section at the end of this file. An example is shown in Appendix E. This section also includes other modes that exist, but not limited to, in this mode shape.

Three other main areas of this file exists. The first main areas provides all modal parameter (system and some "calculated") identified from each model order for each mode shape using the ARMA model and MOSE method. These results are provided if the final modal parameters at the end of this file seem unreasonable for any doubt because of the two problems discussed in Section 4.3. Therefore, the system modes can be identified hands-on from this section. The second main area provides the common modal system parameters between each model found in each mode shape. This section can also be used for hands-on purposes if necessary. The final area just before the final results provides the number of modes, the number of points read in from xn.dat and used for analysis, and the dynamic pressure of the ASENL test case. Recall, the program will not determine or stop at the point of regressive convergence of the modal parameter values.

4.2 Validation of The MOSE Program

The validation of MOSE will be accomplished by comparing the results of the MOSE program with the results presented earlier for each ASENL system. Only the damped frequency and damping factor will be compared, but not the number of points for regressive convergence due to how the program was set up.

4.2.1 Two Mode System (AGARD Wing)

Table 8 shows good comparisons, the last two columns, between the results from Section 3.2.1 and the results from the MOSE program. Figure 41 provides the time history from the complete data file for each mode shape showing the actual points of regressive convergence. This point was determined using the MATHCAD templates.

Mode	Results From Section 3.2.1 Damping frequency f_d (Hz)	Damping Factor ζ	MOSE Program Results Damping frequency f_d (Hz)	Damping Factor ζ	Comparisons % Difference of f_d	Delta of ζ
1	11.428	-0.00041	11.451	-0.00041	0.20	0.00000
2	37.403	0.06455	37.475	0.06452	0.19	0.00003

Table 8. Comparison Between Section 3.2.1 Results and the MOSE Program Using The AGARD System

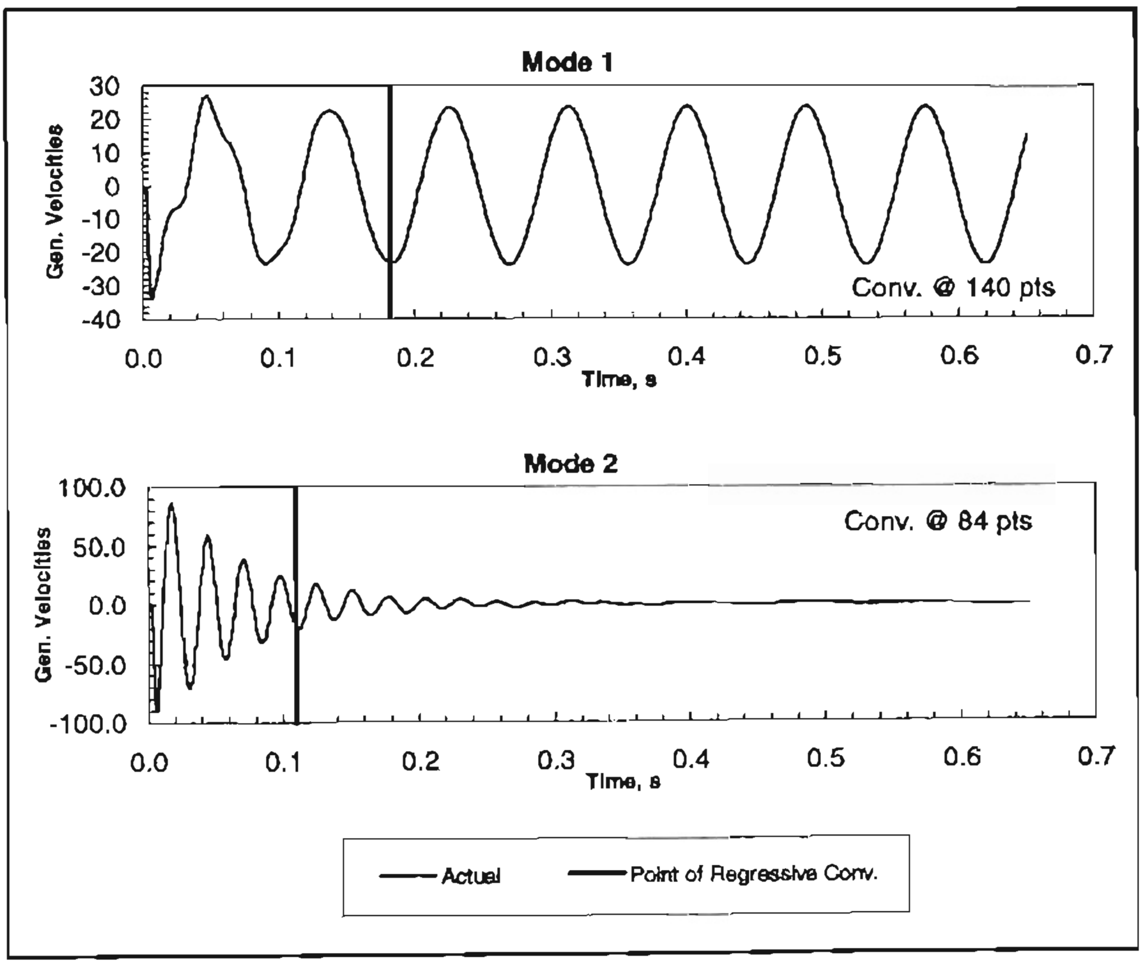


Figure 41. Generalized Velocities Vs. Time for AGARD Wing System Showing Points of Regressive Convergence with Model Order = 4

4.2.2 Six Mode System (Flat Plate)

Table 9 shows good comparisons, the last two columns, between the results from Section 3.2.2 and the results from the MOSE program. Figure 42 provides the time history from the complete data file for each mode shape showing the actual points of regressive convergence. Again, this point was determined using the MATHCAD templates.

	Results From Section 3.2.2		MOSE Program Results		Comparisons	
Mode	Damping frequency f_d (Hz)	Damping Factor ζ	Damping frequency f_d (Hz)	Damping Factor ζ	% Difference of f_d	Delta of ζ
1	3.645	0.03295	3.645	0.03295	0.00	0.00000
2	11.935	0.02809	11.934	0.02809	0.01	0.00000
3	22.017	0.02234	22.015	0.02234	0.00	0.00000
4	29.070	0.02385	29.066	0.02388	0.01	0.00003
5	44.098	0.02217	44.089	0.02216	0.00	0.00001
6	59.172	0.02117	59.154	0.02113	0.03	0.00004

Table 9. Comparison Between Section 3.2.2 Results and the MOSE Program Using The Flat Plate System

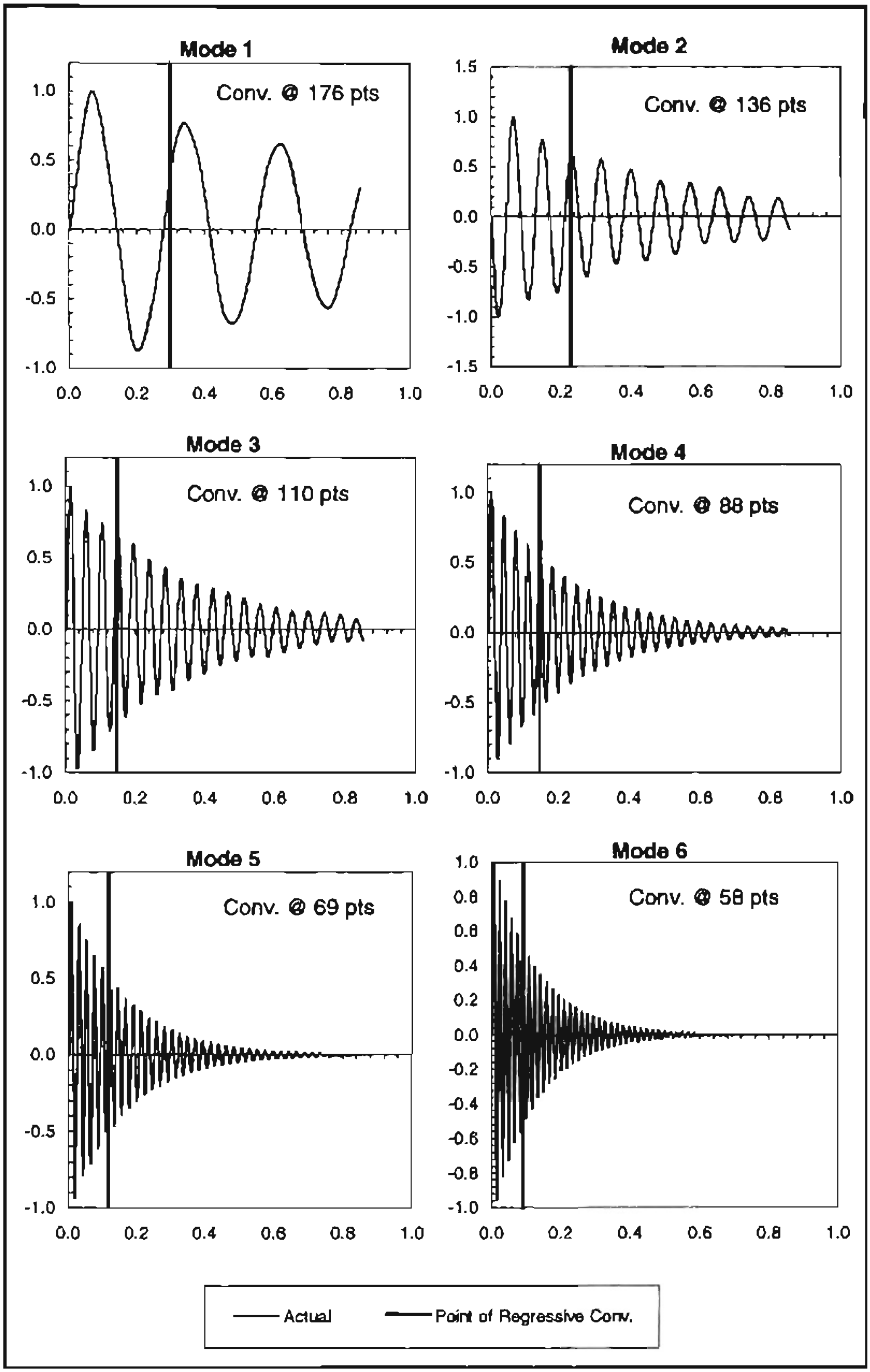


Figure 42. Generalized Velocities Vs. Time for Flat Plate System Showing Points of Regressive Convergence with Model Order = 6

4.2.3 Nine Mode System (GHV)

Finally, Table 10 shows good comparisons, the last two columns, between the results from Section 3.2.3 and the results from the MOSE program. Figure 43 provides the time history from the complete data file for each mode shape showing the actual points of regressive convergence. This point was determined using the MATHCAD templates.

	Results From Section 3.2.3		MOSE Program Results		Comparisons	
Mode	Damping frequency f_d (Hz)	Damping Factor ζ	Damping frequency f_d (Hz)	Damping Factor ζ	% Difference of f_d	Delta of ζ
1	3.659	0.08691	3.660	0.08691	0.03	0.00001
2	4.957	-0.00596	4.958	-0.00597	0.00	-0.00001
3	4.892	0.02579	4.894	0.02581	0.00	0.00002
4	5.544	0.03105	5.546	0.03097	0.00	0.00008
5	6.078	0.07846	6.080	0.07812	0.00	0.00034
6	6.092	0.07475	6.097	0.07515	0.01	-0.00040
7	7.357	0.05455	7.360	0.05447	0.00	-0.00008
8	10.383	0.02088	10.383	0.02089	0.00	-0.00001
9	10.218	0.01883	10.215	0.01906	0.00	-0.00023

Table 10. Comparison Between Section 3.2.3 Results and the MOSE Program Using The GHV System

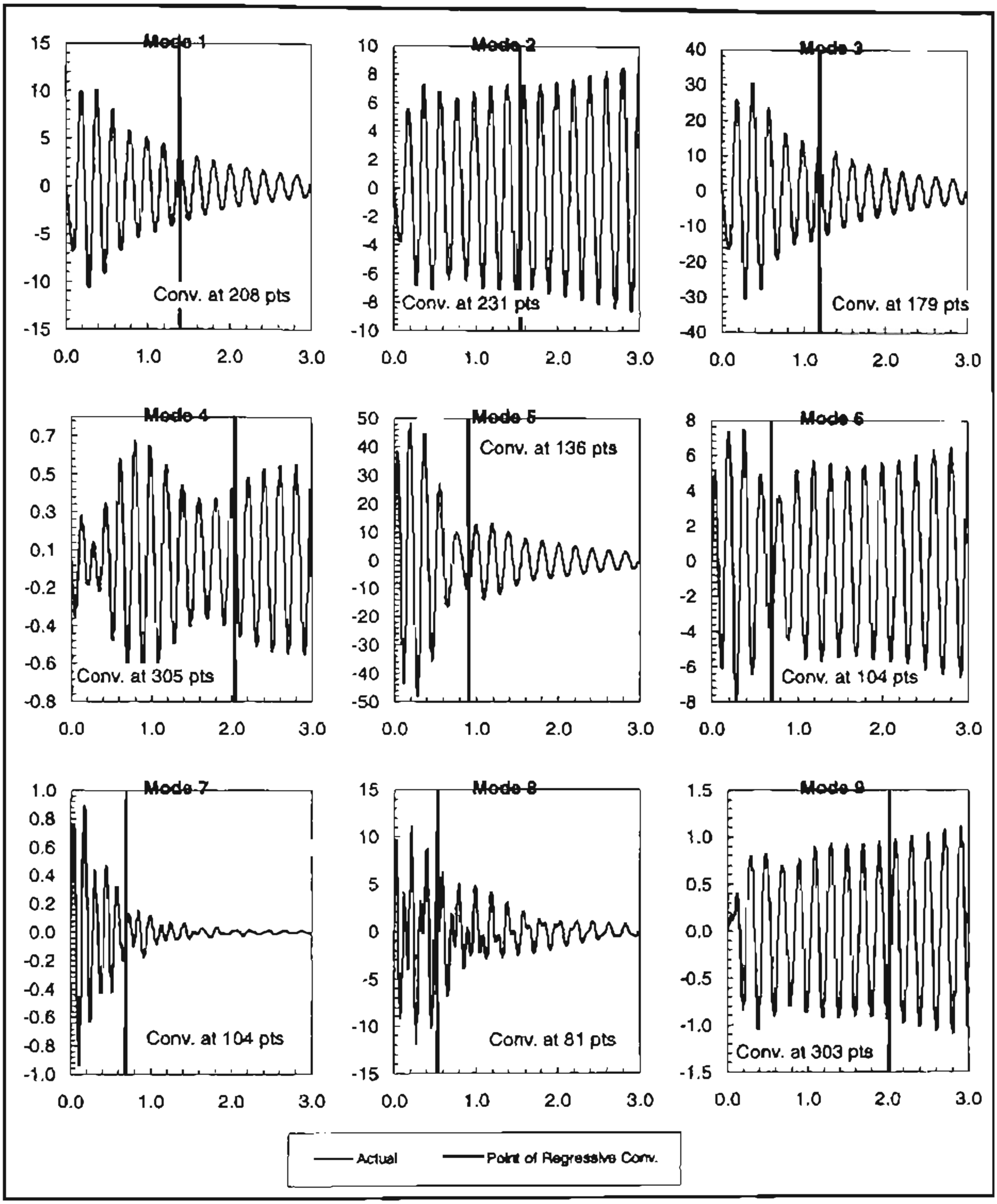


Figure 43. Generalized Velocities Vs. Time for GHV System Showing Points of Regressive Convergence with Model Order = 6

4.3 Problems with the MOSE Program

The program has been shown validated, and its results have been very useful. However, some remarks must be made about its usefulness. From analyzing several test cases from different types of system, two problems occur with MOSE.

The first problem is when two modes are identical. When two modal parameter sets are identical, the ARMA model results with model overspecification, outputted at the top of the output file, provide good results. However, the program has a difficulty in sorting these identical modes from each other to produce the final results section of the output file. The final results may be in error, and therefore, a hands-on decision using the output file more thoroughly is recommended.

The second problem occurs in the post-flutter region and only occurs for very complex systems such as the GHV model (nine modes with six modes closely spaced). In the post-flutter region of a complex system, the unstable mode becomes very dominant in each mode shape, therefore, the ARMA model fails to identify every single mode accurately. However, the final results will show that the system is unstable because the unstable mode will show up for several of the final results. Therefore, these results will have some usefulness in identifying the flutter boundary.

CHAPTER 5

AN EXAMPLE OF FLUTTER BOUNDARY PREDICTION USING THE MOSE PROGRAM

5.1 Method

This program can only be applied to one test case from the ASENL_Unsteady Code of STARS for any model. Now if one test case was accomplished and the MOSE program identified a set of system modal parameters for this case, the user would have a better understanding of how close this case was to the flutter boundary. Recall, the previous method for determining the closeness to the flutter boundary was a visual examination of the time history data. Now, a better judgment for the conditions for the next test case can made. Once several test cases have been ran, the identified modal parameters for each test case, specifically the damping factor, can be plotted against the dynamic pressure for each test case. When the damping factor is equal to zero, the flutter boundary is estimated. This method was applied to two different systems, the AGARD Wing and the GHV Systems.

5.2 AGARD Wing - Six Test Cases

5.2.1 Given Test Cases

Six general and different test cases using the AGARD system were provided. The flutter boundary was not identified until all cases were obtained. Figure 44 provides the time history data for all six test cases for Mode 1 and Mode 2. Again, the point of regressive convergence using M = 4 was determined using the MATCAD template.

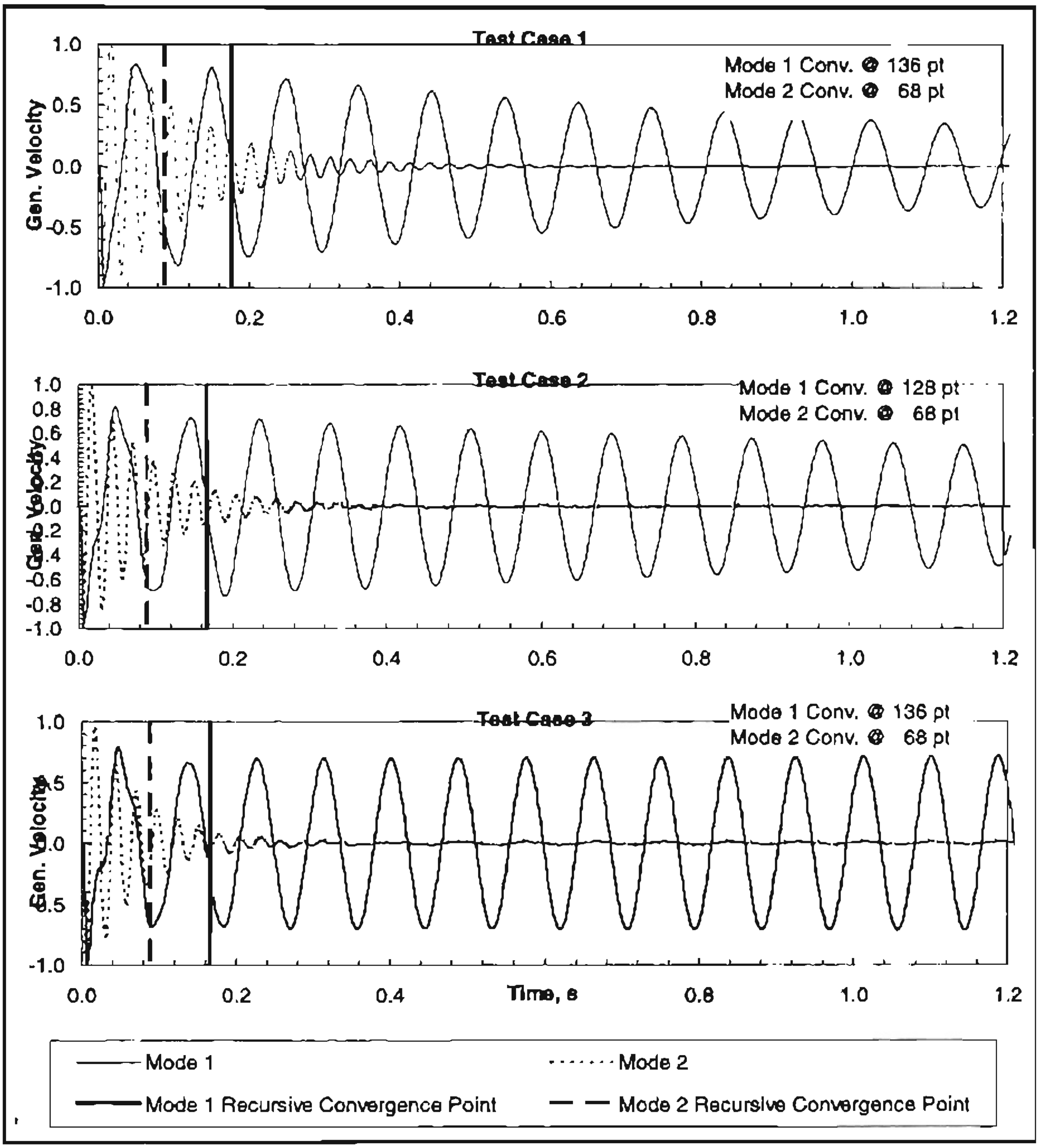


Figure 44. Normalized-Generalized Velocity Time History Plots of All Six Test Cases for AGARD Model and Showing Points of Regressive Convergence For M = 4

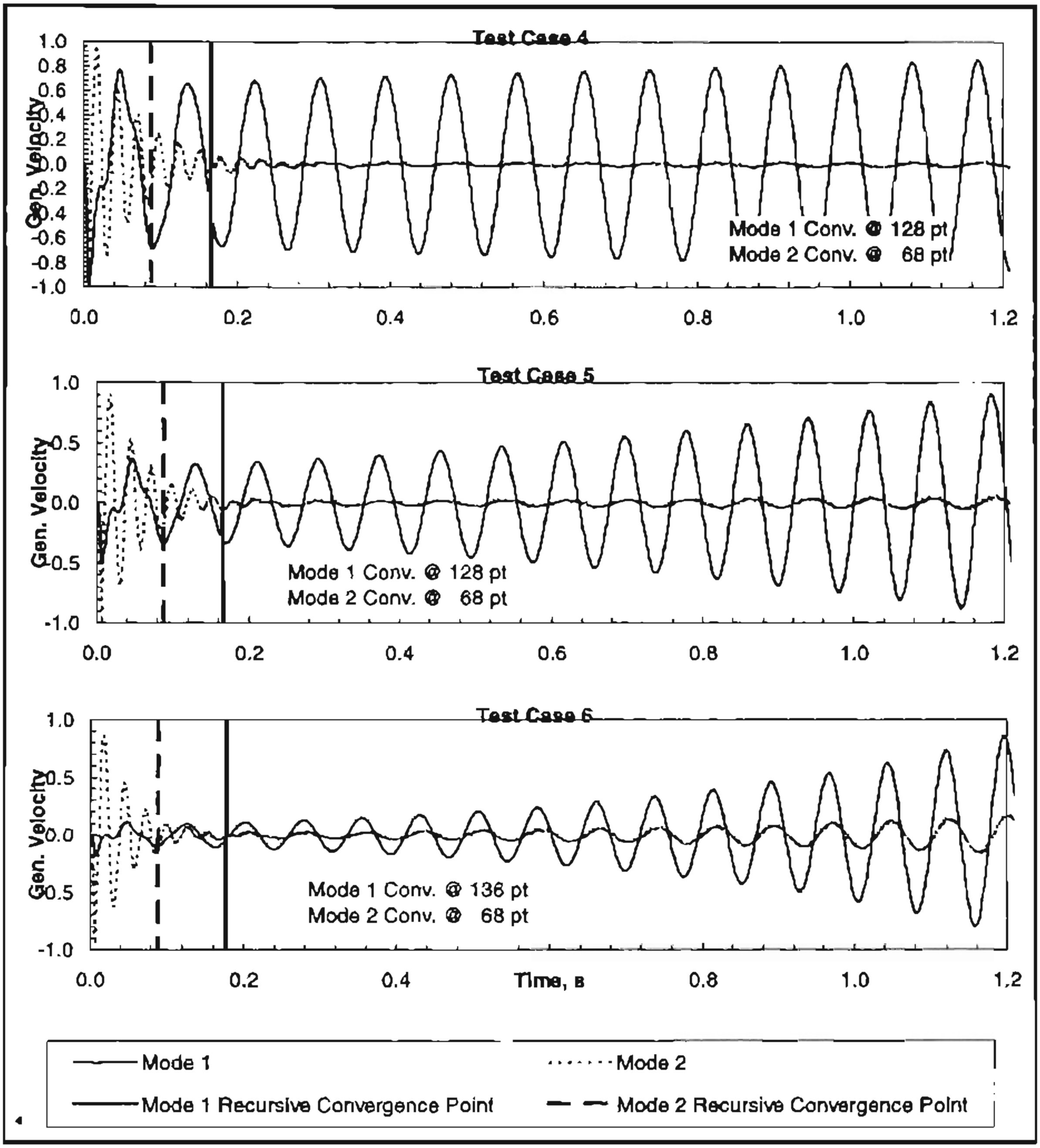


Figure 44 Continued. Normalized-Generalized Velocity Time History Plots of All Six Test Cases for AGARD Model and Showing Points Regressive Convergence For M = 4

5.2.2 Flutter Boundary Prediction for AGARD Test Cases

Table 11 provides the modal parameter results using the six different test cases. These resulting modal parameters are shown plotted against the dynamic pressure for each tests case in Figures 45 and 46 for the damping frequency and damping factor, respectively.

			Mode 1		Mode 2	
Test Case	Rho-Inf (slug/in^3)	Q (lb/ft^2)	Damped frequency f_d (Hz)	Damping Factor ζ	Damped frequency f_d (Hz)	Damping Factor ζ
0		0.0000	9.599		38.165	
1	1.04E-09	0.0283	10.313	0.01270	37.928	0.03628
2	2.00E-09	0.0543	10.967	0.00545	37.669	0.05223
3	2.70E-09	0.0733	11.451	-0.00041	37.475	0.06453
4	3.00E-09	0.0814	11.660	-0.00309	37.390	0.06998
5	4.00E-09	0.1086	12.363	-0.01299	37.107	0.08903
6	5.00E-09	0.1357	13.078	-0.02468	36.824	0.10947

Table 11. Modal Parameter Results For Six AGARD Test Cases (Including Natural Frequencies)

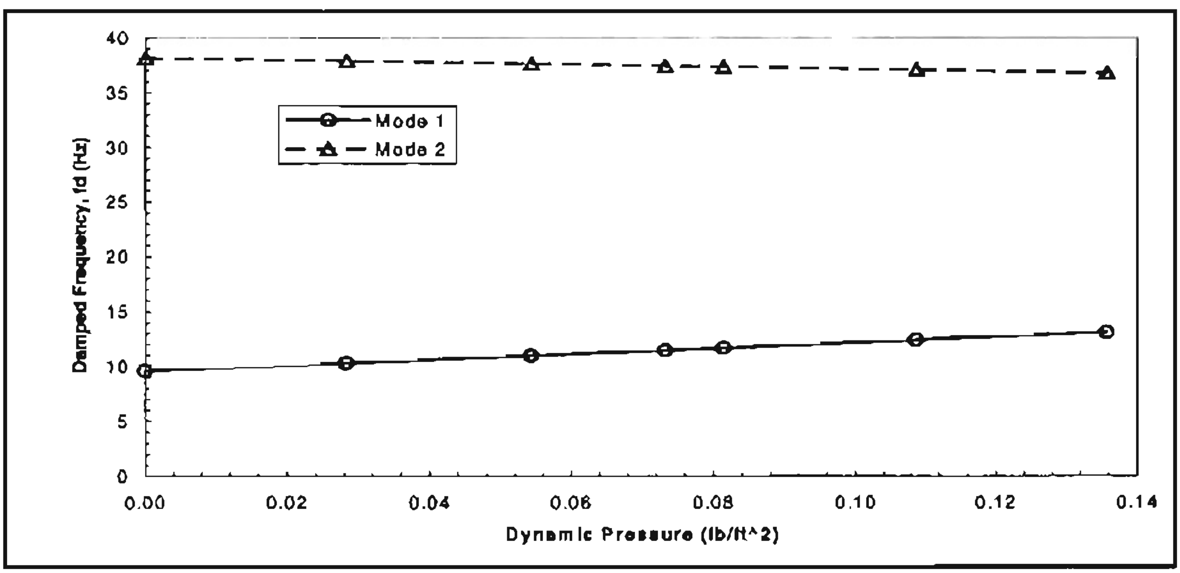


Figure 45. Resulting Damping Frequencies Vs. Dynamic Pressure For Both AGARD Modes For Six Test Cases

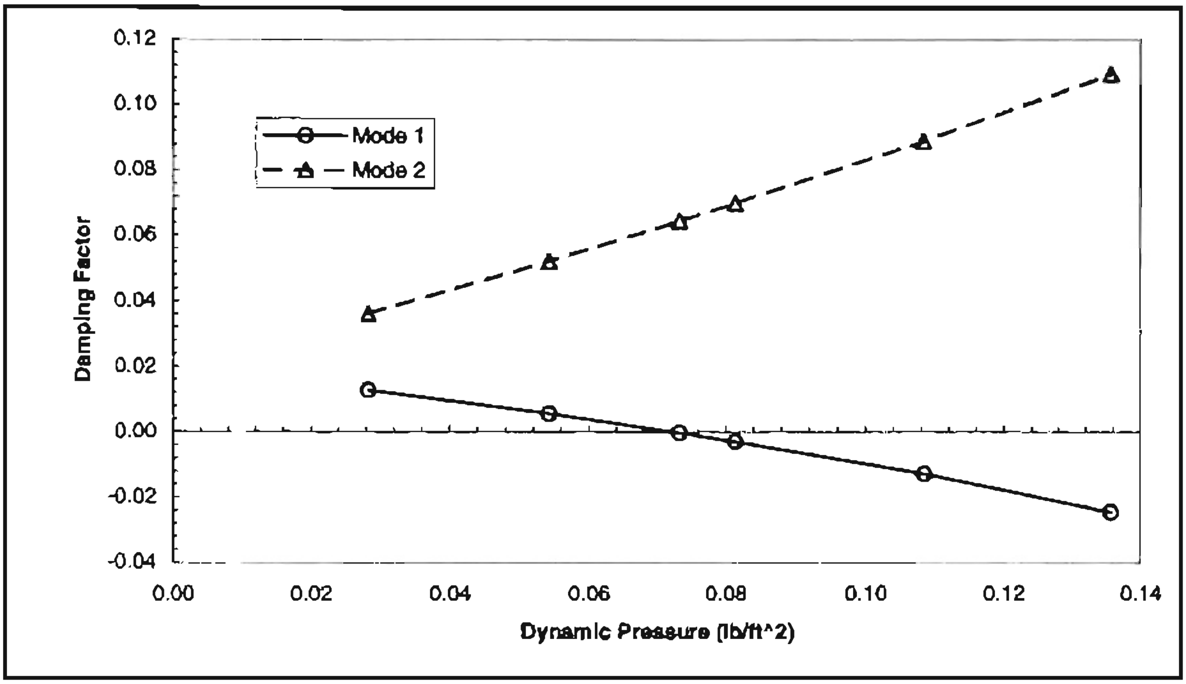


Figure 46. Resulting Damping Factors Vs. Dynamic Pressure For Both AGARD Modes For Six Test Cases

From Figure 46 the flutter boundary can be estimated using a linear interpolation between test cases two and three. Or to be more accurate, another test case can be ran for a slightly lower dynamic pressure than Test Case 3 to avoid a linear interpolation. This was not accomplished for this research. However, these results do show how a more accurate determination of the flutter boundary can be obtained using the damping factor versus the dynamic pressure, instead of using the graphical representation for visual examination, Figure 44, to determine the flutter boundary.

5.3 GHV - Eight Test Cases

5.3.1 Given Test Cases

Again, to show how a flutter boundary can be predicted using the MOSE program, a more complex system eight different test cases for the GHV system were provided. The flutter boundary was not identified until all cases were obtained. Figure 47 through 54 provides the time history data for all eight test cases for Modes 1 through Mode 9. The regressive convergence points was determined using the MATCAD template

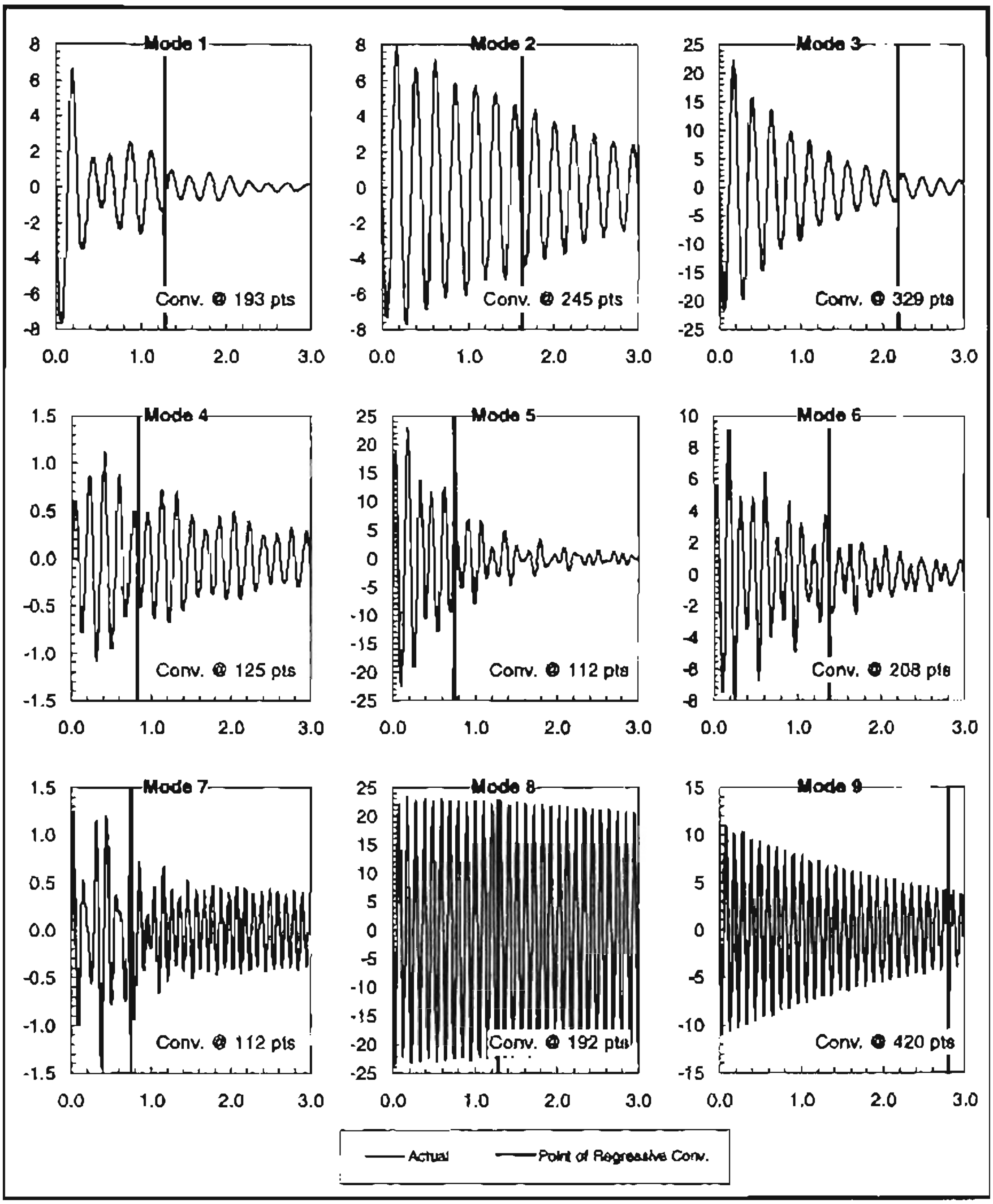


Figure 47. GHV System - Test Case 1, Dynamic Pressure is 16.3 lb/ft^2
Showing Points of Regressive Convergence From M = 6 Model

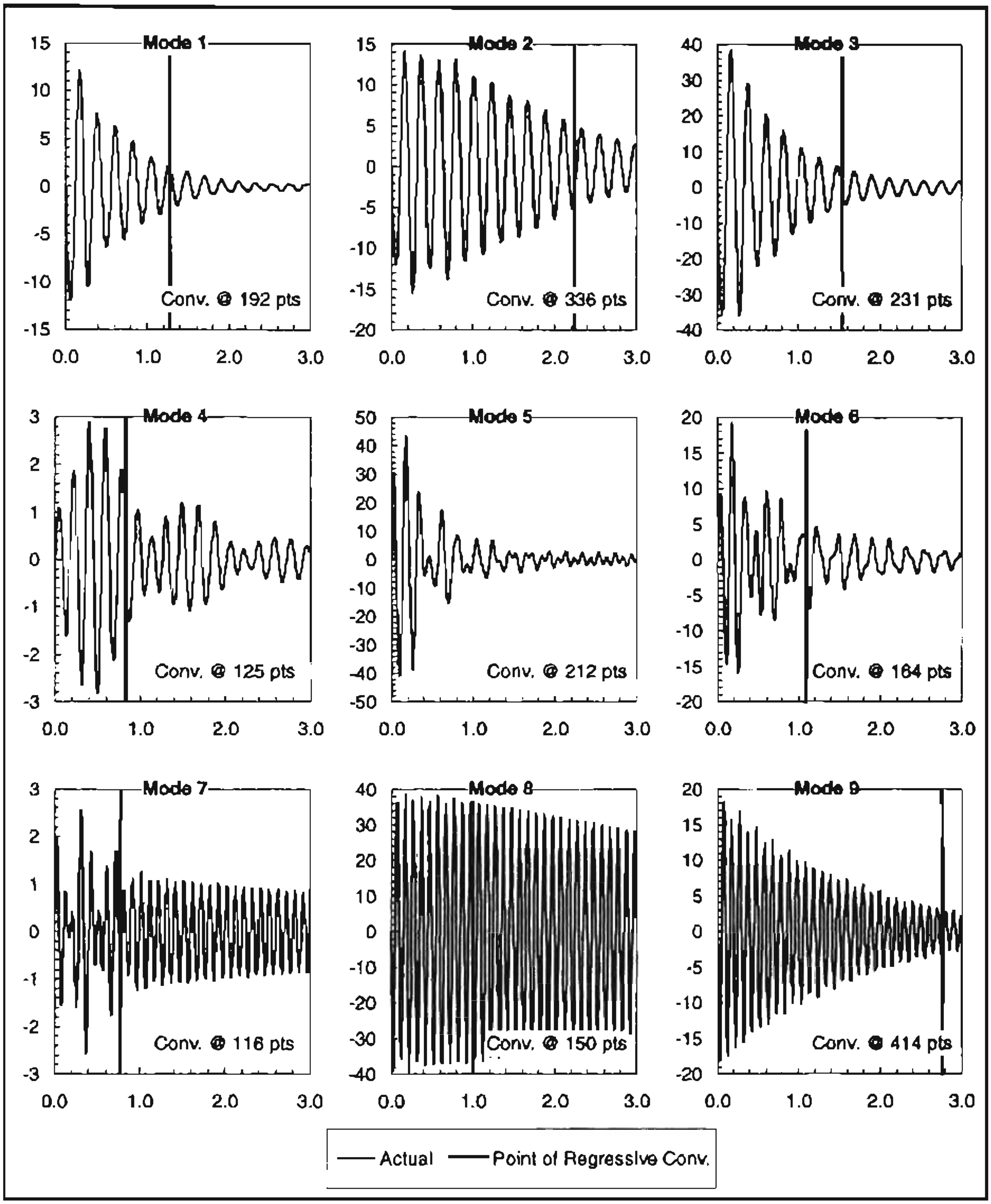


Figure 48. GHV System - Test Case 2, Dynamic Pressure is 27.7 lb/ft^2
Showing Points of Regressive Convergence From M = 6 Model

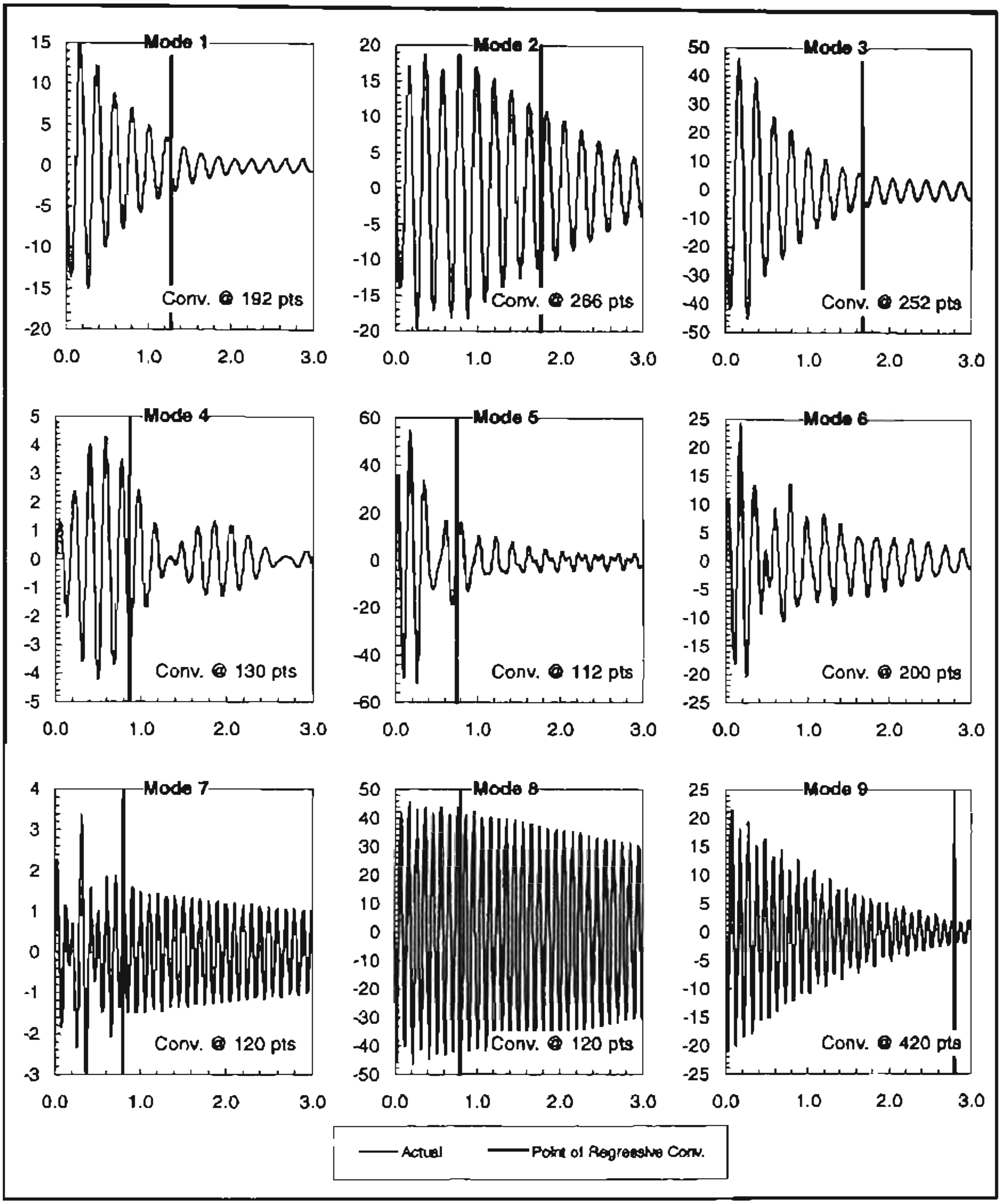


Figure 49. GHV System - Test Case 3, Dynamic Pressure is 32.6 lb/ft^2
Showing Points of Regressive Convergence From M = 6 Model

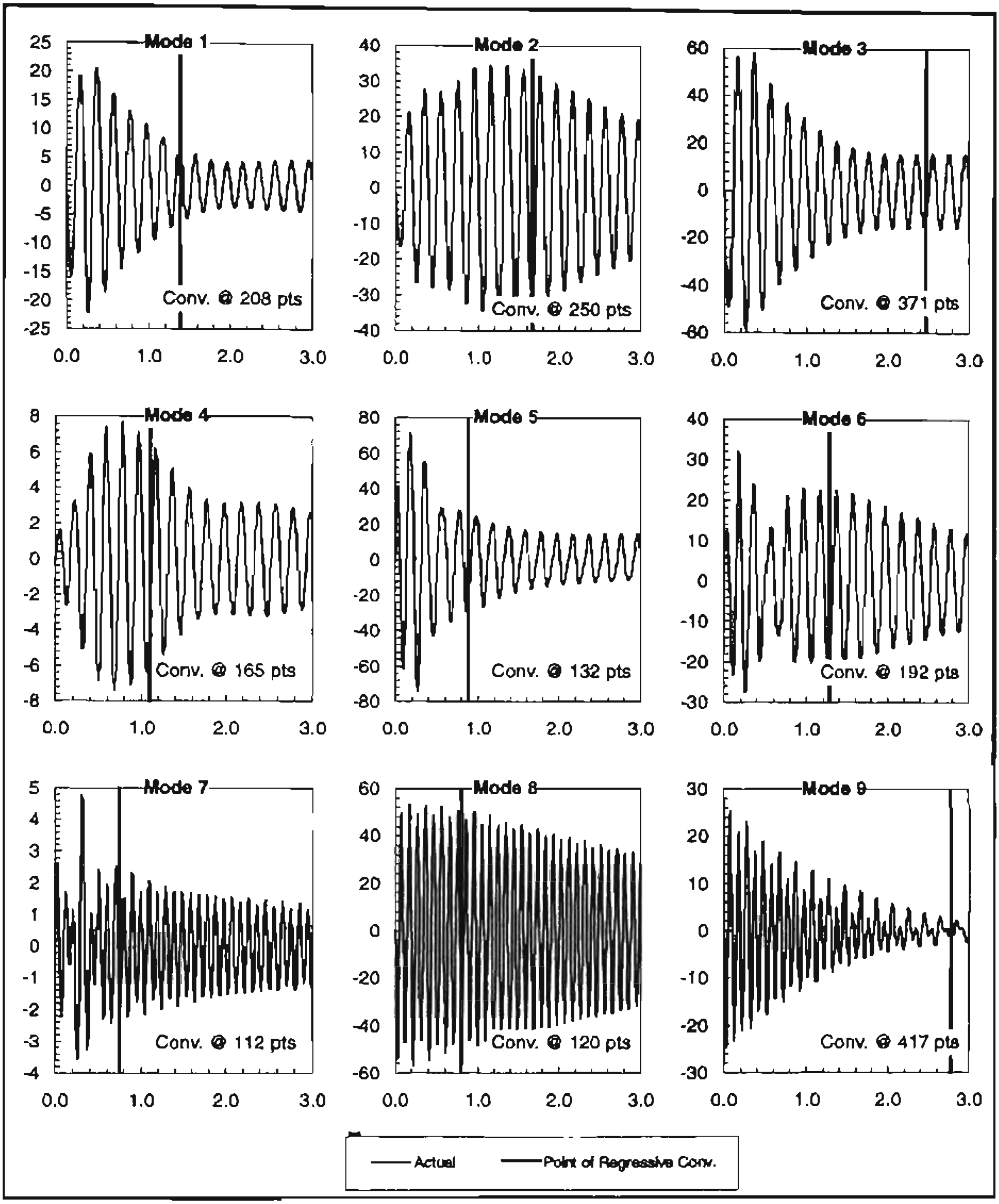


Figure 50. GHV System - Test Case 4, Dynamic Pressure is 39.1 lb/ft^2
Showing Points of Regressive Convergence From M = 6 Model

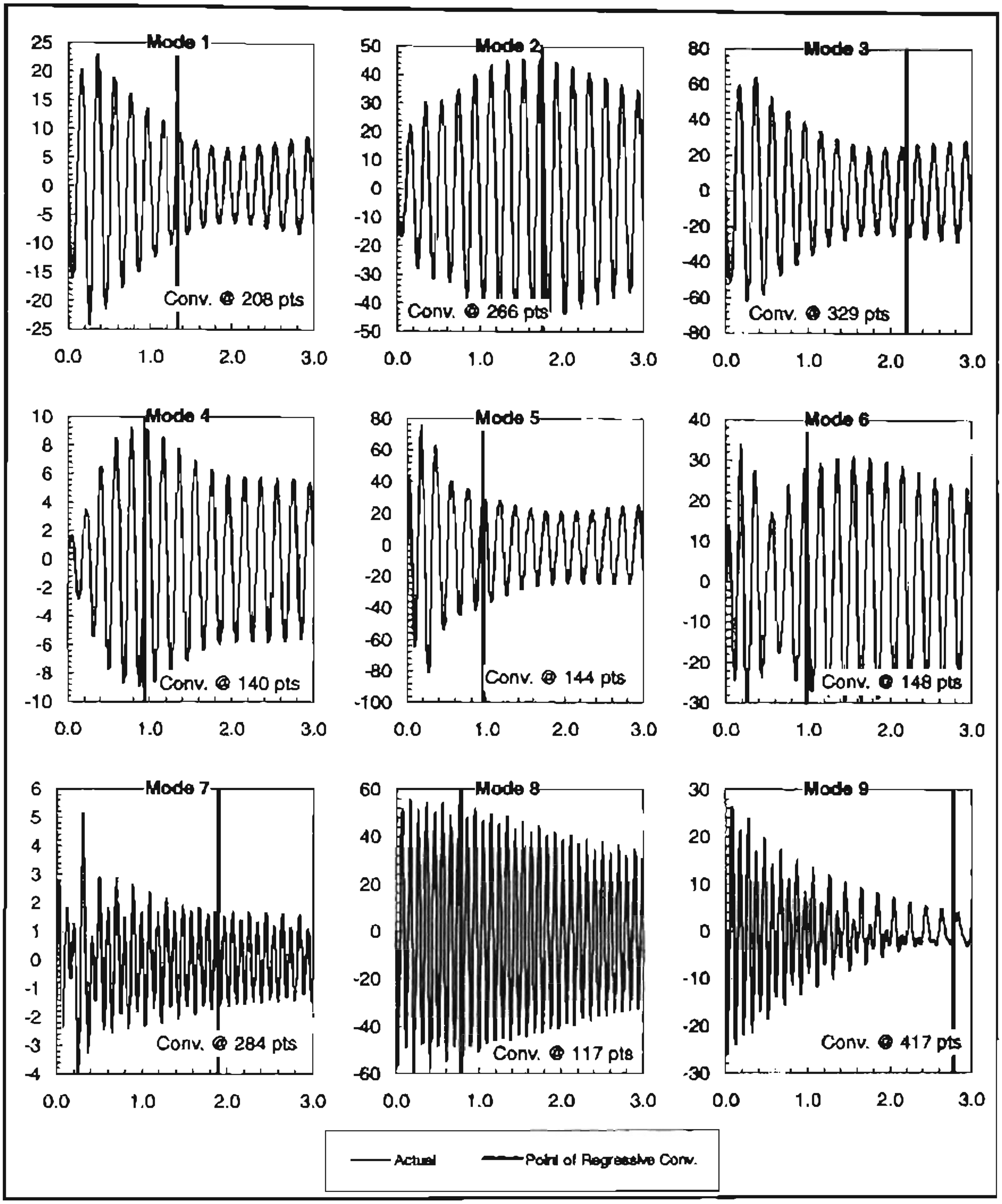


Figure 51. GHV System - Test Case 5, Dynamic Pressure is 40.7 lb/ft^2
Showing Points of Regressive Convergence From M = 6 Model

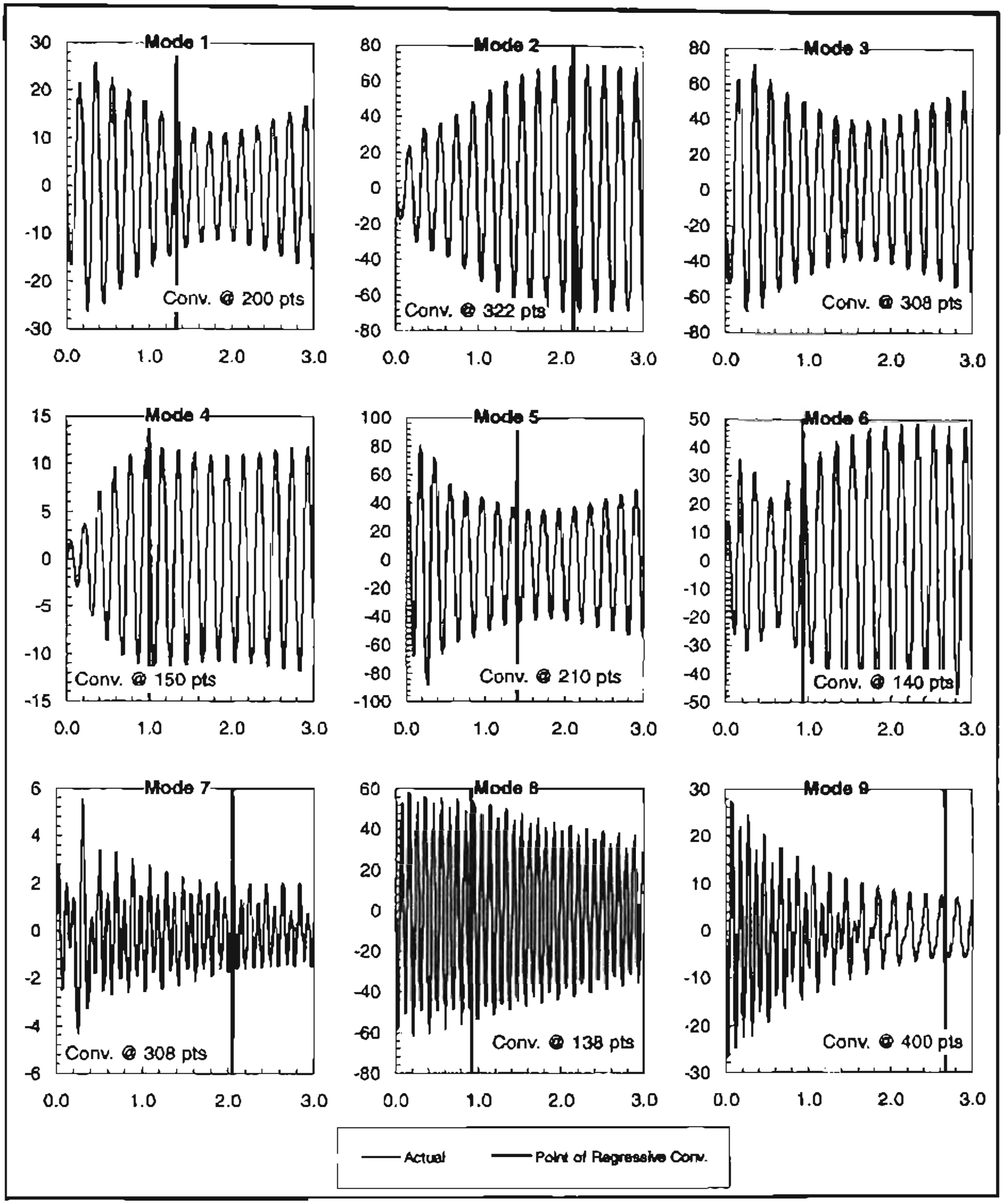


Figure 52. GHV System - Test Case 6, Dynamic Pressure is 42.3 lb/ft^2
Points of Regressive Convergence From M = 6 Model

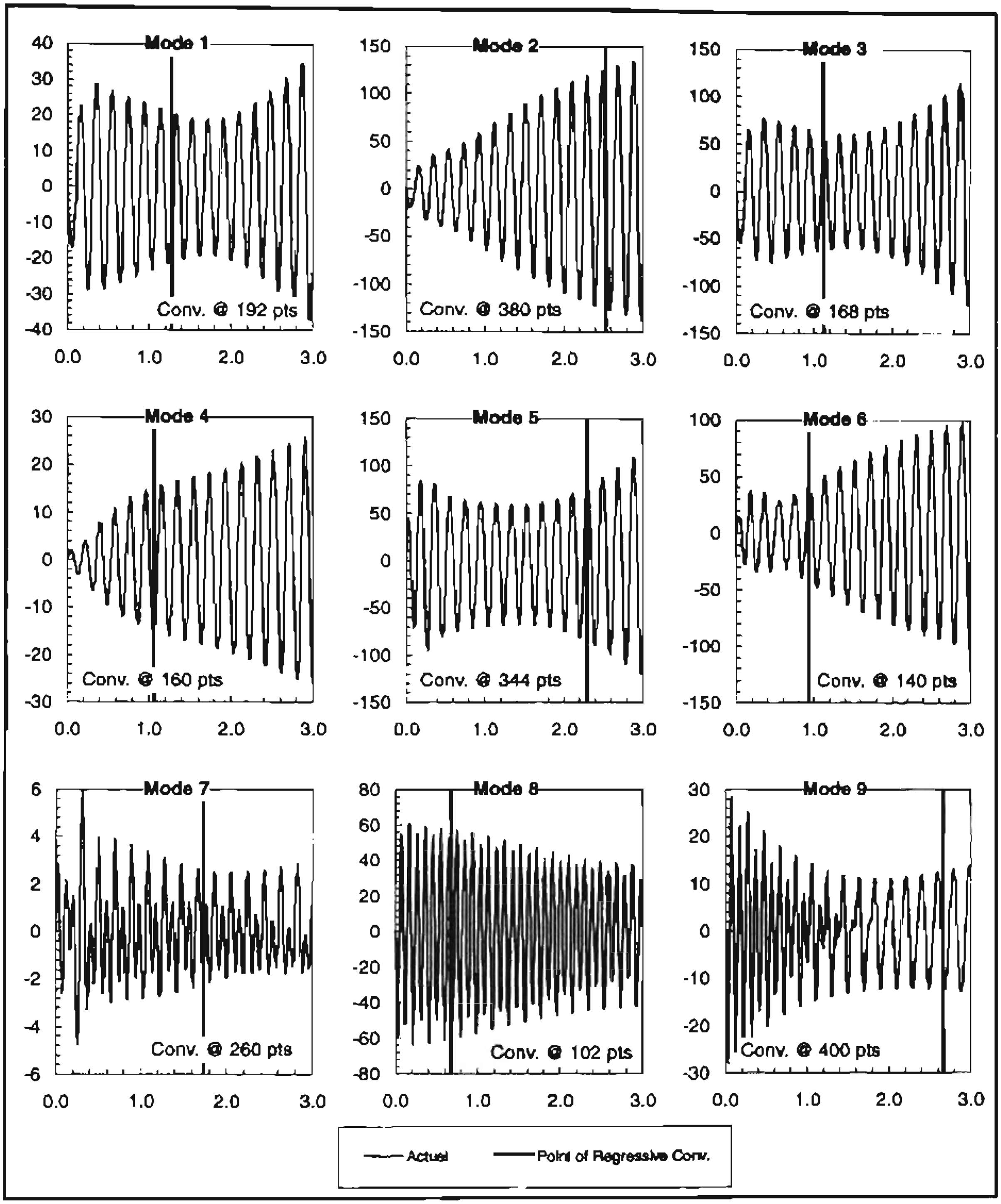


Figure 53. GHV System - Test Case 7, Dynamic Pressure is 43.9 lb/ft^2
Points of Regressive Convergence From M = 6 Model

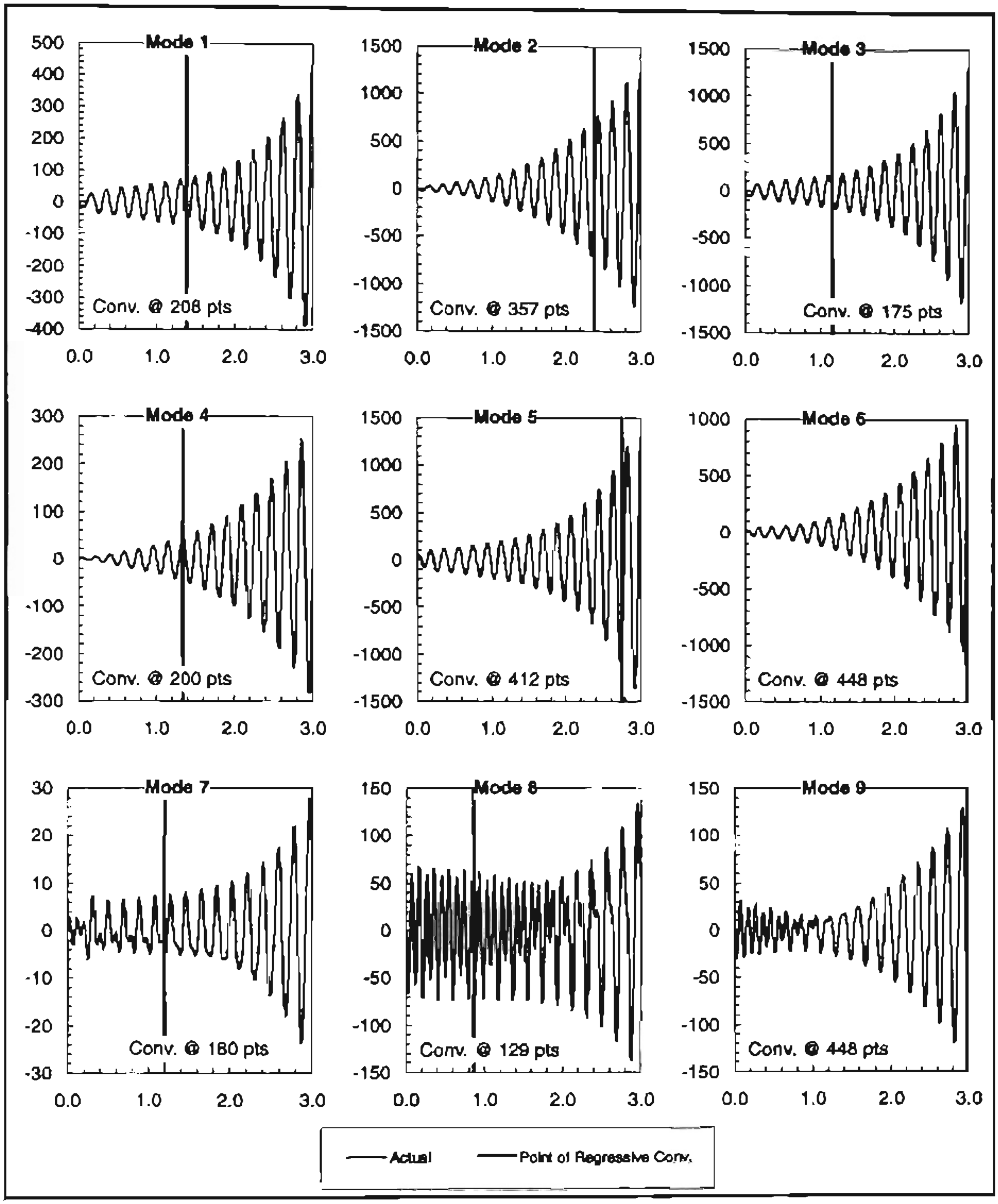


Figure 54. GHV System - Test Case 8, Dynamic Pressure is 48.9 lb/ft^2 Showing Points of Regressive Convergence From M = 6 Model

5.3.2 Flutter Boundary Prediction for GHV Test Cases

Table 12 provides the modal parameter results using the eight different test cases. These resulting modal parameters are shown plotted against the dynamic pressure for each tests case in Figures 55 and 56 for the damping frequency and damping factor, respectively.

Test Case	Q (lb/ft^2)	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7	Mode 8	Mode 9
		Damped Frequency, f_d (Hz)								
0	0.0	3.1820	4.0290	4.0520	5.5570	6.9130	6.9710	7.2670	9.4010	9.4360
1	16.3	3.2547	4.3397	4.2262	5.5407	6.8326	6.8913	7.3870	9.7628	9.7034
2	27.7	3.2959	4.6229	4.4386	5.5239	6.6943	6.8180	7.4356	10.0133	9.8837
3	32.6	3.3013	4.7661	4.5596	5.5159	6.6069	6.7694	7.4538	10.1168	9.9643
4	39.1	3.3091	4.9918	4.7576	5.5117	6.4376	6.6801	7.4454	10.2599	10.0680
5	40.7	3.3074	5.0486	4.8108	5.5155	6.3904	6.6699	7.4152	10.2873	10.0943
6	42.3	3.3040	5.1011	4.8643	5.5234	6.3444	6.6490	7.3841	10.3214	10.1201
7	44.0	3.2974	5.1483	4.9162	5.5355	6.3040	6.6328	7.3616	10.3573	10.1462
8	48.9	3.2745	5.2688	5.0478	5.5776	6.4773	6.4147	7.3522	10.4589	10.2255
		Damping Factor, ζ								
1	16.3	0.08057	0.02459	0.03593	0.01352	0.03560	0.02584	0.04535	0.00130	0.00870
2	27.7	0.15677	0.02983	0.04330	0.02427	0.07112	0.05127	0.07703	0.00234	0.01207
3	32.6	0.19412	0.02547	0.04126	0.02984	0.09111	0.06591	0.09501	0.00271	0.01414
4	39.1	0.24912	0.01121	0.02825	0.03968	0.12802	0.09349	0.11105	0.00261	0.01650
5	40.7	0.26308	0.00483	0.02306	0.04262	0.13858	0.10056	0.10805	0.00328	0.01698
6	42.3	0.27856	-0.00246	0.01674	0.04546	0.14913	0.10953	0.10331	0.00356	0.01764
7	44.0	0.29528	-0.01014	0.00930	0.04771	0.16133	0.12072	0.10138	0.00343	0.01812
8	48.9	0.34418	-0.03323	-0.01697	0.04905	0.22894	0.11715	0.10953	0.00375	0.01978

Table 12. Modal Parameter Results For Eight GHV Test Cases (Including Natural Frequencies)

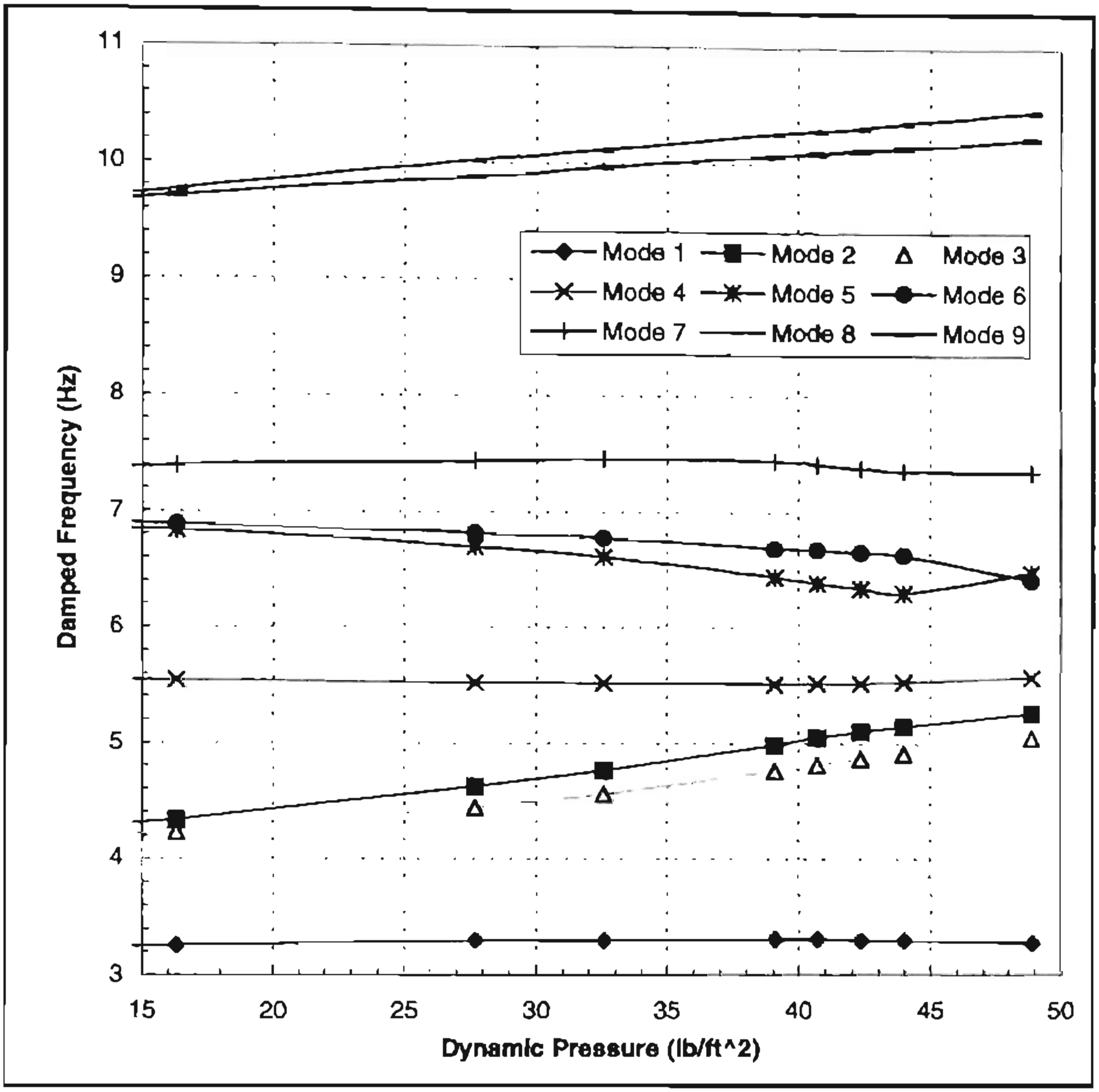


Figure 55. Resulting Damped Frequencies Vs. Dynamic Pressure For All GHV Test Cases

From Figure 56 the flutter boundary can be estimated using a linear interpolation between test cases five and six to obtain the first flutter boundary. Notice another mode goes unstable. Again, these results do show how a more accurate determination of the flutter boundary can be obtained using the damping factor versus the dynamic pressure, instead of using the graphical representation for visual examination, Figure 52, to determine the flutter boundary. In Figure 52, the flutter boundary is not really obvious either.

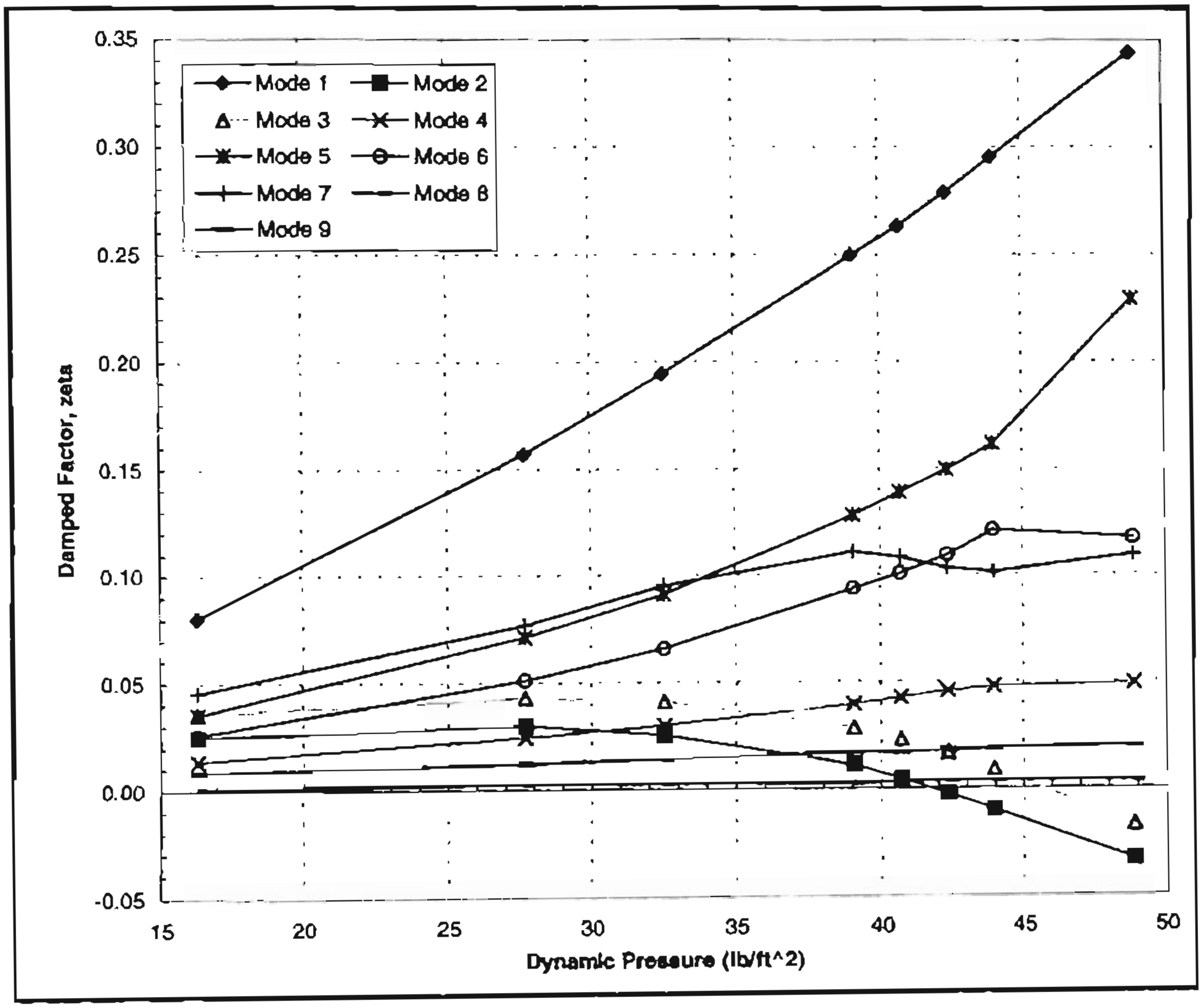


Figure 56. Resulting Damping Factor Vs. Dynamic Pressure For All GHV .... Test Cases Showing Flutter Boundary

Similar to Figure 32 which provides a matching of the input data with a sinusoidal representation with corresponding modal parameters, Figure 57 is provided below. Again, this was to ensure identification was achieved using the MOSE program. Figure 57 only provides results for Mode 2 through 4 from Test Case 6, and it provides once again a good identification was accomplished.

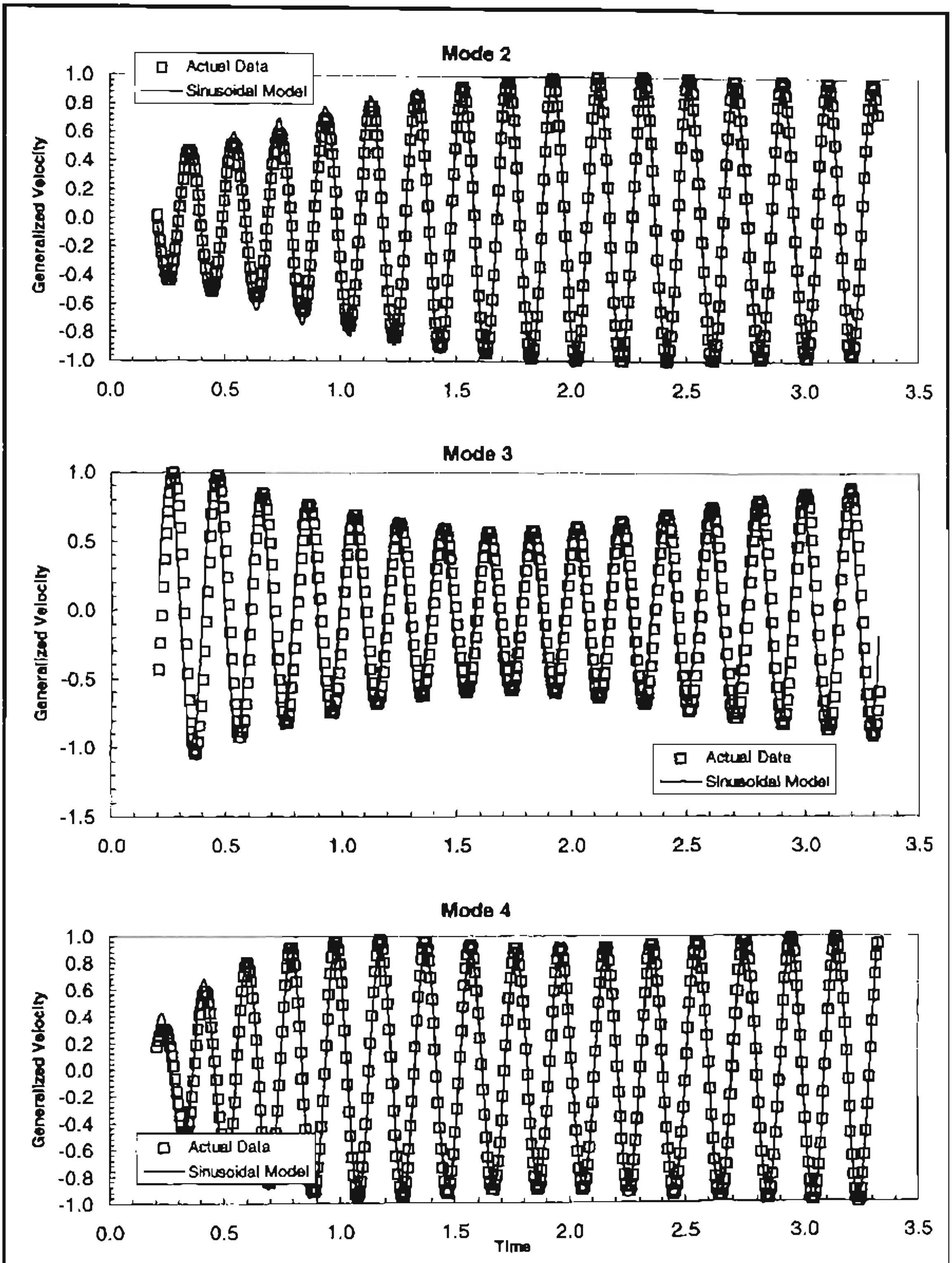


Figure 57. Modeling of Original Normalized Input Response Data From the GHV System (Test Case 6) Using Identified Modal Parameters

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS

6.1 Conclusions

Recall, the objective of this research was to replace the current method of determining the flutter boundary for any model. Specifically, a method was determined that autonomously assisted in predicting the system's flutter boundary from multiple mode time history data, specifically the results from the ASENL Module of STARS (ASENL_Unsteady Code). The following conclusions are made using the findings during this research.

1. Based upon a literature review and a simple analysis using a simulated single mode system, a simplified Single Input - Single Output Auto-Regressive Moving Average Model (2*M Auto-Regressive Coefficients and 1 MA coefficient to account for the bias due to the initial excitation) was determined to be the best method to identify system model parameters for each independent mode shape. Based upon further analysis with more highly complex, "noise-free" data systems with several closely spaced modes, the SISO ARMA model foundation was further proved to be an adequate method, and the method does not have to account for noise.
2. With the ARMA model as the foundation, three methods for determining the AR

coefficients which in turn are used to determine the finite modal parameters for each mode were investigated. These three methods being

- Method of Overdetermined Set of Simultaneous Equations (MOSE)
- On-Line Least Squares (ON-LS)
- On-Line Double Least Squares (ON-DLS)

were extensively analyzed to determine which method identified the modal parameters more efficiently, accurately, and without any convergence problems. When compared to previous methods obtained from literature, these three methods identified modes using significantly less data samples. Even though when compared between each other, all three methods provided very good results, generally, the MOSE method with an ARMA model foundation provided the best results. This was more significant in the analysis of the nine-mode GHV system with closely space modes.

3. When analyzing these three modal parameter identification methods, model overspecification was used to avoid from knowing the model order of the each mode. At first a normalized summation of all modes was considered since the model order is known in this case. However, analyzing each mode shape was more efficient based upon the difficulty with analyzing the normalized summation of all modes. Also, the accuracy of the modal identification was further improved using model overspecification with these methods. Specifically, model orders from M = 2 through 6, depending upon the system, with the MOSE method provided the best results between all methods in terms of using low data samples, the stability of

regressive convergence, and accuracy of the identification.

4. Four reasons were provided in this paper on why analyzing a normalized summation of all independent modes, especially for large mode system, was difficult. The first reason was that to identify more modes in a single data stream, more points are required for regressive convergence. The second reason was for high mode system, thus higher model orders, more data points were required to identify all modes. This also caused "switching". The third reason was choosing a good re-sample factor. No single re-sample factor can be used to efficiently identify all modes. The last reason was if all modes were identified, how can one determine which modal parameter set was belong to which mode when modal values tend to cross each other as the dynamic pressure increases.

5. When used as input, the generalized velocity data streams resulted in a better modal parameter identification. Also, re-sampling this data closer to five times the modal frequency of interest instead of using the always greater sample frequency of the data provided more accurate results with lower data samples.

6. Finally, an efficient and autonomous stand-alone program called MOSE was developed and validated against the resulting modal identifications previously made for all systems. The program showed how autonomously it could identify all modal parameters from a simple, two mode system to a highly complex system with closely space modes. Being that the program can only be applied to a single test case at a time, the flutter boundary can be applied to several different tests cases to identify the flutter boundary. This was shown, by example, from plotting the

resulting identified modes, specifically the damping factor, for each test case.

Two problems with MOSE were discussed. The program does not provide good sorted results for identical modes, and the program will not identify modes in the post-flutter region for complex system very accurately. The first problem was obvious, but the second problem was due to the fact that the unstable mode was very dominant. Overall, if the second problem is seen to occur, the general flutter boundary can still be determined because the dominant unstable mode will be shown for all modes. Therefore, these results are still useful.

6.2 Recommendations

The MOSE program could be made completely autonomous in terms of identifying the flutter boundary. This can be accomplished by further developing the program to read in multiple and different test cases then provide recommendations on the conditions to achieve the flutter test case or to exactly determine the flutter boundary.

The size of the data streams was recommended to be at least 4 data cycles for six mode systems and less and 10 or more data cycles for greater than six modes. The number of cycles is based upon the lowest frequency, and this was to only avoid repeated runs of the same test case. Still, compared to data streams used in obtained literature, these data streams at similar sample rates are significantly less.

SELECTED BIBLIOGRAPHY

Batill, S.M., Carey, C.M., & Kehoe, M.W. (1992). Digital Time Series Analysis For Flutter Test Data. AIAA Paper 92-2103, American Institute of Aeronautics and Astronautics.

Bennett, Robert M. & Desmarais, Robert N. (Oct. 1975). Curve Fitting of Aeroelastic Transient Response Data with Exponential Functions. Flutter Testing Techniques. NASA SP-415, pp. 43-58.

Bousman, W.G. & Winkler, D.J. (1981). Application of the Moving-Block Analysis. AIAA Paper 81-0653, American Institute of Aeronautics and Astronautics.

Copper, J.E. (October 1990). Identification of Time Varying Modal Parameters. Aeronautical Journal, pp. 271-278.

Dobbs, S.K. & Hodson, C.H. (June 1979). Determination of Subcritical Frequency and Damping From B-1 Flight Flutter Test Data., NASA CR 5212, Hugh L. Dryden Flight Research Center.

Dowell, Earl H. (ed.) (1995). A Modern course in Aeroelasticity. (3rd Revised). KLUWER Academic Publishers.

Gerald, Curtis F. & Wheatley, Patrick O. (1994). Applied Numerical Analysis. (5th Edition). Reading, MA: Addison Wesely Publishing Company.

Goodwind, G.C. and Sin. K.S. (1984) Adaptive Filtering Prediction and Control. Englewood Cliffs, NJ: Prentice-Hall, Inc.

Gupta, K. & Peterson, K. (December, 1991) On Some Recent Advances in Multidisciplinary Analysis of Hypersonic Vehicles. AIAA Paper 92-5026. Presented at the AIAA 4th International Aerospace Planes Conference. American Institute of Aeronautics and Astronautics.

Hammond, Charles E. & Doggett, Robert V. (Oct. 1975). Determination of Subcritical Damping by Moving Block/Randomdec Applications. Flutter Testing Techniques. NASA SP-415, pp. 59-76.

Jury, E.I. (1982). Theory and Application of the z-Transform Method. Malabar, FL: Robert E. Krieger Publishing Co.

Lenz, R.W. & McKeever, B. (Oct. 1975). Time Series Analysis in Flight Flutter Testing at the Air Force Flight Test Center: Concepts and Results . Flutter Testing Techniques. NASA SP-415, pp. 287-317.

Ljung, Lennart (1987). System Identification: Theory for the User. Upper Saddle River, New Jersey: Prentice Hall PTR.

Kehoe, M.W. (1988). Aircraft Flight flutter Testing at the NASA Ames-Dryden Flight Research Facility. California: Ames Research Center, Dryden Flight Research Facility.

Pak, Chan-Gi & Freidman, Pertz P. (1992). New Time Domain Technique For Flutter Boundary Identification. AIAA Paper 92-2102, American Institute of Aeronautics and Astronautics.

Pinkelman, J.K., Batill, S.M., Vernon, L.E., & Kehoe, M.W. (1993). An Evaluation of Excitation Techniques For Time domain Based flutter Data Processing. AIAA Paper 93-1602, American Institute of Aeronautics and Astronautics.

Pinkelman, J.K., Batill, & S.M., Kehoe, M.W. (1995). An Investigation of the Total Least Squares Criteria in Time Domain Based, Parameter Identification for Flight Flutter Testing. AIAA Paper 95-1247, American Institute of Aeronautics and Astronautics.

Press, W.H., Teukolsky, S.A., Vetterling, W.T., & Flannery, B.P. (1992). Numerical Recipes in FORTRAN 77: The Art of Scientific Computing. Cambridge, MA: Cambridge University Press.

Price, S.J. & Lee, B.H.K., (May-June 1193). Evaluation and Extension of the Flutter-Margin Method For Flight Flutter Prediction. Journal of Aircraft, Vol. 30, No. 3. pp. 395-402.

Roy, Richard & Walker, Robert (1985). Real-Time Flutter Identification, NASA CR 3933, Ames Research Center, Dryden Flight Research Facility.

Torii, Hiroshi & Matsuzaki, Yuji. (1992). Subcritical Flutter Characteristics of a Swept-Back Wing in A Turbulent Supersonic Flow. AIAA Paper 92-2393, American Institute of Aeronautics and Astronautics.

Torii, Hirsohi & Matsuzaki, Yuji. (May-June 1997). Flutter Boundary Prediction Based on Nonstationary Data Measurement. Journal of Aircraft, Vol. 34, No. 3, pp. 427-432.

Walker, Robert & Gupta, Naren (1984). Real-Time Flutter Analysis, NASA CR 170412, Ames Research Center, Dryden Flight Research Facility.

Zimmerman, N.H. & Weissenburger, J.T. (July-August 1964). Prediction of Flutter Onset Speed Based on flight Testing at Subcritical Speeds. Journal of Aircraft, Vol. 1, No. 4, pp. 190-202.

APPENDICES

APPENDIX A: EXAMPLE OF THREE METHODS DEVELOPED IN MATHCAD v6.0

Method of Overspecification of Equations:

Obtaining time history data from single mode system discussed in Section 3.2:

Read in data file and obtain time and y data vectors:

$data := READPRN(data)$ $\quad$ $tdata := data^{<1>}$ $\quad$ $ydata := data^{<2>}$

Determine length of data and vectorize data:

$N := length(tdata)$ $\quad$ $N = 63$ $\quad$ $i := 1..N$ $\quad$ $y_i := ydata_i$ $\quad$ $t_i := tdata_i$

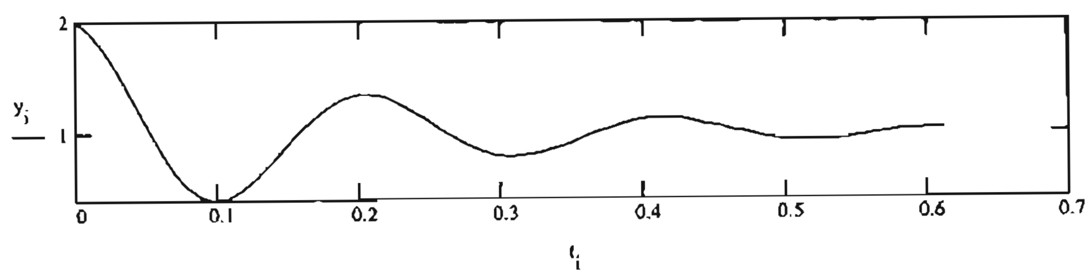
Resample data if necessary:

$n := 1$ $\quad$ $N := \frac{length(y) - 0}{n}$ $\quad$ $N = 63$ $\quad$ $i := 1..N$ $\quad$ $y_i := ydata_{i \cdot n}$ $\quad$ $t_i := tdata_{i \cdot n}$

Set last point of excitation and determine step size:

$z := 0$ $\quad$ $h := (t_N - t_{N-1})$ $\quad$ $h = 0.00986630$ $\quad$ N, here, is the number of points instead of n.

Plotting vectorized data:



Input Modal Order and determine starting point and set some initial conditions:

$M := 1$ $\quad$ $k := (2 \cdot M + 1) + z$ $\quad$ $k = 3$ $\quad$ $o := 1..2 \cdot M$ $\quad$ $Matrix_{o,o} := 0$

Method of Overspecification of Equations Continued:

Algorithm for eigenvalue Determination:

$$
\mathrm{eigen}(\theta) := \left|\begin{array}{l}
\text{for } q \in 1..2\cdot M - 1 \\
\quad \left|\begin{array}{l} \mathrm{Matrix}_{q,1} \leftarrow (-\theta)_q \\ \mathrm{Matrix}_{q,q+1} \leftarrow 1 \end{array}\right. \\
\mathrm{Matrix}_{2\cdot M,1} \leftarrow (-\theta)_{2\cdot M} \\
\mathrm{eig} \leftarrow \mathrm{eigenvals}(\mathrm{Matrix})
\end{array}\right.
$$

Algorithm to determine modal parameters at each data point.

$$
\mathrm{modes} := \text{for } m \in k..N - 1
$$

Set ϕ and ψ at m

$$
\left|\begin{array}{l}
\text{for } q \in 1..2\cdot M \\
\quad \phi_{1,q} \leftarrow (-y)_{k-q} \\
\phi_{1,2\cdot M+1} \leftarrow 1 \\
\psi_1 \leftarrow y_k
\end{array}\right.
$$

Set ϕ and ψ at m+1

$$
\left|\begin{array}{l}
\text{for } n \in 1..2\cdot M \\
\quad \phi_{m-k+2,n} \leftarrow (-y)_{(m+1)-(n)} \\
\phi_{m-k+2,2\cdot M+1} \leftarrow 1 \\
\psi_{(m-k)+2} \leftarrow y_{m+1}
\end{array}\right.
$$

Determine AR Coefficients

$$\theta \leftarrow \left(\phi^{T}\cdot\phi\right)^{-1}\cdot\phi^{T}\cdot\psi$$

Determine eigenvalues

$$\mathrm{eig} \leftarrow \mathrm{eigen}(\theta)$$

Determine modal parameters

$$
\left|\begin{array}{l}
\text{for } u \in 1..M \\
\quad \sigma_{m,u} \leftarrow -\left(\dfrac{1}{2\cdot h}\cdot\ln\left(\mathrm{Re}\left(\mathrm{eig}_{2\cdot u}\right)^2 + \mathrm{Im}\left(\mathrm{eig}_{2\cdot u}\right)^2\right)\right) \\
\quad \omega_{m,u} \leftarrow \dfrac{1}{h}\cdot\mathrm{atan}\left(\dfrac{-\mathrm{Im}\left(\mathrm{eig}_{2\cdot u}\right)}{\mathrm{Re}\left(\mathrm{eig}_{2\cdot u}\right)}\right) \\
\quad r_{m,u} \leftarrow \left|\begin{array}{l}
0 \text{ if } \mathrm{Re}\left(\mathrm{eig}_{2\cdot u}\right) < 0 \\
1 \text{ if } \mathrm{Re}\left(\mathrm{eig}_{2\cdot u}\right) > 0 \\
0 \text{ if } \mathrm{Im}\left(\mathrm{eig}_{2\cdot u}\right) = 0
\end{array}\right.
\end{array}\right.
$$

Condition of modal parameters

Vector of modal parameters

$$\begin{bmatrix} \sigma \\ \omega \\ r \end{bmatrix}$$

Organizing results to obtain modal parameters at N-1

$b := N - 1$ $\quad b\cdot n = 62$ $\quad e := 1$ $\quad (N - e)\cdot n = 62$

$$
\sigma := \left|\begin{array}{l}
\text{for } i \in 1..M \\
\quad \left|\begin{array}{l} p \leftarrow \left[(\mathrm{modes})_1\right]^{\langle i\rangle} \\ \sigma_i \leftarrow p_b \end{array}\right. \\
\sigma
\end{array}\right.
\qquad
\omega := \left|\begin{array}{l}
\text{for } i \in 1..M \\
\quad \left|\begin{array}{l} p \leftarrow \left[(\mathrm{modes})_2\right]^{\langle i\rangle} \\ \omega_i \leftarrow p_b \end{array}\right. \\
\omega
\end{array}\right.
\qquad
r := \left|\begin{array}{l}
\text{for } i \in 1..M \\
\quad \left|\begin{array}{l} p \leftarrow \left[(\mathrm{modes})_3\right]^{\langle i\rangle} \\ r_i \leftarrow p_b \end{array}\right. \\
r
\end{array}\right.
$$

Method of Overspecification of Equations Continued:

Modal Parameters and conditions of each mode at N-1 $M = 1$ $N = 63$

$\sigma^T = 5$

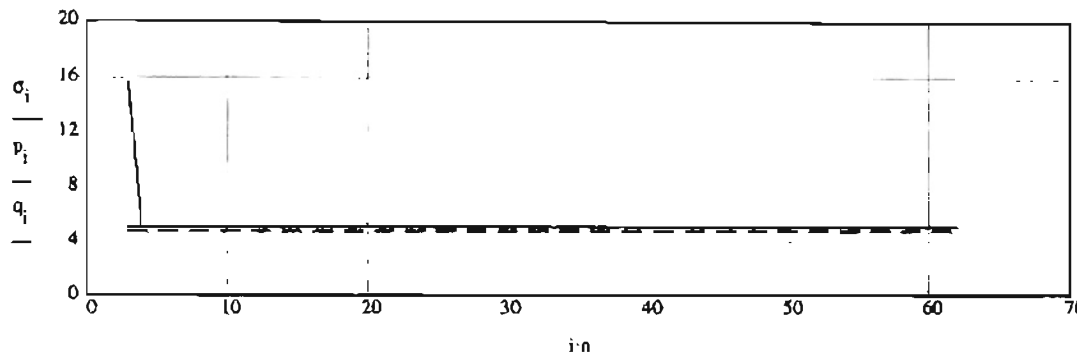
$\omega^T = 30$

$r^T = 1$

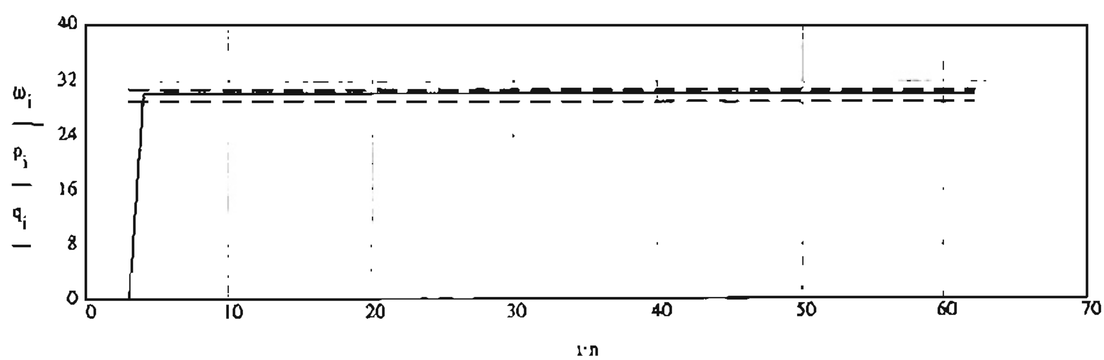
Plot of damping and frequency for each data point and each mode u.

$u := 1$ $n = 1$ $i := k..N - e$ $k = 3$ $e = 1$ $N = 63$

$\sigma := \left[(\text{modes})_1\right]^{<u>}$ 2.5% error band $p_i := (\sigma)_{N-e} \cdot .025 + (\sigma)_{N-e}$ $q_i := \sigma_{N-e} - \sigma_{N-e} \cdot .025$



$\omega := \left[(\text{modes})_2\right]^{<u>}$ 2.5% error band $p_i := (\omega)_{N-e} \cdot .025 + (\omega)_{N-e}$ $q_i := \omega_{N-e} - \omega_{N-e} \cdot .025$



End of Method of Overspecification.

On-Line Least Squares Method:

Obtaining time history data from single mode system discussed In Section 3.2:

Read in data file and obtain time and y data vectors:

$$data := READPRN(data) \qquad tdata := data^{<1>} \quad ydata := data^{<2>}$$

Determine length of data and vectorize data:

$$N := length(tdata) \quad N = 63 \qquad i := 1..N \quad y_i := ydata_i \quad t_i := tdata_i$$

Resample data if necessary:

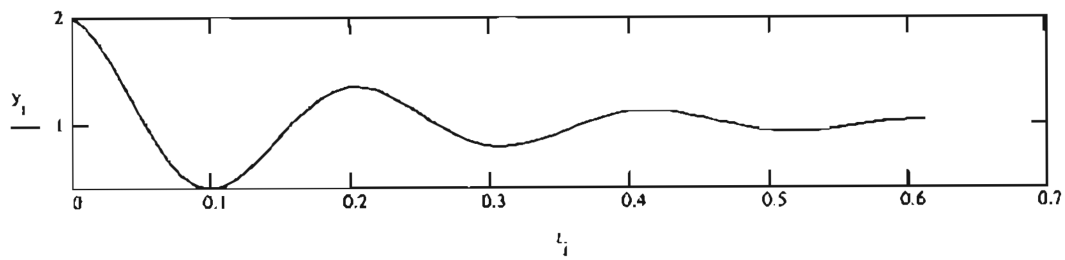
$$n := 1 \qquad N := \frac{length(y) - 0}{n} \qquad N = 63 \qquad i := 1..N \qquad y_i := ydata_{i \cdot n} \quad t_i := tdata_{i \cdot n}$$

Set last point of excitation and determine step size:

$$z := 0 \qquad h := \left(t_N - t_{N-1}\right) \qquad h = 0.00986630$$

N, here, is the number of points instead of n.

Plotting vectorized data:



Input Modal Order and determine starting point and set some initial conditions:

$$M := 1 \quad m := 2 \cdot M + 1 \qquad m = 3 \qquad k := m + z \qquad a := 10^{30} \quad I := a \cdot identity(m) \qquad \mu := 1.0$$

On-Line Least Squares Method Continued:

Algorithm for eigenvalue determination:

$$
\mathrm{eigen}(\theta) := \left|
\begin{array}{l}
a \leftarrow \theta \\
\text{for } q \in 1..2\cdot M - 1 \\
\quad \left| \begin{array}{l} \mathrm{Matrix}_{q,1} \leftarrow (a)_q \\ \mathrm{Matrix}_{q,q+1} \leftarrow 1 \end{array} \right. \\
\mathrm{Matrix}_{2\cdot M,1} \leftarrow (a)_{2\cdot M} \\
\mathrm{eig} \leftarrow \mathrm{eigenvals}(\mathrm{Matrix})
\end{array}
\right.
$$

Algorithm to determine modal parameters at each data point.

$$
\mathrm{modes} := \left|
\begin{array}{ll}
\text{for } q \in 1..m & \text{Set Initial Conditions} \\
\quad \psi_q \leftarrow y_{k-q+1} & \\
\psi_m \leftarrow 1 & \\
\text{for } r \in 1..2\cdot M + 1 & \\
\quad \theta_r \leftarrow 1 & \\
P \leftarrow a\cdot I & \\
\text{for } i \in k..N & \text{Start Loop} \\
\quad \left| \begin{array}{ll}
\theta \leftarrow \theta - P\cdot\psi\cdot\left(\psi^T\cdot P\cdot\psi + \mu\right)^{-1}\cdot\left(\psi^T\cdot\theta - y_i\right) & \text{Update AR Coefficients} \\
\mathrm{eig} \leftarrow \mathrm{eigen}(\theta) & \text{Determine eigenvalues} \\
\text{for } u \in 1..M & \\
\quad \left| \begin{array}{l}
\sigma_{i,u} \leftarrow -\left(\dfrac{1}{2\cdot h}\cdot\ln\left(\mathrm{Re}\left(\mathrm{eig}_{2\cdot u}\right)^2 + \mathrm{Im}\left(\mathrm{eig}_{2\cdot u}\right)^2\right)\right) \quad \text{Determine modal parameters} \\
\omega_{i,u} \leftarrow \dfrac{1}{h}\cdot\mathrm{atan}\left(\dfrac{-\mathrm{Im}\left(\mathrm{eig}_{2\cdot u}\right)}{\mathrm{Re}\left(\mathrm{eig}_{2\cdot u}\right)}\right) \\
r_{i,u} \leftarrow \left| \begin{array}{l} 0 \text{ if } \mathrm{Re}\left(\mathrm{eig}_{2\cdot u}\right) < 0 \\ 1 \text{ if } \mathrm{Re}\left(\mathrm{eig}_{2\cdot u}\right) > 0 \\ 0 \text{ if } \mathrm{Im}\left(\mathrm{eig}_{2\cdot u}\right) = 0 \end{array} \right. \quad \text{Condition of modal parameters}
\end{array} \right. & \\
P \leftarrow \dfrac{P}{\mu} - \dfrac{P\cdot\psi\cdot\left(\psi^T\cdot P\cdot\psi + \mu\right)^{-1}\cdot\left(\psi^T\cdot P\right)}{\mu} & \text{Updata P and } \psi \text{ for next data point} \\
\text{for } q \in 1..m - 1 & \\
\quad \psi_q \leftarrow y_{i-q+1} & \\
\psi_m \leftarrow 1 & \\
\begin{bmatrix} \sigma \\ \omega \\ r \end{bmatrix} & \text{Vector of modal parameters}
\end{array} \right.
\end{array}
\right.
$$

Organizing results to obtain modal parameters at N-1

$b := N - 1 \qquad b\cdot n = 62 \qquad e := 1 \qquad (N - e)\cdot n = 62$

$$
\sigma := \left| \begin{array}{l} \text{for } i \in 1..M \\ \quad \left| \begin{array}{l} p \leftarrow \left[(\mathrm{modes})_1\right]^{\langle i \rangle} \\ \sigma_i \leftarrow p_b \end{array} \right. \\ \sigma \end{array} \right.
\qquad
\omega := \left| \begin{array}{l} \text{for } i \in 1..M \\ \quad \left| \begin{array}{l} p \leftarrow \left[(\mathrm{modes})_2\right]^{\langle i \rangle} \\ \omega_i \leftarrow p_b \end{array} \right. \\ \omega \end{array} \right.
\qquad
r := \left| \begin{array}{l} \text{for } i \in 1..M \\ \quad \left| \begin{array}{l} p \leftarrow \left[(\mathrm{modes})_3\right]^{\langle i \rangle} \\ r_i \leftarrow p_b \end{array} \right. \\ r \end{array} \right.
$$

On-Line Least Squares Method Continued:

Modal Parameters and conditions of each mode at N-1 $M = 1$ $N = 63$

$\sigma^T = 5$

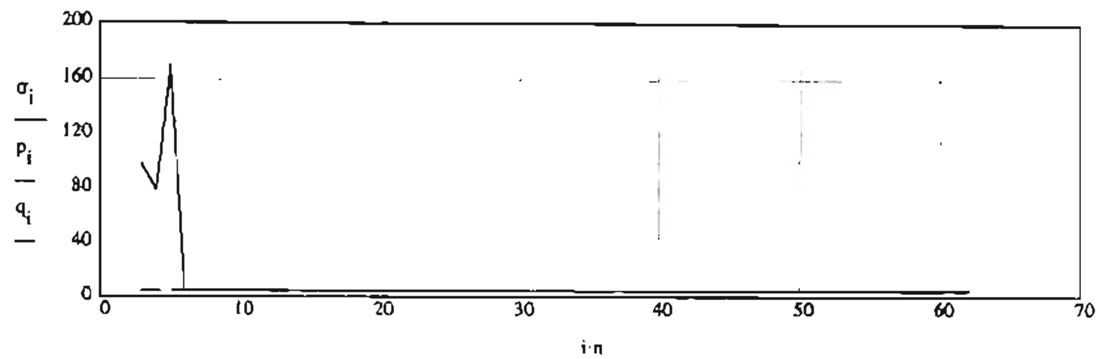
$\omega^T = 30$

$r^T = 1$

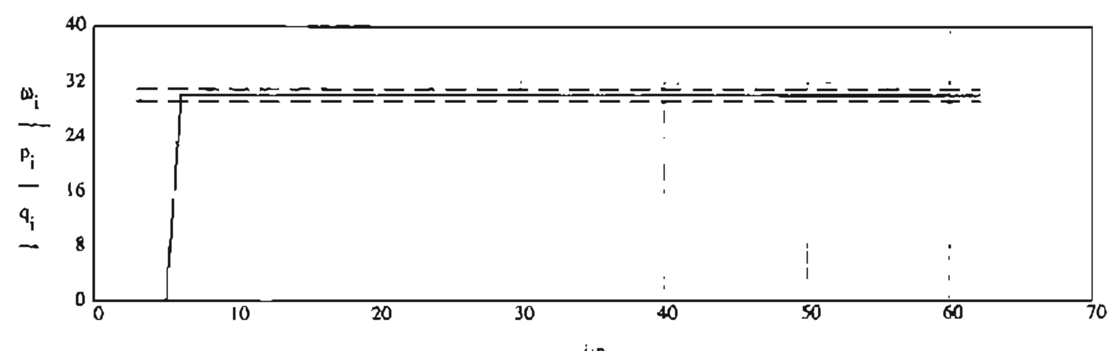
Plot of damping and frequency for each data point and each mode u.

$u := 1$ $n = 1$ $i := k..N - e$ $k = 3$ $e = 1$ $N = 63$

$\sigma := \left[(modes)_1\right]^{\langle u \rangle}$ 2.5% error band $p_i := (\sigma)_{N-e} \cdot .025 + (\sigma)_{N-e}$ $q_i := \sigma_{N-e} - \sigma_{N-e} \cdot .025$



$\omega := \left[(modes)_2\right]^{\langle u \rangle}$ 2.5% error band $p_i := (\omega)_{N-e} \cdot .025 + (\omega)_{N-e}$ $q_i := \omega_{N-e} - \omega_{N-e} \cdot .025$



End of On-Line Least Squares Method

On-Line Double Least Squares Method:

Obtaining time history data from single mode system discussed in Section 3.2:

Read in data file and obtain time and y data vectors:

$$data := READPRN(data) \qquad tdata := data^{<1>} \quad ydata := data^{<2>}$$

Determine length of data and vectorize data:

$$N := length(tdata) \quad N = 63.000 \quad i := 1..N \quad y_i := ydata_i \quad t_i := tdata_i$$

Resample data if necessary:

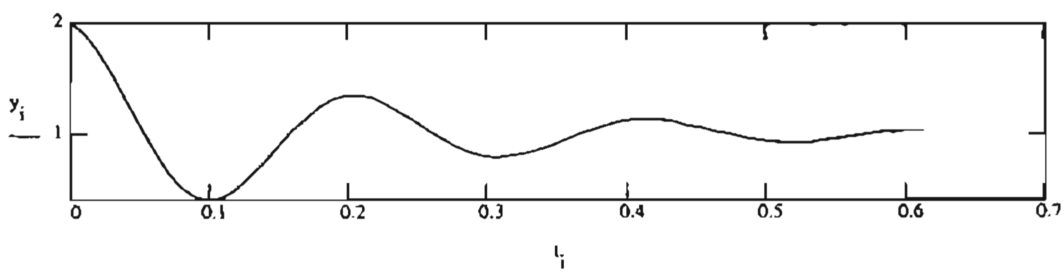
$$n := 1 \qquad N := \frac{length(y) - 0}{n} \qquad N = 63.000 \; i := 1..N \qquad y_i := ydata_{i \cdot n} \quad t_i := tdata_{i \cdot n}$$

Set last point of excitation and determine step size:

$$z := 0 \qquad h := \left(t_N - t_{N-1}\right) \qquad h = 0.00986630$$

N, here, is the number of points instead of n.

Plotting vectorized data:



Input Modal Order and determine starting point and set some initial conditions:

$$M := 1 \quad m := 2 \cdot M + 1 \qquad m = 3.000 \qquad k := m + z \qquad a := 10^{30} I \cdot identity(m) \qquad \mu := 1.0$$

On-Line Double Least Squares Method Continued:

Algorithm for eigenvalue determination:

$$
\mathrm{eigen}(\theta) := \left|\begin{array}{l}
a \leftarrow \theta \\
\text{for } q \in 1..2\cdot M - 1 \\
\quad \left|\begin{array}{l} \mathrm{Matrix}_{q,1} \leftarrow (a)_q \\ \mathrm{Matrix}_{q,q+1} \leftarrow 1 \end{array}\right. \\
\mathrm{Matrix}_{2\cdot M,1} \leftarrow (a)_{2\cdot M} \\
\mathrm{eig} \leftarrow \mathrm{eigenvals}(\mathrm{Matrix})
\end{array}\right.
$$

Algorithm to determine modal parameters at each data point.

Set Up Initial Conditions

$$
\mathrm{modes} := \left|\begin{array}{l}
\text{for } q \in 1..m \\
\quad \left|\begin{array}{l} \beta_q \leftarrow y_{k-q+1} \\ \alpha_q \leftarrow y_{k-q+1} + y_{k-q+2} \end{array}\right. \\
\beta_m \leftarrow 1 \\
\alpha_m \leftarrow 2 \\
\text{for } r \in 1..2\cdot M + 1 \\
\quad \theta_r \leftarrow 1 \\
P \leftarrow a\cdot I \\
\text{for } i \in k..N - 1
\end{array}\right.
$$

Start Loop

Update AR Coefficients

$$
\theta \leftarrow \theta - P\cdot\alpha\cdot\left(\beta^T\cdot P\cdot\alpha + \mu\right)^{-1}\cdot\left(\beta^T\cdot\theta - y_i\right)
$$

Determine eigenvalues

$$
\mathrm{eig} \leftarrow \mathrm{eigen}(\theta)
$$

$$
\text{for } u \in 1..M
$$

Determine modal parameters

$$
\sigma_{i,u} \leftarrow -\left(\frac{1}{2\cdot h}\cdot\ln\left(\mathrm{Re}\left(\mathrm{eig}_{2\cdot u}\right)^2 + \mathrm{Im}\left(\mathrm{eig}_{2\cdot u}\right)^2\right)\right)
$$

$$
\omega_{i,u} \leftarrow \frac{1}{h}\cdot\mathrm{atan}\left(\frac{-\mathrm{Im}\left(\mathrm{eig}_{2\cdot u}\right)}{\mathrm{Re}\left(\mathrm{eig}_{2\cdot u}\right)}\right)
$$

Condition of modal parameters

$$
r_{i,u} \leftarrow \left|\begin{array}{l}
0 \text{ if } \mathrm{Re}\left(\mathrm{eig}_{2\cdot u}\right) < 0 \\
1 \text{ if } \mathrm{Re}\left(\mathrm{eig}_{2\cdot u}\right) > 0 \\
0 \text{ if } \mathrm{Im}\left(\mathrm{eig}_{2\cdot u}\right) = 0
\end{array}\right.
$$

Update P, α, and β for next data point

$$
P \leftarrow \frac{P}{\mu} - \frac{P\cdot\alpha\cdot\left(\beta^T\cdot P\cdot\alpha + \mu\right)^{-1}\cdot\left(\beta^T\cdot P\right)}{\mu}
$$

$$
\begin{array}{l}
\text{for } q \in 1..m - 1 \\
\quad \left|\begin{array}{l} \beta_q \leftarrow y_{i-q+1} \\ \alpha_q \leftarrow y_{i-q+1} + y_{i-q+2} \end{array}\right. \\
\beta_m \leftarrow 1 \\
\alpha_m \leftarrow 2
\end{array}
$$

Vector of Modal Parameters

$$
\begin{bmatrix} \sigma \\ \omega \\ r \end{bmatrix}
$$

On-Line Double Least Squares Method Continued:

Organizing results to obtain modal parameters at N-1

$b := N - 1 \qquad b \cdot n = 62.000 \qquad e := 1 \qquad (N - e) \cdot n = 62.000$

$\sigma := \left| \begin{array}{l} \text{for } i \in 1..M \\ \quad \left| \begin{array}{l} p \leftarrow \left[(modes)_1 \right]^{\langle i \rangle} \\ \sigma_i \leftarrow p_b \end{array} \right. \\ \sigma \end{array} \right. \qquad \omega := \left| \begin{array}{l} \text{for } i \in 1..M \\ \quad \left| \begin{array}{l} p \leftarrow \left[(modes)_2 \right]^{\langle i \rangle} \\ \omega_i \leftarrow p_b \end{array} \right. \\ \omega \end{array} \right. \qquad r := \left| \begin{array}{l} \text{for } i \in 1..M \\ \quad \left| \begin{array}{l} p \leftarrow \left[(modes)_3 \right]^{\langle i \rangle} \\ r_i \leftarrow p_b \end{array} \right. \\ r \end{array} \right.$

Modal Parameters and conditions of each mode at N-1 $\qquad M = 1.000 \qquad N = 63.000$

$\sigma^T = 5.000$

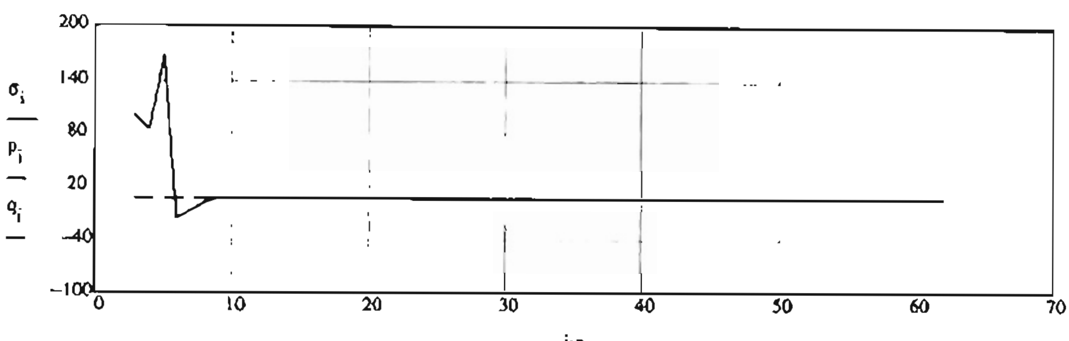
$\omega^T = 30.000$

$r^T = 1.000$

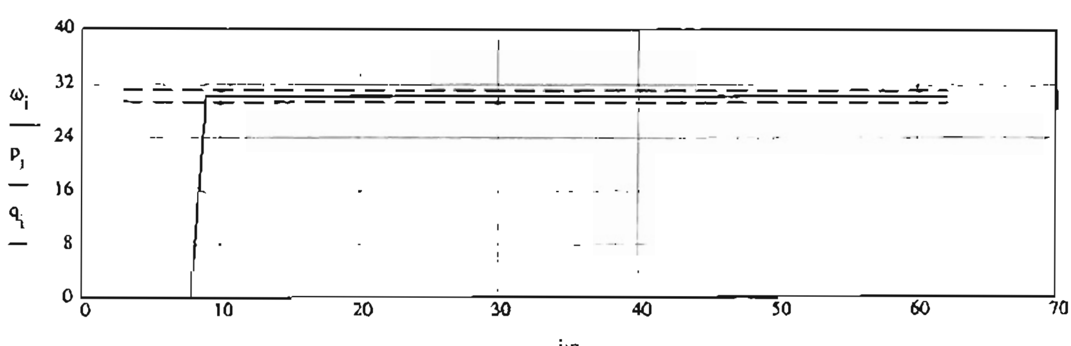
Plot of damping and frequency for each data point and each mode u.

$u := 1 \qquad n = 1.000 \qquad i := k..N - e \qquad k = 3.000 \qquad e = 1.000 \qquad N = 63.000$

$\sigma := \left[(modes)_1 \right]^{\langle u \rangle}$ 2.5% error band $\quad p_i := (\sigma)_{N-e} \cdot .025 + (\sigma)_{N-e} \quad q_i := \sigma_{N-e} - \sigma_{N-e} \cdot .025$



$\omega := \left[(modes)_2 \right]^{\langle u \rangle}$ 2.5% error band $\quad p_i := (\omega)_{N-e} \cdot .025 + (\omega)_{N-e} \quad q_i := \omega_{N-e} - \omega_{N-e} \cdot .025$



End of On-Line Double Least Squares Method

APPENDIX B: RESULTS OF TWO MODE SYSTEM (AGARD WING)

Method	Modal Order	Damping Product σ @ N=500	Damping Frequency, ω_D (rad/s) @ N=500	Points needed for conv. w/in 5% of actual σ	No Switching After This Pt
MOSE	1	.078	71.977	-	-
	2	-.038	71.858	-	-
	3	-.029	71.854	129	225
	4	-.029	71.857	200	200
		18.527	145.429	-	-
	5	-.024	71.858	-	-
		39.072	133.926	-	-
	6	-.028	71.856	-	-
		17.610	124.203	-	-
ON-LS	1	.069	72.037	-	-
	2	-.028	71.856	128	128
	3	-.028	71.856	128	255
	4	-.028	71.856	136	-
	5	-.028	71.856	152	-
	6	-.028	71.856	168	-
		13.649	107.298	-	-
ON-DLS	1	-.018	71.942	-	-
	2	-.028	71.856	120	136
	3	-.028	71.856	120	230
	4	-.028	71.856	136	-
	5	-.028	71.856	152	-
		-14.690	105.335	-	-
	6	-.028	71.856	168	-
		51.649	95.437	-	-
		30.305	139.342	-	-

Table 13. Results From Applying of Each Method Using Generalized Displacements, q, From Mode 1 of AGARD with F = 8.4 (or n = 8)

Method	Modal Order	Damping Product σ @ N=500	Damping Frequency, ω_D (rad/s) @ N=500	Points needed for conv. w/in 5% of actual σ	S After This Pt
MOSE	1	1.349	73.040	-	-
	2	-.024	71.871	360	360
	3	-.028	71.856	125	125
	4	-.027	71.858	140	140
		91.524	138.769	-	-
	5	-.028	71.855	176	260
	6	-.027	71.856	400	400
		102.587	122.822	-	-
ON-LS	1	1.246	72.540	500	500
	2	-.028	71.856	120	120
	3	-.028	71.856	120	275
	4	-.028	71.856	136	300
	5	-.028	71.856	152	-
		7.793	139.381	-	-
	6	-.028	71.856	168	-
ON-DLS	1	.546	72.094	-	-
	2	-.028	71.856	125	125
	3	-.028	71.856	128	-
		-24.025	99.976	-	-
	4	-.028	71.856	136	-
	5	-.028	71.856	152	-
	6	-.028	71.856	168	-
		-8.747	85.357	-	-

Table 14. Results From Applying of Each Method Using Mode 1 Generalized Velocities, qdot, On Mode 1 From AGARD with F = 8.4 (or n = 8)

Method	Modal Order	Damping Product σ @ N=500	Damping Frequency, ω_D (rad/s) @ N=500	Points needed for conv. w/in 5% of actual σ	No Switching After This Pt
MOSE	1	17.196	232.447	330	330
	2	-.003	70.828	-	-
		15.158	234.981	60	60
	3	-.030	71.852	-	-
		15.207	235.145	60	60
	4	-.043	71.916	-	-
		15.205	235.143	88	88
	5	-.032	71.843	-	-
		15.209	235.145	136	-
	6	-.050	71.847	-	-
		15.198	235.134	190	190
ON-LS	1	19.149	228.863	-	-
	2	-0.028	71.851	190	190
		15.205	235.145	52	52
	3	-.028	71.849	116	160
		15.206	235.145	56	160
	4	-.028	71.849	156	160
		15.206	235.145	72	160
	5	-.028	71.849	104	304
		15.205	234.145	76	400
	6	-.028	71.849	108	-
		55.83	149.814	-	-
		15.205	235.145	96	-
ON-DLS	1	38.810	233.336	-	-
	2	-.027	71.851	250	250
		15.207	235.145	64	64
	3	-.028	71.849	116	-
		15.206	235.145	56	-
	4	-.028	71.849	124	-
		15.206	235.145	68	-
	5	-.028	71.849	96	-
		-11.884	178.337	-	-
		15.206	235.145	76	-
	6	-.028	71.849	108	-
		-6.342	131.007	-	-
		15.206	235.145	84	-

Table 15. Results From Applying of Each Method Using Mode 2 Generalized Displacements, q, of (AGARD) with F = 5.6 (or n = 4)

Method	Modal Order	Damping Product σ @ N=500	Damping Frequency, ω_D (rad/s) @ N=500	Points needed for conv. w/in 5% of actual σ	No Switching After This Pt
MOSE	1	15.221	234.899	50	50
	2	-.175	72.101	-	-
		15.111	235.165	36	36
	3	-.028	71.842	-	-
		15.207	235.142	52	52
	4	-.034	71.849	-	-
		15.204	235.143	84	84
	5	-.030	71.856	-	-
		15.202	235.143	88	88
	6	-.028	71.855	-	-
		15.206	235.145	108	-
ON-LS	1	16..008	234.596	-	-
	2	-.027	71.849	330	330
		15.205	235.145	64	64
	3	-.028	71.849	164	164
		15.206	235.145	64	164
	4	-.028	71.849	136	260
		15.206	235.145	68	200
	5	-.028	71.849	92	470
		15.206	235.145	72	375
	6	-.028	71.849	128	-
		15.205	234.145	80	-
ON-DLS	1	15.614	234.312	80	80
	2	-.027	71.849	300	300
		15.205	234.145	64	64
	3	-.028	71.849	128	250
		15.206	235.145	64	250
	4	-.028	71.849	136	300
		15.205	235.145	68	260
		36.174	298.562	-	-
	5	-.028	71.849	124	-
		-42.739	206.906	-	-
		15.205	235.145	72	425
	6	-.028	71.849	124	-
		5.821	110.704	-	-
		15.206	235.145	84	-
		-26.034	283.115	-	-

Table 16. Results From Applying of Each Method Using Mode 2 Generalized Velocities, qdot, From AGARD with F = 5.6 (or n = 4)

APPENDIX C: INPUT FILES FOR MOSE.F

INPUT FILE: *.scalars

```
$ aeroelastic scalars data file ( factor=0.50 at mach=2.0 )
$ nr, ibc ( 0=full modes, 1=q(1) = 0.01, 2=q(nr+1)=0.01 )
 2, 0, 5.0
  7, 1, 2, 3, 4, 5, 7, 9
```

Number of Modes

```
$ iread, iprint
  2,   1
$ dimensional params; mach-inf, rho-inf(sl/in**3), a-inf(in/sec), gamma,  pinf
                          0.96   1.04185957e-9     12571.08    1.4     0.0
$ shift factor and gravity constant
   0.0, 1.0
$ flag,  ffi, ns, ne
   2, 10.0, 2,  7
```

Used for Qinf calculation

```
$ cfa, cfi
   1,  1
$ nterms, nsteps
    20,    2
$ na, nb
 2, 3
```

Last Point of Input becomes INPUT_STOP which is then multiplied times 3

```
 .1     .4509e-9
 .2     1.803819e-9
 .3     4.05859375e-9
 .35    5.52419705e-9
 .4     7.21527777e-9

        •
        •
        •
        •
        •
        •
        •
<End of Data File>
```

INPUT FILE: *.arrays (output file consisting of natural frequencies)

```
$ DESCRIPTION OF MODEL
$ nna, nela
 10013  20022
$ Frequencies ( in hz )
        3.182
        4.029
        4.052
        5.557
        6.913
        6.971
        7.267
        9.401
        9.436
$ Restart Option
  0
$ COMPLETE GENERALIZED STIFFNESS MATRIX
      4.38649 ,   0.0
      0.0     ,   20.7834
$ COMPLETE GENERALIZED MASS MATRIX
      1.2056e-3 ,  0.0
      0.0       ,  3.6143e-4
$ COMPLETE GENERALIZED Damping MATRIX
      0.002908949    ,  0.0
      0.0      ,  0.003467299

   •
   •
   •
   •
   •
   •
<End of Data File>
```

Natural Frequencies of Each Independent Mode

INPUT FILE: xn.dat

```
TITLE
 NCHANS    27
 NAMES
 X1
 X2
 X3
 X4
 X5
 X6
 X7
 X8
 X9
 X10
 X11
 X12
 X13
 X14
 X15
 X16
 X17
 X18
 F1
 F2
 F3
 F4
 F5
 F6
 F7
 F8
 F9
 DATA001
  .000000E+00  .000000E+00  .............  .000000E+00  .000000E+0
  .133240E-01 -.188670E-02  .............  .232247E+00  .433781E+0
  .199861E-01 -.747877E-02  .............  .442354E+00  .777978E+0
  .266481E-01 -.166700E-01  .............  .610002E+00  .963052E+0
  .333101E-01 -.294991E-01  .............  .719607E+00  .969542E+0
  .399721E-01 -.461688E-01  .............  .760892E+00  .803093E+0
•
•
•
•
•
•
  .332437E+01 -.215349E+00  ............  .652379E-02 -.338264E+0
  .333103E+01 -.220359E+00  ............  .531898E-02 -.409151E+0
```

Start of Time

q data not used

Start of qdot being y(N,1)

<End of Data File>

APPENDIX D: SOURCE CODE FOR MOSE.F

```
C ====================================================================
C THIS PROGRAM, MOSE.F, DETERMINES MODAL PARAMETERS FROM ASENL'S OUTPUT
C TIME HISTORY DATA FROM ASENL_CODE. IT USES AN ARMA MODEL TO MODEL THE
C DATA. THE AR COEFFICIENTS OF THE ARMA MODEL ARE DETERMINED USING
C SIMULTANEOUS OVERDETERMINED SET OF EQUATIONS COUPLED WITH MODEL
C OVERSPECIFICATION. THE MODAL PARAMETERS ARE DETERMINED FROM A
C CHARACTERISTIC EQUATION OF AR COEFFICIENTS. THE ROOTS OF THIS
C CHARACTERISTIC EQUATION ARE DETERMINED USING BAIRSTOW'S METHOD OF
C QUADRATIC FACTORING. THE MODAL PARAMETERS ARE THEN FOUND FROM EACH
C QUADRATIC FACTORIAL.
C
C EACH INDEPENDENT MODE SHAPE OF STARS IS ANALYZED USING THE ABOVE
C DESCRIPTION. FROM THESE RESULTS, COMMON MODAL PARAMETERS ARE
C DETERMINED. AFTER ALL MODE SHAPES ARE ANALYZED, ALL COMMON MODAL
C PARAMETERS ARE SORTED AND THE SPECIFIC MODE FOR THAT MODE SHAPE IS
C DETERMINED.
C
C THIS PROGRAM CAN HANDLE UP TO 10,000 DATA POINTS OF TIME HISTORY DATA
C AND UP TO 20 INDEPENDENT MODES.
C
C PROGRAMMER:    COREY L. ECKHART  SPRING 1998
C
C ====================================================================
C
C MAIN PROGRAM
C
C ====================================================================
C
C GLOBAL VARIABLES:
C
C NMAX               MAXIMUM NUMBER OF POINTS PROGRAM CAN HANDLE (= 10000)
C MODEMAX            MAXIMUM NUMBER OF INDEPENDENT MODE SHAPES PROGRAM
C                    CAN HANDLE (= 20)
C J                  THE COUNT OF X VALUES IN MOSE.DAT (TWICE THE NUMBER
C                    OF MODES)
C N                  NUMBER OF DATA POINTS WORKING WITH
C FLAG1              FLAG TO DETEMINE IF COMMON PARAMETERS BETWEEN EACH
C                    MODEL FOR EACH MODE WERE FOUND. IF NONE WERE FOUND
C                    PROGRAM STOPS AND MORE POINTS ARE NEEDED.
C FLAG2              ANOTHER FLAG TO DETERMINE IF ENOUGH POINTS WERE USED
C                    IN SORT SUBROUTINE FOR REGRESSIVE CONVERGENCE
C FLAG3              ANOTHER FLAG TO DETERMINE IF ENOUGH POINTS WERE USED
C                    IN SORT SUBROUTINE FOR MODE CONVERGENCE
C FLAG4              ANOTHER FLAG USED TO REQUIRE MORE POINTS IF PHIM
C                    CANNOT BE INVERTED DUE TO IT BEING SMALL AND SINGULAR
C TIME(NMAX)         ARRAY (MAX=2000) FOR THE TIME COLUMN
C Y(NMAX,MODEMAX)    MAXIMUM 2000X20 ARRAY FOR ALL QDOT DISPLACEMENT DATA
C QINF               FREE-STREAM DYNAMIC PRESSURE (0.5*RHO*(MACH*A)^2)
```

```
C FREQ(MODEMAX)     ARRAY (MAX=20) OF UNDAMPED NATURAL FREQUENCIES AT
C                   QINF=0
C INPUT_STOP        LAST POINT OF STRUCTURAL EXCITATION INPUT IN MOSER
C                   FILE
C MODES             NUMBER OF MODES IN SYSTEM
C DTORIG            ACTUAL INPUT DATA SAMPLE PERIOD (DELTA TIME),SECONDS
C MODE              MODE WHICH TO USE MODEL ON
C COUNT             COUNT HOW MANY TIMES SUBROUTINE MODAL_PARAMETERS C
                    IS USED
C RE_SAMP_FACTOR    DATA RE-SAMPLE FACTOR TO RE-SAMPLE INPUT DATA WITH
C                   TO WORK WITH HOWEVER NO LESS THAN 25/200 PTS OR
C                   RE_SAMP_FACTOR=8
C MODEL             MODEL OF MOSE TO USE
C THETA(MODEMAX)    ARRAY OF AR COEFFICIENTS
C MPR               MATRIX OF R VALUES (CONDITION OF OMEGA) FOR EACH MODE
C                   AND EVERY MODEL
C MPSIGMA           MATRIX OF DAMPING FACTORS FOR EACH MODE AND EVERY
C                   MODEL
C MPOMEGA           MATRIX OF DAMPING FREQUENCIES FOR EACH MODE AND
                    EVERY MODEL
C MPCONV            IF = 1 THEN HAVE REGRESSIVE CONVERGENCE FOR THAT
C                   MODAL VALUE
C MP(MODEMAX,4)     MATRIX OF COMMON MODAL PARAMETERS DETERMINED FROM
C                   MPR, MPSIGMA, AND MPOMEGA
C NMODE             NTH COMMON MODE
C FINAL_MODES       MATRIX OF FINAL MODAL PARAMETERS FOR ENTIRE SYSTEM
C OTHER_MODES       MATRIX OF OTHER EXISTING MODES IN EACH MODE SHAPE
C DUMMY             USED FOR MODAL PARAMETER CONVERSION
C
C LOCAL VARIABLES TO MAIN PROGRAM:
C
C NOLD              SET N TO NOLD TO BE USED LATER FOR OTHER MODES
C SAMPLE_FREQ       ORIGINAL SAMPLE FREQUENCY OF MOSER DATA
C CALIAS            USED FOR CHECKING FOR MAXIMUM RE-SAMPLE FACTOR
C PI                PI= 4.D0 *DATAN(1.D0)
C K,T               USED FOR MISCELLANEOUS DO LOOPS
C
C ==================================================================
C

      PARAMETER  (MODEMAX=20)
      PARAMETER  (NMAX=10000)

      INTEGER       J,N,FLAG1,FLAG2,FLAG3,INPUT_STOP,MODES,MODE,
     +          COUNT,RE_SAMP_FACTOR,MODEL,FILEN,SCREEN,
     +          MPCONV(MODEMAX,MODEMAX),NMODE,FLAG4,
     +          OTHER_MODES(MODEMAX,MODEMAX),NOLD,CALIAS,K,T,
     +          HIGHMODE,LOWMODE
      DOUBLE PRECISION TIME(NMAX),Y(NMAX,MODEMAX),QINF,FREQ(MODEMAX),
     +          DTORIG,THETA(2*MODEMAX+1),MPR(MODEMAX,MODEMAX),
     +          MPSIGMA(MODEMAX,MODEMAX),
     +          MPOMEGA(MODEMAX,MODEMAX),MP(MODEMAX*MODEMAX,4),
     +          FINAL_MODES(MODEMAX,MODEMAX),DUMMY,SAMPLE_FREQ,
     +          PI
      CHARACTER PROBNAME*20
```

```
C PRINTING OUT START OF PROGRAM AND ASKING FOR FILE NAME

      WRITE(*,*) '            *** Program MOSE ***'
      WRITE(*,*) ' ASENL Modal Parameter Identification Program  v1.1'

      WRITE(*,'(/,a,$)') ' Enter problem name : '
      READ(*, '(A)',ERR = 1001) PROBNAME

      J=0
      N=1
      FLAG1 = 0
      FLAG2 = 0
      FLAG3 = 0
      FLAG4 = 0
      SCREEN = 0

C CALLING SUBROUTINE TO READ INPUT DATA (TO GET ALL TIME AND Y DATA )

      CALL INPUT(TIME,Y,J,N,QINF,FREQ,INPUT_STOP,PROBNAME,FILEN,
     +          FLAG5)

C CHECK TO SEE IF RESULTS ARE PIPED TO A FILE
C SYNTAX TO PIPE TO A FILE IS TO ADD A SPACE THEN A PERIOD

      IF (PROBNAME(FILEN+2:FILEN+2) .EQ. '.') THEN
       WRITE (*,*) 'PIPING TO ',probname(1:FILEN)//'.txt'
       WRITE (*,*)
       SCREEN = 2
       OPEN(SCREEN,FILE = PROBNAME(1:FILEN)//'.txt',STATUS='UNKNOWN')
      ENDIF

      NOLD = N


C DETERMINING NUMBER OF MODES, MODEL ORDERS TO USE, AND SAMPLE
FREQUENCY

      MODES=J
      SAMPLE_FREQ = 1/(TIME(N)-TIME(N-1))
      DTORIG = 1/SAMPLE_FREQ

      IF (MODES .LE. 2 ) THEN
       LOWMODE = 1
       HIGHMODE = MODES+2
      ELSE
       LOWMODE = 1
       HIGHMODE = 6

      ENDIF

C PRINT OUT HEADER INFORMATION

      IF (SCREEN .EQ. 2) THEN
       WRITE(2,*) 'ARMA MODEL WITH MODEL OVERSPECIFICATION RESULTS',
     +          ' SHOWING MODAL PARAMETERS'
```

```
      WRITE(2,*) 'CONVERGED UPON IN EACH MODE.'
     ELSE
      WRITE(*,*) 'ARMA MODEL WITH MODEL OVERSPECIFICATION RESULTS',
    +          ' SHOWING MODAL PARAMETERS'
      WRITE(*,*) 'CONVERGED UPON IN EACH MODE.'
     ENDIF

C LOOP TO MOVE THROUGH EACH MODE

     DO 10 MODE=1,MODES
       COUNT = 0
       N = NOLD

       WRITE (*,*) 'ANALYZING MODE ',MODE

C DETERMINE RE_SAMP_FACTOR FACTOR AND CHECK IF PROPER AND NEW N

       RE_SAMP_FACTOR = SAMPLE_FREQ/(5*FREQ(MODE))

       IF ((SAMPLE_FREQ/FREQ(MODE)) .LE. 40) THEN
        CALIAS= 200/RE_SAMP_FACTOR
        IF (CALIAS .LT. 25) THEN
        RE_SAMP_FACTOR = 8
        ENDIF
       ELSE
        RE_SAMP_FACTOR = SAMPLE_FREQ/(5*FREQ(MODE))
       ENDIF


C LOOP TO DETERMINE MODAL PARAMETERS FOR EACH MODEL ON EACH MODE

       DO 20 MODEL=LOWMODE,HIGHMODE,1
       N=NOLD

C DETERMINE AR COEFFICIENTS

       CALL MODAL_PARAMETERS(THETA,Y,MODEL,MODE,N,RE_SAMP_FACTOR,
    +                  DTORIG,MPR,MPSIGMA,MPOMEGA,MPCONV,COUNT,
    +                  FLAG4)

       IF (FLAG4 .EQ. 1) THEN
       WRITE (*,*)
       PAUSE 'NEED MORE POINTS. A MATRIX CANNOT BE INVERTED, SMALL'
       STOP
       ENDIF

C END OF MODEL LOOP

   20   CONTINUE

C WRITE OUT HEADER INFO FOR EACH INDEPENDENT MODE

     IF (SCREEN .EQ. 2) THEN
       WRITE(2,*)
       WRITE(2,500) MODE,FREQ(MODE),RE_SAMP_FACTOR,SAMPLE_FREQ
     ELSE
```

```
        WRITE(*,*)
        WRITE(*,500) MODE,FREQ(MODE),RE_SAMP_FACTOR,SAMPLE_FREQ
      ENDIF

C CALL TO DETERMINE COMMON PARAMETER FOR EACH MODE BETWEEN EACH
C MODEL ORDER USED

        CALL MP_COMMON(MP,MPR,MPSIGMA,MPOMEGA,MPCONV,COUNT,MODEL,
     +               MODE,NMODE,FLAG1,FLAG2,SCREEN,LOWMODE,
     +               HIGHMODE)

C STOP IF REGRESSIVE CONVERGENCE BETWEEN N-2 AND N POINTS IS NOT FOUND

        IF (FLAG2 .EQ. 1) THEN
        WRITE(*,700) MODE
        WRITE(*,*)
        PAUSE  'HIT ENTER TO CONTINUE'
        ENDIF

C STOP IF MODE CONVERGENCE IS NOT FOUND

        IF (FLAG1 .EQ. 1) THEN
         WRITE (*,*)
         WRITE(*,*) 'WARNING! MAY NEED MORE POINTS. MODE ',
     +           'CONVERGENCE FAILED FOR MODE', MODE
         WRITE(*,*)
         PAUSE 'HIT ENTER TO CONTINUE'
        ENDIF

C END OF MODE LOOP

   10 CONTINUE

C CALL SORT TO DETERMINE FINAL MODAL RESULTS AFTER COMMON PARAMETERS
C WERE FOUND

      MODE = MODE-1

      CALL MP_SORT(MP,NMODE,MODE,FLAG3,FINAL_MODES,OTHER_MODES,SCREEN,
     +        FREQ)

      IF (FLAG3 .EQ. 1) THEN
        WRITE (*,*)
        WRITE(*,*) 'WARNING: MAY NOT BE ENOUGH POINTS, OR MODES ARE',
     +         ' VERY CLOSE TO CORRECTLY'
        WRITE(*,*) 'DEFINE PARAMETERS.  USE ABOVE LISTS TO DETERMINE',
     +          ' MODAL PARAMETERS'
        PAUSE 'HIT ENTER TO CONTINUE'
      ENDIF

C CONVERSION OF MODAL PARAMETERS TO DAMP. FREQ IN HZ AND ZETA FOR
C DAMPING FACTOR

      PI = 4.D0 *DATAN(1.D0)
```

```
      DO 30, K = 1,MODE
      IF (FINAL_MODES(K,1) .NE. 0D0 .OR. FINAL_MODES(K,2) .NE. 0D0) THEN
       DUMMY = FINAL_MODES(K,2)/(SQRT(FINAL_MODES(K,2)**2 +
     +          FINAL_MODES(K,1)**2))
       FINAL_MODES(K,2) = DUMMY
       FINAL_MODES(K,1) = FINAL_MODES(K,1)/(2*PI)
      ENDIF
   30 CONTINUE

C WRITE OUT FINAL MODAL RESULTS FOR SYSTEM TO *.TXT OR TO SCREEN

      IF (SCREEN .EQ. 2) THEN
       WRITE(2,*)
       WRITE(2,100) MODES
       WRITE(2,300) INPUT_STOP
       WRITE(2,400) NOLD
       WRITE(2,200) QINF
       WRITE(2,*)
       WRITE(2,*) 'FINAL ESTIMATED MODAL PARAMETERS'
       WRITE(2,*)
       WRITE(2,*) '     NATURAL     DAMPING      DAMPING'
       WRITE(2,*) '     FREQUENCY   FREQUENCY    FACTOR'
       WRITE(2,*) 'MODE    (HZ)        (HZ)                 ',
     +          'OTHER MODES*'
       DO 40, T=1,MODE
         WRITE(2,600) T,FREQ(T),
     +              (FINAL_MODES(T,K),K=1,2),
     +              (OTHER_MODES(T,K) ,K=1,MODE-1)
   40  CONTINUE
       IF (FLAG5 .EQ. 1) THEN
        WRITE(2,*) 'LAST MODE WAS A CONTROL MODE'
       ENDIF

      WRITE(2,602)
      WRITE(2,*)
      WRITE(2,*) 'WARNING: IF ABOVE RESULTS LOOK UNREASONABLE DUE',
     +     ' TO IDENTICAL MODES '
      WRITE(2,*) 'USE RESULTS FROM THE ARMA MODELS FOR EACH MODE',
     +        ' SHAPE. IF IN THE POST-FLUTTER '
      WRITE(2,*) 'REGION AND AN USTABLE MODE IS SEEN FOR SEVERAL MODES',
     +         ' THEN THE SYSTEM IS'
      WRITE(2,*) 'UNSTABLE AND MODES COULD NOT BE CLEARLY IDENTIFIED.'

      ELSE
       WRITE(*,*)
       WRITE(*,100) MODES
       WRITE(*,300) INPUT_STOP
       WRITE(*,400) NOLD
       WRITE(*,200) QINF
       WRITE(*,*)
       WRITE(*,*) 'FINAL ESTIMATED MODAL PARAMETERS'
       WRITE(*,*)
       WRITE(*,*) '     NATURAL     DAMPING      DAMPING'
       WRITE(*,*) '     FREQUENCY   FREQUENCY    FACTOR'
       WRITE(*,*) 'MODE    (HZ)        (HZ)                 ',
     +          'OTHER MODES*'
```

```
   DO 50, T=1,MODE
     WRITE(*,600) T,FREQ(T),
  +           (FINAL_MODES(T,K),K=1,2),
  +           (OTHER_MODES(T,K) ,K=1,MODE-1)
 50  CONTINUE


   IF (FLAG5 .EQ. 1) THEN
     WRITE(*,*) 'LAST MODE WAS A CONTROL MODE'
   ENDIF
   WRITE(*,602)
   WRITE(*,*)
   WRITE(*,*) 'WARNING: IF ABOVE RESULTS LOOK UNREASONABLE DUE',
  +      ' TO IDENTICAL MODES '
   WRITE(*,*) 'USE RESULTS FROM THE ARMA MODELS FOR EACH MODE',
  +       ' SHAPE. IF IN THE POST-FLUTTER '
   WRITE(*,*) 'REGION AND AN USTABLE MODE IS SEEN FOR SEVERAL MODES',
  +        ' THEN THE SYSTEM IS'
   WRITE(*,*) 'UNSTABLE AND MODES COULD NOT BE CLEARLY IDENTIFIED.'

   ENDIF

   WRITE (*,*)
   PAUSE 'Program MOSE Complete.  Check Data!'

   IF (SCREEN .EQ. 2) CLOSE(SCREEN)

 100 FORMAT(' NUMBER OF MODES: ',I2)
 200 FORMAT(' FREE-STREAM DYANAMIC PRESSURE (0.5*RHO*(MACH*A)^2): ',
  +        F9.5)
 300 FORMAT(' LAST POINT OF INPUT: ',I5)
 400 FORMAT(' NUMBER OF POINTS READ IN AFTER INPUT AND USED FOR ',
  +      'ANALYSIS: ',I5)
 500 FORMAT(' MODE=',I2,'  NAT. FREQ=',F10.5,'  RE-SAMPLE FACTOR=',
  +      I3,'  ORIG. SAMPLE FREQ. =',F6.1)
 600 FORMAT(I3,'  ',F10.5,'  ',F10.5,'   ',F10.6,'  ',10I3)
 602 FORMAT(' *  OTHER MODES FOUND BUT NOT LIMITED TOO.')
 700 FORMAT(' WARNING! MAY NEED MORE POINTS. REGRESSIVE CONV. ',
  +        ' FAILED FOR MODE',I2,'. CHECK ABOVE LISTS.')
 800 FORMAT(' PIPING TO',A20,'.TXT')
   STOP
1001 WRITE(*,*) ' ERROR READING PROBLEM NAME FROM SCREEN'
   END

C END OF MAIN PROGRAM
```

```
C =====================================================================
C
C START OF ALL SUBROUTINES
C
C =====================================================================


C =====================================================================
C
C SUBROUTINE INPUT: READS IN INPUT DATA FROM XN.DAT,*.ARRAYS, *.SCALARS
C
C
C LOCAL VARIABLES:
C
C XVALUE:    CHARACTER OF LENGTH 6 USED FOR DETERMINATION OF NUMBER
C            OF MODES
C MACHINF:   FREE-STREAM MACH NUMBER FROM *.SCALARS
C RHOINF:    FREE-STREAM DENSITY FROM *.SCALARS  (SLUGS/IN^3)
C AINF:      FREE-STREAM VELOCITY OF SOUND    (IN/SEC)
C R          COUNT HOW MANY NATURAL FREQUECIES FROM *.ARRAYS
C DATA(25)   FOR READING IN EACH LINE OF TIME HISTORY DATA
C I,K        USED FOR DO LOOPS
C P          NOT USED
C
C =====================================================================

    SUBROUTINE INPUT(TIME,Y,J,N,QINF,FREQ,INPUT_STOP,PROBNAME,FILEN)

    PARAMETER  (NMAX=10000)
    PARAMETER  (MODEMAX=20)

    CHARACTER  XVALUE*6,PROBNAME*20
    INTEGER    P,N,INPUT_STOP,K,G,FILEN
    DOUBLE PRECISION    DATA(25),TIME(NMAX),Y(NMAX,MODEMAX),FLAG(4),
   +        QINF,MACHINF,RHOINF,AINF,FREQ(MODEMAX),DUMMY(MODEMAX)

    K=1
    R=1

C   DETERMINING PROGRAM NAME LENGTH CALLING NAMLEN FUNCTION

    FILEN = NAMLEN(PROBNAME)

C   OPENING NECESSARY INPUT FILES FOR READING DATA

    OPEN(UNIT=1,FILE='xn.dat',STATUS='OLD',ERR=2001)
    OPEN(UNIT=3,FILE = PROBNAME(1:FILEN)//'.scalars',STATUS='OLD',
   +      ERR=2003)
    OPEN(UNIT=4,FILE = PROBNAME(1:FILEN)//'.arrays',STATUS='OLD',
   +      ERR=2004)

C DETERMINING NUMBER OF MODES FROM *.SCALARS

  1  READ(3,'(A)',ERR=2010)
     READ(3,'(A)',ERR=2010)
     READ(3,*,ERR=2010) J
```

```
C READING IN THE LAST POINT OF THE EXCITATION AND CALCULATING Q
C FROM *.SCALARS

     DO 10, P = 1,4
      READ(3,'(A)',ERR=2011)
  10  CONTINUE

     READ(3,*,ERR=2011) MACHINF,RHOINF,AINF

     QINF = 0.5*RHOINF*MACHINF*MACHINF*AINF*AINF

     DO 11, P = 1,3
      READ(3,'(A)',ERR=2012)
  11  CONTINUE

     READ(3,*,ERR=2012) (FLAG(K),K=1,4)
     INPUT_STOP = FLAG(4)


C READING IN THE NATURAL UNDAMPED FREQUENCY AT QINF=0 FROM *.ARRAYS


     DO 13, P = 1,4
     READ(4,'(A)',ERR=2013)
  13  CONTINUE

     DO 14, R = 1,J
      READ(4,*,ERR=2013) FREQ(R)
  14  CONTINUE


C READING IN ACTUAL QDOT TIME HISTORY DATA FROM XN.DAT FROM ASENL_CODE

  4  READ(1,'(A)',ERR=2014) XVALUE(:4)
    IF (XVALUE(:4) .EQ. ' DAT') GOTO 2
    GOTO 4

  2  DO 15,I=1,INPUT_STOP*3
     READ(1,*) DUMMY(I)
  15 CONTINUE

  5  READ(1,*,END=97,ERR=2014) (DATA(G),G=1,J*2+1)
      TIME(N) = DATA(1)
       DO 6, G=1,J
        Y(N,G) = DATA(J+G+1)
  6     CONTINUE
      N=N+1
    GOTO 5

C  DETERMINE IF LAST MODE IS A CONTROL MODE

    CMSUM = 0.D0

  97 DO 98, ZZ = 1,50
```

```
      CMSUM = DABS(Y(ZZ,J))+CMSUM
   98 CONTINUE

      IF (CMSUM .LT. .01) THEN
       J = J -1
       FLAG5 = 1
      ENDIF

C COUNTING BACK ON N

   99 N=N-1

     CLOSE(UNIT=1)
     CLOSE(UNIT=3)
     CLOSE(UNIT=4)

   8  RETURN

 2001 WRITE(*,*) 'ERROR WHEN OPENING XN.DAT FILE'
     STOP
 2003 WRITE(*,*) 'ERROR WHEN OPENING SCALARS FILE'
     STOP
 2004 WRITE(*,*) 'ERROR WHEN OPENING ARRAYS FILE'
     STOP
 2010 WRITE(*,*) 'ERROR WHEN READING SCALARS FILE - NROOTS'
     STOP
 2011 WRITE(*,*) 'ERROR WHEN READING SCALARS FILE - MACH, RHO, OR AINF'
     STOP
 2012 WRITE(*,*) 'ERROR WHEN READING SCALARS FILE - NEND'
     STOP
 2013 WRITE(*,*) 'ERROR WHEN READING ARRAYS FILE - FREQUENCIES'
     STOP
 2014 WRITE(*,*) 'ERROR WHEN READING ARRAYS FILE - GENERALIZED ',
    +         'VELOCITY DATA'
     STOP

     END

C END OF INPUT SUBROUTINE
```

```
C ===========================================================================
C
C SUBROUTINE MODAL_PARAMETERS:
C       THIS SUBROUTINE FIRST DETERMINES THE AR COEFFICIENTS
C       USING THE ARMA MODEL AND OVERDETERMINED SET OF EQS. FROM
C       A THEN DETEMINED RE-SAMPLED INPUT FOR N-5 AND N
C       NUMBER OF POINTS.  FROM THE AR COEFFICIENTS BAIRSTOWS
C       METHOD OF QUADRATIC FACTORIALS IS USED ON THE
C       CHARACTERISTIC EQUATION OF AR COEFFICIENTS.  (CALL
C       QUADFACT_OF_THETA).  FROM THESE FACTORIAL THE MODAL
C       VALUES OF EACH IF USEABLE ARE DETERMINED (CALL
C       MODAL_VALUES).  FROM THESE MODAL VALUES FOR BOTH N-5
C       AND N SETS OF POINTS, REGRESSIVE CONVERGENCE IS DETERMINED
C       BY COMPARING THE MODAL VALUES FROM EACH SET. THE RESULTS
C       FOR N POINTS ARE RETURNED.
C
C LOCAL VARIABLES:
C
C FLAG                IF PHIM IS SINGULAR, FLAG =1,INCREMENT K BY 1
C FACT                NUMBER OF BAIRSTOW FACTORIALS (EQUAL TO MODEL)
C MPCOUNT             USED TO COUNT FOR STORAGE OF TEMPORARY ARRAYS
C                     BETWEEN N-5 AND N POINTS
C NOLDAR              OLD NUMBER OF POINTS AFTER RE-SAMPLING
C RCOUNT              COUNT NUMBER OF DATA POINTS RETRIEVED
C Z                   FIRST DATA POINT OF Y DATA
C NEWY()              RE-SAMPLED INPUT DATA FOR THAT MODE SHAPE
C K                   FOR STARTING POINT (2M+1) OF MOSE METHOD
C PHI(,)              REGRESSION DATA MATRIX
C PSI()               DATA VECTOR
C PHIT(,)             TRANSPOSE FO PHI
C PHIM(,)             MULITPLICATION OF PHIT AND PHI AND THEN ITS
C                     INVERTED
C NEW(,)              MULTIPLICATION OF INVERTED PHIM AND PHIT
C TEMP                USED FOR MATRIX MULTIPLICATIONS
C QUADFACT(,3)        MATRIX OF QUADRATIC FACTORIALS RETURNED
C DT                  NEW SAMPLE PERIOD BASED FROM RE-SAMPLE FACTOR
C MPR_OLD(,)          MATRIX OF R VALUES (CONDITION OF OMEGA) FOR
C                     EACH MODE RETURNED FROM MODAL_VALUES
C                     SUBROUTINE
C MPSIGMA_OLD(,)      MATRIX OF DAMPING FACTORS FOR EACH MODE
C                     RETURNED FROM MODAL_VALUES SUBROUTINE
C MPOMEGA_OLD(,)      MATRIX OF DAMPING FREQUENCIES FOR EACH MODE
C                     RETURNED FROM MODAL_VALUES SUBROUTINE
C MPR_TEMP(,)         TEMPORARY MATRIX OF R VALUES (CONDITION OF
C                     OMEGA) FOR EACH MODE AT N-5 POINTS TO COMPARE
C                     WITH MPR_OLD RETURNED FROM MODAL_VALUES SUB-
C                     ROUTINE
C MPSIGMA_TEMP(,)     TEMPORARY MATRIX OF DAMPING FACTORS FOR EACH
C                     MODE AT N-1 POINTS TO COMPARE WITH MPSIGMA_OLD
C MPOMEGA_TEMP(,)     TEMPORARY MATRIX OF DAMPING FREQUENCIES FOR
C                     EACH MODE AT N-1 POINTS TO COMPARE WITH
C                     MPOMEGA_OLD
C OMEGA_TEMP_DIFF     % DIFFERENCE BT OMEGAS FROM N-1 AND N POINTS
C SIGMA_TEMP_DIFF     DIFFERENCE BT SIGMAS FROM N-2 AND N POINTS
C TOLS                TOLERANCE USED WITH SIGMA_TEMP_DIFF
C V,L,M,Q,W,I,J       USED FOR DO LOOPS
```

```
C
C ========================================================================

      SUBROUTINE MODAL_PARAMETERS(THETA,Y,MODEL,MODE,N,RE_SAMP_FACTOR,
     +                  DTORIG,MPR,MPSIGMA,MPOMEGA,MPCONV,
     +                  COUNT,FLAG4)

      PARAMETER  (NMAX=10000)
      PARAMETER  (MODEMAX=20)

      INTEGER     K,Q,M,W,RCOUNT,H,RE_SAMP_FACTOR,Z,FLAG,MODEL,
     +            FACT,COUNT,NOLDAR,MPCONV(MODEMAX,MODEMAX),
     +            MPCONV_OLD(MODEMAX),FLAG4
      DOUBLE PRECISION    PHI(NMAX,2*MODEMAX+1),PHIT(2*MODEMAX+1,NMAX),
     +            THETA(2*MODEMAX+1),TEMP,PSI(NMAX),
     +            PHIM(2*MODEMAX+1,2*MODEMAX+1),Y(NMAX,MODEMAX),
     +            NEW(2*MODEMAX+1,NMAX),NEWY(NMAX),
     +            QUADFACT(MODEMAX,3),DT,DTORIG,
     +            MPR(MODEMAX,MODEMAX),MPOMEGA(MODEMAX,MODEMAX),
     +            MPSIGMA(MODEMAX,MODEMAX),
     +            MPR_OLD(MODEMAX),MPOMEGA_OLD(MODEMAX),
     +            MPSIGMA_OLD(MODEMAX),MPR_TEMP(MODEMAX),
     +            MPOMEGA_TEMP(MODEMAX),MPSIGMA_TEMP(MODEMAX),
     +            OMEGA_TEMP_DIFF,SIGMA_TEMP_DIFF,TOLS


      FLAG = 0
      FLAG4 = 0
      FACT = 0
      MPCOUNT = 0

C DETERMINE NEW NUMBER OF POINTS

      N=(N/RE_SAMP_FACTOR)
      NOLDAR = N

C START LOOP TO DETERMINE MODAL PARAMETERS IF REGRESSIVE CONVERGENCE
C IS SEEN

      DO 40 V=2,0,-2
       RCOUNT=0
       N = NOLDAR-V
       Z = 1
 100   IF (FLAG .EQ. 1) THEN
        IF(N .LT. 20 ) THEN
         FLAG4 = 1
        RETURN
        ENDIF
       Z = Z+1
       ENDIF

C DETERMINE NEW Y VECTOR USING RE-SAMPLE FACTOR

      DO 20 L = Z,N
        NEWY(L)=Y((L-1)*RE_SAMP_FACTOR+1,MODE)
  20 CONTINUE
```

```
C DETERMINING PHI MATRIX AND PSI VECTOR WHICH
C DEVELOPS ARMA MODEL AND OVERDETERMINED SPECIFIED EQUATIONS.

      K=2*MODEL+1

      DO 1, M=K,N-1
       DO 2, Q=1,2*MODEL
        PHI(1,Q)=-NEWY(K-Q)
  2    CONTINUE
       PHI(1,K)=1
       PSI(1)=NEWY(K)
       DO 3, W=1,2*MODEL
        PHI(M-K+2,W)=-NEWY(M+1-W)
  3    CONTINUE
       PHI(M-K+2,K)=1
       PSI(M-K+2)=NEWY(M+1)
       RCOUNT=RCOUNT+1
  1   CONTINUE

C TRANSPOSE OF PHI

      DO 4 H=1,K
       DO 5, Q=1,RCOUNT
        PHIT(H,Q)=PHI(Q,H)
  5    CONTINUE
  4    CONTINUE


C MULTIPLICATION OF PHIT*PHI MAKING PHIM

      DO 6 I=1,K
       DO 7 J=1,K
        TEMP = 0.0
        DO 8 Q=1,RCOUNT
         PHIM(I,J) = PHIT(I,Q)*PHI(Q,J)+TEMP
         TEMP=PHIM(I,J)
  8     CONTINUE
  7    CONTINUE
  6   CONTINUE


C MATRIX INVERSE OF PHIM IN ITS PLACE

      CALL MATRIX_INVERSE(PHIM,K,FLAG)

      IF (FLAG .EQ. 1) GOTO 100

C MULTIPLICATION OF PHIM*PHIT

      DO 9 I=1,K
       DO 10 J=1,RCOUNT
        TEMP = 0.0
        DO 11 Q=1,K
        NEW(I,J) = PHIM(I,Q)*PHIT(Q,J)+TEMP
        TEMP=NEW(I,J)
  11    CONTINUE
```

```
 10  CONTINUE
  9 CONTINUE

C MULTIPLICATION OF NEW*PSI MAKING THETA BEING THE ARRAY OF AR
C COEFFICIENTS

    DO 12 I=1,K
     TEMP = 0.0
     DO 13 Q=1,RCOUNT
     THETA(I) = NEW(I,Q)*PSI(Q)+TEMP
      TEMP=THETA(I)
 13  CONTINUE
 12  CONTINUE

C CALL TO DETERMINE QUADRATIC FACTORIALS FOR CHARACTERISTIC EQUATION
C OF AR COEFFICIENTS

      CALL QUADFACT_OF_THETA(THETA,QUADFACT,MODEL,FACT)

C RESET DT

      DT = DTORIG*RE_SAMP_FACTOR

C CALL TO DETERMINE ROOTS OF FACTORIALS AND FREQUENCIES AND DAMPING
C FACTOR

      CALL MODAL_VALUES(MPR_OLD,MPOMEGA_OLD,MPSIGMA_OLD,
   +                    QUADFACT,FACT,DT)

C PUTTING RESULTS FOR N-2 POINTS IN TEMPORARY ARRAYS FOR COMPARING
C LATER

    IF (MPCOUNT .NE. 1) THEN

     DO 30 W=1,MODEL
      MPOMEGA_TEMP(W)=MPOMEGA_OLD(W)
      MPSIGMA_TEMP(W)=MPSIGMA_OLD(W)
      MPR_TEMP(W)=MPR_OLD(W)
 30  CONTINUE
    ENDIF

C SETTING MPCOUNT FOR IF STATEMENT ABOVE

     MPCOUNT = MPCOUNT + 1

C END OF REGRESSIVE CONVERGENCE OF N-2 AND N POINTS

 40 CONTINUE

C LOOKING FOR REGRESSIVE CONVERGENCE

     DO 50 W=1,MODEL
      MPCONV_OLD(W) = 0
      IF (MPOMEGA_OLD(W) .NE. 0.0 .AND. MPR_OLD(W) .NE. 0.D0) THEN
```

```
      OMEGA_TEMP_DIFF = 100*(MPOMEGA_TEMP(W)-MPOMEGA_OLD(W))/
     +              MPOMEGA_OLD(W)
      IF (ABS(MPSIGMA_OLD(W)) .GT. 1.D0) THEN
       SIGMA_TEMP_DIFF = 100*(MPSIGMA_TEMP(W)-MPSIGMA_OLD(W))/
     +              MPSIGMA_OLD(W)
       TOLS = 5.D0
      ELSE
       SIGMA_TEMP_DIFF = (MPSIGMA_TEMP(W)-MPSIGMA_OLD(W))
       TOLS = .5D0
      ENDIF

      IF(ABS(SIGMA_TEMP_DIFF) .LE. TOLS .AND. ABS(OMEGA_TEMP_DIFF)
     +       .LE. 10.D0 .AND. MPR_OLD(W) .NE. 0.D0) THEN
        MPCONV_OLD(W) = 1
      ELSE
        MPCONV_OLD(W) = 0
      ENDIF
     ENDIF
   50 CONTINUE

C COUNTING HOW MANY TIME BEEN THROUGH THIS SUBROUTINE

      COUNT = COUNT+1

C SETTING WORKING ARRAYS INTO FINAL ARRAYS TO RETURN

    DO 60 J=1,MODEL
     MPOMEGA(COUNT,J) = MPOMEGA_OLD(J)
     MPSIGMA(COUNT,J) = MPSIGMA_OLD(J)
     MPR(COUNT,J) = MPR_OLD(J)
     MPCONV(COUNT,J) = MPCONV_OLD(J)
  60 CONTINUE

    RETURN
    END

C END OF MODAL_PARAMETERS SUBROUTINE
```

```
C ===========================================================================
C
C SUBROUTINE MATRIX_INVERSE: INVERT PHIM
C
C THIS ROUTINE WAS TAKEN FROM NUMERICAL RECIPES IN FORTRAN 77: THE ART
C OF SCIENTIFIC COMPUTING, CHAPTER 2.1. THIS ROUTINE INVERTS A GENERAL
C MATRIX OF (NXN) USING GAUSS-JORDAN ELIMINATION WITH FULL PIVOTING IN
C PLACE. THE ORIGINAL MATRIX A IS DETROYED AND REPLACED BY THE INVERSE
C OF THE ORIGINAL MATRIX A.
C
C LOCAL VARIABLES:
C
C IPIV(MODEMAX)          INTEGER ARRAY USED FOR BOOKKEEPING OF PIVOTING
C BIG                    USED FOR PIVOTING COLUMNS
C A(MODEMAX,MODEMAX)     DUMMY VARIABLE FOR PHIM (MATRIX TO INVERT)
C IROW                   ITH ROW VALUE FOR SORTING
C ICOL                   ITH COLUMN VALUE FOR SORTING
C DUM                    USED FOR INTERCHANGING ROWS,COLUMNS, AND ETC.
C N                      NUMBER ROW OR COL
C INDXR(MODEMAX)         INTEGER ARRAY USED FOR BOOKKEEPING OF ROW
C                        PIVOTING
C INDXC(MODEMAX)         INTEGER ARRAY USED FOR BOOKKEEPING OF COLUMN
C                        PIVOTING
C PIVINV                 USED FOR DIVIDING PIVOT ROW BY PIVOT ELEMENT
C J,I,L,Q,LL,K           USED FOR DO LOOPS
C
C ===========================================================================

      SUBROUTINE MATRIX_INVERSE(A,N,FLAG)

      PARAMETER  (MODEMAX=20)

      INTEGER           N,I,ICOL,IROW,J,Q,L,LL,INDXC(2*MODEMAX+1),
     +             INDXR(2*MODEMAX+1),IPIV(2*MODEMAX+1),FLAG
      DOUBLE PRECISION    A(2*MODEMAX+1,2*MODEMAX+1),BIG,DUM,PIVINV

      DO 11  J=1,N
        IPIV(J)=0
   11 CONTINUE

C MAIN LOOP OVER THE COLUMNS TO BE REDUCED

      DO 22  I=1,N
        BIG=0.

C SEARCH FOR PIVOT ELEMENT OF EACH COLUMN

        DO 13 J=1,N
         IF(IPIV(J).NE.1) THEN
          DO 12 K=1,N
           IF(IPIV(K).EQ.0) THEN
            IF(DABS(A(J,K)).GE.BIG) THEN
             BIG=DABS(A(J,K))
             IROW=J
             ICOL=K
            ENDIF
```

```
        ELSE IF(IPIV(K).GT.1) THEN
         PAUSE 'PHIM IS A SINGULAR MATRIX. CANNOT INVERT'
         FLAG = 1
         RETURN
        ENDIF
 12     CONTINUE
      ENDIF
 13   CONTINUE

C HAVING THE PIVOT ELEMENT, ROWS ARE NOW INTERCHANGED TO PUT THE PIVOT
C ON THE DIAGONAL.  THE COLUMNS ARE NOT PHYSICALLY INTERCHANGED, JUST
C RELABED.

      IPIV(ICOL)=IPIV(ICOL)+1
      IF(IROW.NE.ICOL) THEN
       DO 14 L=1,N
         DUM=A(IROW,L)
         A(IROW,L)=A(ICOL,L)
         A(ICOL,L)=DUM
 14    CONTINUE
      ENDIF

C NOW READY TO DIVIDE THE PIVOT ROW BY THE PIVOT ELEMENT, LOCATED AT
C IROW AND ICOL

      INDXR(I)=IROW
      INDXC(I)=ICOL
      IF (A(ICOL,ICOL).EQ.0.) THEN
        PAUSE 'PHIM IS A SINGULAR MATRIX. CANNOT INVERT'
        FLAG = 1
        RETURN
      ENDIF
      PIVINV=1./A(ICOL,ICOL)
      A(ICOL,ICOL)=1.D0
      DO 16 L=1,N
        A(ICOL,L)=A(ICOL,L)*PIVINV
 16   CONTINUE

C NOW REDUCE THE ROWS EXCEPT FOR THE PIVOT ONE.
      DO 21 LL=1,N
       IF(LL.NE.ICOL) THEN
        DUM=A(LL,ICOL)
        A(LL,ICOL)=0.D0
        DO 18 L=1,N
         A(LL,L)=A(LL,L)-A(ICOL,L)*DUM
 18      CONTINUE
       ENDIF
 21   CONTINUE
 22 CONTINUE

C END OF MAIN LOOP OVER COLUMNS OF REDUCTION.  HAVE INVERSE EXCEPT
C COLUMNS ARE OUT OF PLACE SO MUCH PUT THEM IN THE RIGHT ORDER BY
C THE NEXT LOOP.

    DO 24 L=N,1,-1
      IF(INDXR(L).NE.INDXC(L)) THEN
```

```
      DO 23 Q=1,N
       DUM=A(Q,INDXR(L))
       A(Q,INDXR(L))=A(Q,INDXC(L))
       A(Q,INDXC(L))=DUM
 23     CONTINUE
      ENDIF
 24 CONTINUE

C HAVE INVERSE IN MATRIX A SO NOW RETURN

   RETURN
   END
C END OF MATRIX_INVERSE SUBROUTINE
```

```
C ====================================================================
C
C SUBROUTINE QUADFACT_OF_THETA:
C       DETERMINE ROOTS OF CHARACTERISTIC EQUATION OF AR
C       COEFFICIENTS USING BAIRSTOWS METHOD FOR QUADRATIC
C       FACTORING.
C LOCAL VARIABLES:
C
C M            ORDER OF CHARACTERISTIC POLYNOMIAL OF AR COEFFICIENTS
C TOL          TOLERANCE VALUE FOR CONVERGENCE ON R AND S
C R,S          PARAMETERS IN TRIAL FACTOR X^2 - RX - S
C ITER         NUMBER OF ITERATIONS FOR BAIRSTOWS METHOD, NO SET LIMIT
C K            NUMBER OF QUADRATIC FACTORS
C COEFF()      DUMMY VECTOR OF AR COEFFICIENTS
C B()          ARRAY HOLDING THE COEFFICIENTS WHEN THE INTIAL
C              CHARACTERISTIC POLYNOMIAL IS DIVIDED BY TRIAL QUADRATIC
C              FACTOR
C C()          ARRAY HOLDING THE PARTIAL DERIVATIVES
C DENOM        DENOMINATOR FOR DELR AND DELS
C DELR,DELS    RATIO OF DETERMINANTS FOR ADJUSTMENTS TO IMPROVE R AND S
C
C ====================================================================

      SUBROUTINE QUADFACT_OF_THETA(THETA,QUADFACT,MODEL,FACT)

      PARAMETER  (MODEMAX=20)

      INTEGER          I,ITER,K,J,FACT
      DOUBLE PRECISION    COEFF(2*MODEMAX),B(2*MODEMAX),C(2*MODEMAX),
     +             THETA(2*MODEMAX+1),R,S,TOL,
     +             QUADFACT(MODEMAX,3),DENOM,DELR,DELS

C INITIAL NECESSARY PARAMETERS

      M = 2*MODEL
      TOL = 1.D-8
      R = 1.D0
      S = 1.D0
      ITER = 1
      K=0

C FILL IN STARTING COEFFICIENTS WITH AR COEFFICIENTS

      COEFF(1) = 1.D0
      DO 1, I=1,M
        COEFF(I+1) = THETA(I)
        B(I) = 0.D0
        C(I) = 0.D0
    1 CONTINUE

C CALCULATE B AND C VECTORS

    3 B(1) = 1.D0
      C(1) = 1.D0

    4 B(2) = COEFF(2) + R*B(1)
```

```
      C(2) = B(2) + R*C(1)

      DO 2, J=3,M+1
        B(J) = COEFF(J) + R*B(J-1) + S*B(J-2)
        C(J) = B(J) + R*C(J-1) + S*C(J-2)

    2 CONTINUE

      DENOM = C(M-1)*C(M-1) - C(M)*C(M-2)

C CHECK TO SEE R AND S GUESS PROVIDE BAD RESULTS

      IF (DENOM .EQ. 0.D0) THEN
         R = R + 1.D0
         S = R + 1.D0
         ITER=1
         GOTO 3
      ENDIF

C COMPUTE NEW R AND S

      DELR = (-B(M)*C(M-1) + B(M+1)*C(M-2))/DENOM
      R = R + DELR
      DELS = (-C(M-1)*B(M+1) + C(M)*B(M))/DENOM
      S = S + DELS


C CHECK IF CONVERGENCE UPON R AND S IS FOUND.  IF IT HAS SET
C FIRST QUADRATIC FACTOR.

      IF ((DABS(DELR) + DABS(DELS)) .GT. TOL) THEN
        ITER = ITER+1
        GOTO 3
      ELSE
        K=K+1
        QUADFACT(K,1) = 1.D0
        QUADFACT(K,2) = -R
        QUADFACT(K,3) = -S
      ENDIF

C IF CONVERGENCE FOUND FIND NEW REDUCED POLYNOMIAL FROM ONE OF THE
C CASES
C CASE 1: IF LINEAR EQUATION RESULT SET THEN STOP (NEVER OCCURS)
C CASE 2: IF QUADRATIC EQUATION RESULTS SET THEN STOP
C CASE 3: IF HIGHER ORDER POLYNOMIAL SET AND DIVIDE OUT NEW FACTORIALS
C         BY RETURNING TO 4

      M = M - 2

      SELECT CASE (M)
      CASE (1)
       K=K+1
       DO 5 J=1,2
        QUADFACT(K,J) = B(J)
    5  CONTINUE
      CASE (2)
```

```
      K = K+1
      DO 6, J = 1,3
       QUADFACT(K,J) = B(J)
   6   CONTINUE
     CASE (3:MODEMAX)
       DO 7, J=1,M+1
       COEFF(J) = B(J)
       ITER = 0
       R = 1.D0
       S = 1.D0
   7    CONTINUE
       GOTO 4
     END SELECT


     FACT=K

     RETURN
     END

C END OF QUADFACT_OF_THETA SUBROUTINE
```

```
C ==================================================================
C
C SUBROUTINE MODAL_VALUES:
C       DETERMINE ROOTS FROM QUADRATIC FACTORIALS
C       USING BASIC QUADRATIC EQUATIONS THEN TAKING ROOTS
C       AND DETERMINE MODAL PARAMETERS (SIGMA AND OMEGA)
C
C LOCAL VARIABLES:
C
C B            B VALUE OF QUADRATIC EQUATION
C A            A VALUE OF QUADRATIC EQUATION
C C            C VALUE OF QUADRATIC EQUATION
C REALROOT     REAL VALUE OF COMPLEX ROOT
C IMAGROOT     IMAGINARY VALUE OF COMPLEX ROOT
C SIGMA        DAMPING PRODUCT (RAD/S)
C OMEGA        DAMPING FREQUENCY (RAD/S)
C J            USED FOR DO LOOPS
C ==================================================================

      SUBROUTINE MODAL_VALUES(MPR_OLD,MPOMEGA_OLD,MPSIGMA_OLD,
     +                  QUADFACT,FACT,DT)

      PARAMETER  (MODEMAX=20)

      INTEGER         J,FACT,R
      DOUBLE PRECISION   MPR_OLD(MODEMAX),QUADFACT(MODEMAX,3),
     +           SIGMA,REALROOT,IMAGROOT,A,B,C,DT,OMEGA,
     +           MPOMEGA_OLD(MODEMAX),
     +           MPSIGMA_OLD(MODEMAX)


C DO LOOP TO DETERMINE MODAL PARAMETERS FROM EACH QUADRATIC
FACTORIALS

      DO 1, J=1,FACT

C SET QUADRATIC EQUATION COEFFICIENTS

       A = QUADFACT(J,1)
       B = QUADFACT(J,2)
       C = QUADFACT(J,3)

C CHECK IF COMPLEX OR NOT AND THEN DETERMINE REAL AND/OR IMAG VALUES
C OF ROOT

      IF ((B*B - 4*A*C) .LT. 0.0) THEN
       REALROOT = -B/2*A
       IMAGROOT =  SQRT(DABS(B*B-4*A*C))/2*A
      ELSE
       REALROOT = (-B+(SQRT(B*B-4*A*C)))/2*A
       IMAGROOT = 0.D0
      ENDIF

C CALCULATE SIGMA AND OMEGA FROM ALL ROOTS

      SIGMA = -DLOG(REALROOT*REALROOT+IMAGROOT*IMAGROOT)/(2.D0*DT)
```

```
      OMEGA = DATAN(IMAGROOT/REALROOT)/DT

C CONDITION CHECK OF ROOTS TO SEE IF ROOTS ARE GOOD ESTIMATES

      IF (IMAGROOT .EQ. 0.D0) THEN
       R = 0
       SIGMA = 0.D0
       OMEGA = 0.D0
      ENDIF

      IF (REALROOT .GT. 0.D0) THEN
       R = 1
      ELSE
       R = 0
       SIGMA = 0.D0
       OMEGA = 0.D0
      ENDIF

C CONDITION TO CHECK IF OVERDAMPED MODAL PARAMETERS

      IF(SIGMA .GE. OMEGA) THEN
       R=0
       SIGMA = 0.D0
       OMEGA = 0.D0
      ENDIF

        MPOMEGA_OLD(J)=OMEGA
        MPSIGMA_OLD(J)=SIGMA
        MPR_OLD(J)=R

    1 CONTINUE

      RETURN
      END

C END OF MODAL_VALUES SUBROUTINE
```

```
C ===========================================================================
C
C SUBROUTINE MP_COMMON:
C       DETERMINES COMMON MODAL PARAMETERS FROM EACH MODEL FOR
C       EACH MODE.
C
C LOCAL VARIABLES:
C
C PI              PI= 4.D0 *DATAN(1.D0)
C PRINT_SIGMA     USED FOR PRINTING DAMPING FACTOR VALUES FOR EACH MODEL
C                 AND MODE IN TERMS OF DIMENSIONLESS ZETA
C PRINT_OMEGA     USED FOR PRINTING OMEGA VALUES FOR EACH MODEL AND MODE
C                 IN TERMS OF HERTZ
C TEMP            COUNTER USED TO DETERMINE IF NO MODAL PARMATERS ARE
C                 MATCHED BETWEEN MODEL FOR THAT MODE
C MATCHMODE       COUNTER USED TO DETERMINE HOW MANY MATCHED MODAL
C                 PARAMETER SETS OCCURED BETWEEN EACH MODEL FOR THAT C
                  MODE
C CONVCOUNT       COUNTER USED TO DETERMINE IF REGRESSIVE CONVERGENCE
C                 OCCURRED BETWEEN ALL MODELS FOR THAT MODE
C SIGMATEMP       TEMPORARY STORAGE OF SIGMAS FOR LATER COMPARISON
C OMEGATEMP       TEMPORARY STORAGE OF OMEGAS FOR LATER COMPARISON
C SIGMADIFF       USED FOR DETERMINING COMMON SIGMAS BETWEEN EACH MODEL
C OMEGADIFF       USED FOR DETERMINING COMMON OMEGAS BETWEEN EACH MODEL
C TOLS            TOLERANCE WHEN COMPARING SIGMAS
C NMODE           PLACEMENT OF NTH MATCHED MODAL PARAMETER SET
C G,J,K,U         USED FOR DO LOOPS
C ===========================================================================

      SUBROUTINE MP_COMMON(MP,MPR,MPSIGMA,MPOMEGA,MPCONV,COUNT,MODEL,
     +            MODE,NMODE,FLAG1,FLAG2,SCREEN,LOWMODE,
     +            HIGHMODE)

      PARAMETER  (MODEMAX=20)
      PARAMETER  (NMAX=10000)

      INTEGER          MODE,COUNT,MODEL,K,NMODE,MATCHMODE,TEMP,G,U,
     +           FLAG1,CONVCOUNT,MPCONV(MODEMAX,MODEMAX),FIAG2,
     +           SCREEN,LOWMODE,HIGHMODE
      DOUBLE PRECISION    MPR(MODEMAX,MODEMAX),MPOMEGA(MODEMAX,MODEMAX),
     +           MPSIGMA(MODEMAX,MODEMAX),SIGMADIFF,OMEGADIFF,
     +           MP(MODEMAX*MODEMAX,4),TOLS,DUMMY,
     +           PRINT_OMEGA,PI,OMEGATEMP,SIGMATEMP,CONVTEMP

      PI = 4.D0 *DATAN(1.D0)

C PRINT OUT ALL MODAL PARAMETERS FOR EACH MODEL AND MODE
C THIS IS THE DATA USED FOR BACKUP JUST IN CASE SOMETHING GOES
C WRONG WITH FINDING COMMON PARAMETERS AND SORTING

      DO 1, G = 1,HIGHMODE-LOWMODE+1

      IF (SCREEN .EQ. 2) THEN
       WRITE (2,100) G+LOWMODE-1
      ELSE
       WRITE (*,100) G+LOWMODE-1
```

```
      ENDIF

      DO 2, J = 1,MODEL-1
       IF(MPR(G,J) .NE. 0.D0 .AND. MPOMEGA(G,J) .NE. 0.D0) THEN
        DUMMY=MPSIGMA(G,J)/(SQRT(MPSIGMA(G,J)**2 +
     +          MPOMEGA(G,J)**2))
        PRINT_OMEGA = MPOMEGA(G,J)/(2*PI)

        IF(SCREEN .EQ. 2) THEN
         WRITE(2,200) PRINT_OMEGA,DUMMY,MPCONV(G,J)
        ELSE
         WRITE(*,200) PRINT_OMEGA,DUMMY,MPCONV(G,J)
        ENDIF

       ENDIF
    2 CONTINUE
    1 CONTINUE

      TEMP = 0

C START OF COMPARISON LOOP

      DO 3, J = 1,MODEL-1
        MATCHMODE = 0
        CONVCOUNT = 0

C IF USEABLE MODE AND MODAL PARAMETERS ARE NOT ZERO THEN CONTINUE
C START COMPARING FROM LAST MODEL

        IF(MPR(COUNT,J) .NE. 0.D0 .AND. MPOMEGA(COUNT,J) .NE. 0.D0) THEN
          OMEGATEMP = MPOMEGA(COUNT,J)
          SIGMATEMP = MPSIGMA(COUNT,J)
          CONVTEMP  = MPCONV(COUNT,J)
C             IF(MPCONV(COUNT,J) .EQ. 0 ) THEN
C              CONVCOUNT=CONVCOUNT+1
C             WRITE(2,*) 'HERE1',MATCHMODE,CONVCOUNT
C             ENDIF


C COMPARE EACH MODAL PARAMETER TO PREVIOUS TO MODELS

          DO 4, K=COUNT-1,1,-1

C START COMPARING EACH MODAL PARAMETER BT. MODELS

          DO 5, U=1,MODEL-1
           OMEGADIFF = 100.D0*(OMEGATEMP-MPOMEGA(K,U))/OMEGATEMP
           IF (ABS(SIGMATEMP) .LT. 2.D0) THEN
            SIGMADIFF = SIGMATEMP-MPSIGMA(K,U)
            TOLS = .5D0
           ELSE
            SIGMADIFF = 100.D0*(SIGMATEMP-MPSIGMA(K,U))/SIGMATEMP
            TOLS = 20.D0
           ENDIF

C IF MATCHED A MODE BETWEEN THE MODELS
```

```
          IF(DABS(OMEGADIFF) .LT. 10.D0 .AND.
     +        DABS(SIGMADIFF) .LT. TOLS) THEN

C CHECK TO SEE IF MATCHED MODE REGRESSIVELY CONVERGED

          IF(MPCONV(K,U) .EQ. 0 .AND. K .GT. 2) THEN
           CONVCOUNT = CONVCOUNT + 1
          ENDIF
C COUNT IF A MODE IS MATCHED

          MATCHMODE = MATCHMODE + 1
          IF (MATCHMODE .EQ. 1) THEN
           NMODE = NMODE+1
          ENDIF
          MP(NMODE,1) = OMEGATEMP
          MP(NMODE,2) = SIGMATEMP
          MP(NMODE,3) = MODE
          MP(NMODE,4) = MATCHMODE
          TEMP = TEMP+1
         ENDIF
 5     CONTINUE
 4     CONTINUE
      ENDIF
 3 CONTINUE

C IF NO MATCHES OF MODES BETWEEN EACH MODEL DUE TO REGRESSIVE
CONVERNCE
C PROGRAM STOPS AND ASKS USER TO SUPPLY MORE POINTS

      FLAG2 = 0
      IF (CONVCOUNT .GE. 2) THEN
       FLAG2 = 1
       RETURN
      ENDIF

C IF NO MATCHES OF MODES BETWEEN EACH MODEL FOR THAT MODE SHAPE
C PROGRAM STOPS AND ASKS USER TO SUPPLY MORE POINTS

      FLAG1 = 0
      IF (TEMP .EQ. 0) THEN
       FLAG1 = 1
       RETURN
      ENDIF


 100   FORMAT(' MODEL ',I2)
 200   FORMAT(F10.4,F10.6,I2)
      RETURN
      END

C END OF MP_COMMON SUBROUTINE
```

```
C ==============================================================
C
C SUBROUTINE MP_SORT:
C     FROM COMMON MODAL PARAMETERS FOR EACH MODE AND ALL MODES THIS
C     SUBROUTINE SORTS THE MODE WITHIN EACH MODE BY THE FREQUENCY
C     THEN DETERMINES FROM ALL MODES THE EXACT MODAL PARAMETERS FOR
C     EACH MODE
C
C
C LOCAL VARIABLES:
C
C P              COUNTER OF HOW MANY COMMON MODAL PARAMETERS
C TEMPO()        TEMPORARY ARRAY OF TEMPERARY COMMON OMEGAS FOR
C                THAT MODE FOR SORTING AND FINDING FINAL MODES
C TEMPS()        SIMILAR TO TEMPO, BUT FOR SIGMAS
C TEMPM()        SIMILAR TO TEMPS, BUT FOR NUMBER OF MATCHED
C                MODES WITHIN EACH MODE SHAPE
C JMIN           PLACEMENT OF MINIMUM OMEGA
C TEMP1,2,4      NEEDED FOR SORTING COMMON MODES WITHIN EACH
C                MODE SHAPE COUNTING NUMBER OF COMMON MODES
C                WITHIN ECH MODE SHAPE
C ANS1           = 1 IF OMEGA IS GREATER THAN LAST MODE OMEGA
C ANS2           = 1 IF MODE IS COMPARABLE TO A PREVIOUS MODE
C DIFFO,DIFFS    DIFFERENCES NEEDED FOR COMPARING PRESENT MODES
C                WITH PAST MODES
C TOLS           TOLERANCE FOR COMPARING SIGMAS
C U,K,J,I,Y,E    FOR DO LOOPS ONLY
C
C ==============================================================

      SUBROUTINE MP_SORT(MP,NMODE,MODE,FLAG3,FINAL_MODES,OTHER_MODES
     +              ,SCREEN)

      PARAMETER  (MODEMAX=20)
      PARAMETER  (NMAX=10000)

      INTEGER          NMODE,MODE,P,U,K,J,I,T,FLAG3,
     +              OTHER_MODES(MODEMAX,MODEMAX),E,Y,SCREEN
      DOUBLE PRECISION    MP(MODEMAX*MODEMAX,4),TEMPO(MODEMAX),
     +              TEMPS(MODEMAX),TEMPM(MODEMAX),
     +              TEMP1,TEMP2,TEMP4,DIFFO,DIFFS,TOLS,
     +              FINAL_MODES(MODEMAX,MODEMAX),DUMMY
      PI = 4.D0 *DATAN(1.D0)

C SORTING ALL MATCHED (CALCULATED AND SYSTEM) MODES WITHIN EACH MODE BY
C FREQUENCY

      DO 4, U=1,NMODE
       P = 0
       DO 1, K=1,NMODE
         IF (MP(K,3) .EQ. U) THEN
          P = P + 1
          TEMPO(P) = MP(K,1)
          TEMPS(P) = MP(K,2)
          TEMPM(P) = MP(K,4)
         ENDIF
```

```
 1 CONTINUE
     DO 2, I=1,P-1
      JMIN = I
      DO 3, J = I+1,P
       IF (TEMPO(J) .LT. TEMPO(JMIN)) JMIN = J
 3     CONTINUE
       TEMP1 = TEMPO(I)
       TEMP2 = TEMPS(I)
       TEMP4 = TEMPM(I)
       TEMPO(I) = TEMPO(JMIN)
       TEMPS(I) = TEMPS(JMIN)
       TEMPM(I) = TEMPM(JMIN)
       TEMPO(JMIN) = TEMP1
       TEMPS(JMIN) = TEMP2
       TEMPM(JMIN) = TEMP4
 2    CONTINUE

     DO 5, Y=1,P
      R = R+1
      MP(R,1)= TEMPO(Y)
      MP(R,2)= TEMPS(Y)
      MP(R,4)= TEMPM(Y)
 5    CONTINUE

 4 CONTINUE

C PRINT OUT SORTED MATCH MODES

    IF (SCREEN .EQ. 2) THEN
     WRITE(2,*)
     WRITE(2,*)
     WRITE(2,*) 'ALL MATCHED MODAL PARAMETERS FROM ABOVE RESULTS'
     WRITE(2,*)
     WRITE(2,*) '      DAMPING    DAMPING'
     WRITE(2,*) '    FREQUENCY   FACTOR      # OF'
     WRITE(2,*) 'MODE    (HZ)              MATCHES'
     DO 40, KK=1,NMODE
       DUMMY = MP(KK,2)/(SQRT(MP(KK,2)**2 +MP(KK,1)**2))
        WRITE(2,500) MP(KK,3),MP(KK,1)/(2.D0*PI),DUMMY,MP(KK,4)
 40  CONTINUE
    ELSE
     WRITE(*,*)
     WRITE(*,*)
     WRITE(*,*) 'ALL MATCHED MODAL PARAMETERS FROM ABOVE RESULTS'
     WRITE(*,*)
     WRITE(*,*) '      DAMPING    DAMPING'
     WRITE(*,*) '    FREQUENCY   FACTOR      # OF'
     WRITE(*,*) 'MODE    (HZ)              MATCHES'
     DO 41, KK=1,NMODE
       DUMMY = MP(KK,2)/(SQRT(MP(KK,2)**2 +MP(KK,1)**2))
       WRITE(*,500) MP(KK,3),MP(KK,1)/(2.D0*PI),DUMMY,MP(KK,4)
 41  CONTINUE
    ENDIF

 500     FORMAT(' ',F3.1,'   ',F7.4,F10.6,'       ',F3.1)
```

```
C DETERMINE FINAL MODE RESULTS FOR SORTED RESULTS

      DO 6, U=1,MODE
       ANS1 = 0
       ANS2 = 0
       P = 0

C FILL UP WORKING ARRAY OF OMEGA AND SIGMA

       DO 7, K=1,NMODE
         IF (MP(K,3) .EQ. U) THEN
          P = P+1
          TEMPO(P) = MP(K,1)
          TEMPS(P) = MP(K,2)
          TEMPM(P) = MP(K,4)
         ENDIF
    7  CONTINUE

C CHECK TO SEE IF FIRST MODE

       IF (U .EQ. 1) THEN
        FINAL_MODES(U,1) = TEMPO(1)
        FINAL_MODES(U,2) = TEMPS(1)
        GOTO 6
       ENDIF

C CHECK TO SEE IF ONLY ONE COMMON MODE RESULT FOR THIS
C MODE SHAPE

      IF (P .EQ. 1) THEN
       FINAL_MODES(U,1) = TEMPO(P)
       FINAL_MODES(U,2) = TEMPS(P)
       GOTO 6


C IF MORE THAN ONE COMMON MODE COMPARE TO PREVIOUS EXACT MODES
GATHERED



      ELSE
       NOM = 0
       ANS1 = 1
       DO 9, E=1,P
        ANS2 = 0
        FREQDUM = TEMPO(E)
         IF (DABS(FREQDUM - FREQ(U)*2*PI) .LE. 10.D0) THEN
         DO 11, Y = 1,U-1


          DIFFO = 100*(FINAL_MODES(Y,1)-TEMPO(E))/TEMPO(E)
          IF (DABS(TEMPS(E)) .LT. 1.D0 .AND. FINAL_MODES(Y,2) .LT. 1.D0
     +            ) THEN
           DIFFS = FINAL_MODES(Y,2)-TEMPS(E)
           TOLS  = .25D0
```

```
        ELSE
         DIFFS = 100.D0*(FINAL_MODES(Y,2)-TEMPS(E))/TEMPS(E)
         TOLS  = 7.0D0
        ENDIF


        IF (DABS(DIFFO) .LT. 5.D0 .AND. DABS(DIFFS) .LT. TOLS) THEN
         ANS2 =  1
        ENDIF

   11    CONTINUE

C IF NO MODES MATCH PREVIOUS MODES USE IT AND FIND DOMINANT MATCH

      IF(ANS2 .EQ. 1) THEN
        NOM = 0
        GOTO 9
        ANS1=0
      ENDIF

       IF(TEMPM(E) .GE. NOM) THEN
        FINAL_MODES(U,1) = TEMPO(E)
        FINAL_MODES(U,2) = TEMPS(E)
        NOM = TEMPM(E)
        ANS1=0
       ENDIF

      ENDIF
    9 CONTINUE

     ENDIF

       IF (ANS1 .EQ. 1) THEN
        FINAL_MODES(U,1) = TEMPO(E-1)
        FINAL_MODES(U,2) = TEMPS(E-1)
       ENDIF


    6 CONTINUE


C CHECK TO DETERMINE IF ENOUGH POINTS WERE USED AND ALL MODES WERE
C IDENTIFIED

     DO 15, T=1,MODE
      IF (FINAL_MODES(T,1) .EQ. 0.D0 .OR.
    +       FINAL_MODES(T,2) .EQ. 0.D0) THEN
       FLAG3=1
      ENDIF
   15 CONTINUE

C DETERMINE IF OTHER MODES THAT MAY EXIST BUT NOT LIMITED TO

     DO 12, U=1,MODE
     Y = 0
      DO 13, K=1,NMODE
```

```
      IF (MP(K,3) .EQ. U) THEN
       DO 14 T=1,MODE
       IF (FINAL_MODES(T,1) .NE. 0.D0) THEN
        DIFFO = 100.D0*(MP(K,1)-FINAL_MODES(T,1))/FINAL_MODES(T,1)

        IF (DABS(FINAL_MODES(T,2)) .LT. 1.D0 .AND. MP(K,2) .LT.
     +        1.D0)THEN
         DIFFS = FINAL_MODES(T,2)-MP(K,2)
         TOLS  = .3D0
        ELSE
         DIFFS = 100.D0*(MP(K,2)-FINAL_MODES(T,2))/FINAL_MODES(T,2)
         TOLS  = 5.D0
        ENDIF
       ENDIF

       IF (DABS(DIFFO) .LT. 2.D0 .AND. U .NE. T .AND. DABS(DIFFS)
     +        .LT. TOLS) THEN
        Y = Y+1
        OTHER_MODES(U,Y) = T
       ENDIF
 14    CONTINUE
      ENDIF
 13  CONTINUE
 12 CONTINUE

     RETURN
     END

C END OF MP_SORT SUBROUTINE
```

```
C ==================================================================
C
C FUNCTION NAMLEN:
C       THIS FUNCTION DETERMINES THE CHARACTER LENGTH OF THE PROBLEM
C       NAME ENTERED BY THE USER
C
C
C ==================================================================

      INTEGER FUNCTION NAMLEN( FILEN )

      CHARACTER*20 FILEN

      NAMLEN = 0
      DO I = 1,30
         IF ( FILEN(I:I) .NE. ' ' ) THEN
            NAMLEN = NAMLEN+1
         ELSE
            GOTO 101
         ENDIF
      ENDDO
  101 RETURN
      END
C END OF FUNCTION NAMLEN AND SOURCE CODE
```

APPENDIX E: OUTPUT FILE *.TXT FROM MOSE.F AND TO SCREEN

OUTPUT FILE:

ARMA MODEL WITH MODEL OVERSPECIFICATION RESULTS SHOWING MODAL PARAMETERS CONVERGED UPON IN EACH MODE.

```
MODE= 1  NAT. FREQ=  3.18204  RE-SAMPLE FACTOR=  8  ORIG. SAMPLE FREQ. = 150.2
MODEL  3
MODEL  4
  3.3786   .479601 1
MODEL  5
  3.4779   .338666 1
MODEL  6
  3.2668   .349334 0

MODE= 2  NAT. FREQ=  4.02851  RE-SAMPLE FACTOR=  7  ORIG. SAMPLE FREQ. = 150.2
MODEL  3
  5.0525  -.016670 1        Damped Frequency for this mode
  5.2696  -.033264 1        and model order.
MODEL  4
  5.0447  -.018206 1        Damping factors for this
  5.2683  -.033348 1        mode and model order.
MODEL  5
  5.0478  -.016894 1        If = 1, then these parameters
  5.2691  -.033249 1        have regressively converged.
MODEL  6
  5.0474  -.017189 1
  5.2688  -.033234 1

MODE= 3  NAT. FREQ=  4.05237  RE-SAMPLE FACTOR=  7  ORIG. SAMPLE FREQ. = 150.2
MODEL  3
  5.0452  -.016799 1
  5.2694  -.033213 1
MODEL  4
  5.0470  -.015857 1
  5.2685  -.033329 1
MODEL  5
  5.0489  -.017099 0
  3.3195   .391550 1
  5.2688  -.033208 1
MODEL  6
  5.0478  -.016974 1
  3.2547   .326707 1
  5.2688  -.033233 1

MODE= 4  NAT. FREQ=  5.55739  RE-SAMPLE FACTOR=  5  ORIG. SAMPLE FREQ. = 150.2
MODEL  3
  5.3849   .014703 1
```

5.2637 -.031192 1
MODEL 4
5.2710 -.031966 1
5.9743 .044350 0
5.1296 .005510 1
MODEL 5
5.2714 -.032784 1
5.7542 .038391 1
5.0796 -.008818 1
MODEL 6
6.3306 .175831 1
5.5776 .049054 1
5.0472 -.016760 1
5.2691 -.033224 1

MODE= 5 NAT. FREQ= 6.91297 RE-SAMPLE FACTOR= 4 ORIG. SAMPLE FREQ. = 150.2
MODEL 3
5.2402 .038059 0
5.2631 -.036314 1
MODEL 4
5.2683 -.033067 1
6.7402 .208362 1
5.0578 -.017902 1
MODEL 5
6.5493 .207436 1
5.0513 -.017882 1
5.2689 -.033124 1
MODEL 6
6.4773 .228936 1
9.3721 .381071 1
5.0511 -.017611 1
5.2697 -.033102 1

MODE= 6 NAT. FREQ= 6.97122 RE-SAMPLE FACTOR= 4 ORIG. SAMPLE FREQ. = 150.2
MODEL 3
5.2246 .031905 0
5.2680 -.029445 1
MODEL 4
5.2683 -.033376 1
6.5850 .133175 1
5.0450 -.018216 1
MODEL 5
6.4912 .118702 1
5.0463 -.018723 1
5.2682 -.033422 1
MODEL 6
7.2741 .384362 0
6.4147 .117147 1
5.0469 -.018792 1
5.2687 -.033175 1

MODE= 7 NAT. FREQ= 7.26663 RE-SAMPLE FACTOR= 4 ORIG. SAMPLE FREQ. = 150.2
MODEL 3
6.2727 .110399 1
5.2896 -.039026 1
MODEL 4

5.1122 -.015335 1
7.5330 .106581 1
5.2540 -.032529 1
MODEL 5
7.2423 .106086 0
5.0595 -.019073 1
5.2669 -.032638 0
MODEL 6
4.4404 .355047 1
5.2643 -.032839 1
7.3522 .109533 1
5.0699 -.017089 1

MODE= 8 NAT. FREQ= 9.40107 RE-SAMPLE FACTOR= 3 ORIG. SAMPLE FREQ. = 150.2
MODEL 3
6.3573 .121916 0
5.2755 -.037683 1
10.4524 .003213 1
MODEL 4
5.0757 -.014608 1
5.2665 -.032963 1
10.2978 .022802 1
10.4585 .003877 1
MODEL 5
8.5513 .607777 1
5.0682 -.017288 1
5.2682 -.032729 1
10.2217 .016902 1
10.4587 .003698 1
MODEL 6
6.2986 .192675 1
5.0523 -.017312 1
10.2221 .019284 1
5.2640 -.033534 1
10.4589 .003746 1

MODE= 9 NAT. FREQ= 9.43561 RE-SAMPLE FACTOR= 3 ORIG. SAMPLE FREQ. = 150.2
MODEL 3
6.6417 .131255 0
5.2531 -.030888 1
10.2630 .017290 1
MODEL 4
5.1100 -.007635 1
5.2756 -.032086 1
10.1917 .020735 1
10.4104 .008522 1
MODEL 5
5.1053 -.008146 1
5.2740 -.032206 1
10.2092 .021731 1
10.4464 .008656 1
MODEL 6
5.2699 -.033664 1
6.7337 .134047 1
5.0460 -.018332 0
10.2255 .019782 0

10.4580 .003747 0

ALL MATCHED MODAL PARAMETERS FROM ABOVE RESULTS

MODE	DAMPING FREQUENCY (HZ)	DAMPING FACTOR	# OF MATCHES
1.0	3.2668	.349334	1.0
2.0	5.0474	-.017189	3.0
2.0	5.2688	-.033234	3.0
3.0	5.0478	-.016974	3.0
3.0	5.2688	-.033233	3.0
4.0	5.0472	-.016760	1.0
4.0	5.2691	-.033224	3.0
4.0	5.5776	.049054	2.0
5.0	5.0511	-.017611	2.0
5.0	5.2697	-.033102	3.0
5.0	6.4773	.228936	2.0
6.0	5.0469	-.018792	3.0
6.0	5.2687	-.033175	3.0
6.0	6.4147	.117147	2.0
7.0	5.0699	-.017089	2.0
7.0	5.2643	-.032839	4.0
7.0	7.3522	.109533	2.0
8.0	5.0523	-.017312	2.0
8.0	5.2640	-.033534	3.0
8.0	10.2221	.019284	2.0
8.0	10.4589	.003746	3.0
9.0	5.0460	-.018332	5.0
9.0	5.2699	-.033664	3.0
9.0	6.7337	.134047	1.0
9.0	10.2255	.019782	3.0
9.0	10.4580	.003747	2.0

All parameters that regressively converged and converged with increasing model order.

Header information for final results determined from above data.

NUMBER OF MODES: 9
LAST POINT OF INPUT: 7
NUMBER OF POINTS READ IN AFTER INPUT: 159
FREE-STREAM DYANAMIC PRESSURE (0.5*RHO*(MACH*A)^2): 129.44746

FINAL ESTIMATED MODAL PARAMETERS

Final estimated results based upon above data.

MODE	NATURAL FREQUENCY (HZ)	DAMPING FREQUENCY (HZ)	DAMPING FACTOR	OTHER MODES*
1	3.18204	3.2668	.349334	
2	4.02851	5.0474	-.017189	3
3	4.05237	5.2688	-.033233	2
4	5.55739	5.5776	.049054	2 3
5	6.91297	6.4773	.228936	2 3
6	6.97122	6.4147	.117147	2 3
7	7.26663	7.3522	.109533	2 3
8	9.40107	10.2221	.019284	2 3 9
9	9.43561	10.4580	.003747	2 3 8

* OTHER MODES FOUND BUT NOT LIMITED TOO

WARNING: IF ABOVE RESULTS LOOK UNREASONABLE DUE TO IDENTICAL MODES

USE RESULTS FROM THE ARMA MODELS FOR EACH MODE SHAPE. IF IN THE POST-FLUTTER REGION AND AN USTABLE MODE IS SEEN FOR SEVERAL MODES THEN THE SYSTEM IS UNSTABLE AND MODES COULD NOT BE CLEARLY IDENTIFIED.

2

VITA

Corey L. Eckhart

Candidate for the Degree of

Master of Science

Thesis: USING ARMA MODELS TO IDENTIFY MODAL PARAMETERS FOR FLUTTER BOUNDARY PREDICTION

Major Field: Mechanical Engineering

Biographical:

Personal Data: Born in Tulsa, Oklahoma, on July 20, 1969, the son of Cary Eckhart and Diane Pense. Married Jennifer T. Aucoin, an Oklahoma State University graduate, on May 11, 1995.

Education: Graduated from Broken Arrow High School, Broken Arrow, Oklahoma, in May 1987; received Bachelor of Science degree in Mechanical Engineering (Aerospace Option) from Oklahoma State University, Stillwater, Oklahoma, in December, 1993. Completed the requirements for the Master of Science Degree with a major in Mechanical Engineering at Oklahoma State University in May, 1997.

Experience: A Senior Flight Test Engineer at Raytheon Aircraft in Wichita, Kansas from December, 1997 to Current; Senior Flight Test/Aeronautical Engineer at FlightSafety International in Tulsa, Oklahoma from May 1995 to June 1997; Aero-Mechanical Engineer at Aeromet, Inc. in Tulsa, Oklahoma from August 1993 to May 1995; Cooperative Education Certificate earned while working at Gulf States Utilities in St. Francisville, Louisiana, and American Airlines in Tulsa, Oklahoma, while an undergraduate.