NREL/TP-442-7110 • UC Category: 1213 • DE95009251

Advanced System Identification Techniques for Wind Turbine Structures with Special Emphasis on Modal Parameters

Jan T. Bialasiewicz

NREL Technical Monitor:
Richard M. Osgood

National Renewable Energy Laboratory
1617 Cole Boulevard
Golden, Colorado 80401-3393
A national laboratory of the U.S. Department of Energy
Managed by Midwest Research Institute
for the U.S. Department of Energy
under contract No. DE-AC36-83CH10093

Prepared under Task No. 429170

June 1995

Available to DOE and DOE contractors from:
Office of Scientific and Technical Information (OSTI)
P.O. Box 62
Oak Ridge, TN 37831
Prices available by calling (615) 576-8401

Available to the public from:
National Technical Information Service (NTIS)
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161
(703) 487-4650


Printed on paper containing at least 50% wastepaper, including 10% postconsumer waste

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

Foreword

The U. S. Department of Energy (DOE), in conjunction with the U. S. wind industry, is supporting the development of technology for advanced, higher efficiency wind energy conversion systems. As part of the Turbine Development Program, the DOE, through the National Renewable Energy Laboratory, is conducting several full system modal tests of advanced wind turbine systems. The objective of a modal survey is to identify the system's modal parameters, which have two main uses to wind researchers. First, in designing a large utility-grade wind turbine, knowledge of the modal frequencies, damping values, and mode shapes are essential to adjusting an analytical model so that it correctly predicts the dynamic response of the operating wind turbine. Second, modal survey results are useful when interpreting operating loads data, since knowledge of the system's natural frequencies helps determine the wind system's forced response. Field testing a full-size wind turbine presents special problems when compared to a modal survey of a structure conducted in the laboratory. The most significant problem incurred during a full system modal test is the lack of control over the environment. For example, wind of sufficient velocity blowing across the structure induces uncontrollable motion of the wind turbine. Wind researchers at Sandia National Laboratories have successfully used wind excitation in their Natural Excitation Technique (NExT) to extract some of the wind turbine's modal parameters including natural frequencies and damping values. However, because the wind input is not measured, the NExT does not yield an input-output model of the system (such as a transfer function or state-space representation) and clearly defined mode shapes corresponding to the structure's natural frequencies. This work supplements the field of full system modal testing by developing an advanced system identification method which excites the structure with a measured frequency-rich input and then derives a complete state-space representation of the structure. After computing a state-space model of the wind turbine structure, the system identification technique calculates the inherent modal parameters such as natural frequencies, damping ratios, and mode shapes. Also, the system identification technique can be used to adjust an analytical model with input-output time series obtained from experimental structural tests.

Richard M. Osgood, Technical Monitor
National Renewable Energy Laboratory

Contents

List of Figures

List of Tables

1.0 Introduction

The goal of this research is to develop advanced system identification techniques that can be used to accurately measure the frequency response functions of a wind-turbine structure immersed in wind noise. To allow for accurate identification, we have developed a special test signal called the **P**seudo-**R**andom **B**inary **S**equence (PRBS). The Matlab program that generates this signal allows the user to interactively tailor its parameters for the frequency range of interest based on the response of the wind turbine under test. By controlling NREL's Mobile Hydraulic Shaker System, which is attached to the wind turbine structure, the PRBS signal produces the wide-band excitation necessary to perform system identification in the presence of wind noise.

The techniques presented here will enable researchers to obtain modal parameters from an operating wind turbine, including frequencies, damping coefficients, and mode shapes. More importantly, the algorithms we have developed and tested (so far using input-output data from a simulated structure) permit state-space representation of the system under test, particularly the modal state space representation. This is the only system description that reveals the internal behavior the system, such as the interaction between the physical parameters, and which, in contrast to transfer functions, is valid for non-zero initial conditions.

The Sandia National Laboratory's (SNL) Natural Excitation Technique (NExT) for modal parameter extraction from operating wind turbines uses the measured system outputs obtained as a result of natural wind excitation [1]. Generally, the cross-correlation function of such outputs has the shape of the system's pulse response and therefore allows one to extract modal frequencies and damping ratios. SNL's researchers have done this using one of the system realization algorithms (such as the Eigensystem Realization Algorithm developed at the National Aeronautical and Space Administration (NASA) Langley Research Center). In other words, they assume that the cross-correlation function represents the sequence of Markov parameters or pulse response of the system to be identified. Unfortunately, such an assumption will not lead to any input-output model of the system, such as a transfer function or state-space representation, or mode shapes related to those models. To identify any of these input-output characteristics, it is not enough to excite the system with a frequency-rich signal (such as natural wind noise) one must also measure this signal. System identification requires a frequency-rich input-output history that is not incorporated in the SNL's approach.

Researchers have developed many system identification techniques and applied them to state-space models to identify modal parameters. Most techniques use sampled-pulse system response histories, known as system Markov parameters. The new approach, presented here, uses the results obtained by researchers at NASA's Langley Research Center [2-4]. Rather than identifying the system Markov parameters (which may exhibit very slow decay), one can use an asymptotically stable observer to form a stable, discrete state-space model to identify the system.

This report is organized as follows. In Section 2, we introduce the **O**bserver/**K**alman Filter **Id**entification technique (OKID) and show its relationship to a Kalman Filter. Section 3 presents the principles of eigensystem realization with a special emphasis on the relationship to modal parameters. We also introduce the Modal Amplitude Coherence and Modal Singular Values as the measures needed to distinguish true modes from noise modes. Because some measurement data may be noisier than others or sensors may malfunction during the test, these quantifying means are necessary if we are to consider only good strong measured signals without losing any capability. The system model with stochastic uncertainties. We represent these uncertainties are by process noise and output or measurement noise. We justify the need for strong correlation between the input test signal and the output and we recommend the PRBS as a test signal. In Section 5, we present the principles of generating and using the PRBS signal, as well as its properties. Finally, in Section 6, we give very condensed information on the simulation/identification programs developed in this research. Appendix A provides more detailed information on the programs, and Appendix B provides numerical results and some application notes on these programs.

2.0 Observer/Kalman Filter-Based Identification: Computation of Markov Parameters

A state-space model of a continuous-time system may be represented by the following equations:

$$\begin{aligned} \dot{x} &= A_c x + B_c u \\ y &= Cx + Du \end{aligned} \tag{2-1}$$

with r inputs and m outputs. The corresponding discrete-time representation is

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned} \tag{2-2}$$

where

$$A = e^{A_c T} = I + A_c T + \tfrac{1}{2!}(A_c T)^2 + \tfrac{1}{3!}(A_c T)^3 + \ldots \tag{2-3}$$

$$B = \int_0^T e^{A_c \tau}\, d\tau B_c = [IT + \tfrac{1}{2!} A_c T^2 + \tfrac{1}{3!} A_c^2 T^3 + \ldots] B_c \tag{2-4}$$

and T is the sampling interval. Assuming that $x(0)= 0$, we obtain

$$x(k) = \sum_{i=1}^{k} A^{i-1} B u(k-i) \tag{2-5}$$

$$y(k) = \sum_{i=1}^{k} CA^{i-1} B u(k-i) + Du(k) \tag{2-6}$$

It can be easily shown that equation (2-6) represents the convolution of the system's unit pulse response and the input sequence $u(k)$. In other words, as may be seen by calculating the response to the pulse input, defined as

$$u_i(0)=1 \quad , \quad i=1,2,...,r$$
$$u_i(k)=0 \quad , \quad i=1,2,...,r; \qquad k=1,2,...$$

$y(k)$ becomes an $m \times r$ pulse response matrix $Y(k)$ and

$$Y_0 = D,\ Y_1 = CB,\ Y_2 = CAB,\ ...,\ Y_k = CA^{k-1}B \tag{2-7}$$

These constant matrices are known as Markov parameters. The jth column of the Markov parameter matrix Y_i represents the pulse response at the ith time step with a unit pulse applied at the jth input. We can rewrite equation (2-6) in another form, expressing the output in terms of the input sequence and the Markov parameters:

$$y(k) = \sum_{i=0}^{k} Y_i u(k-i) \tag{2-8}$$

Let us assume that we must identify the system using l data points each consisting of r inputs and m outputs. Using equation (2-5) and assuming zero initial conditions, i.e., $x(0)=0$, we get

$$\begin{aligned} x(1) &= Bu(0) \\ x(2) &= ABu(0) + Bu(1) \\ &\vdots \end{aligned} \tag{2-9}$$

and then, using equation (2-6), we can obtain the sequence of l outputs

$$\begin{aligned} y(0) &= Du(0) \\ y(1) &= CBu(0) + Du(1) \\ y(2) &= CABu(0) + CBu(1) + Du(2) \\ &\vdots \end{aligned} \tag{2-10}$$

The set of l equations in (2-10) can be represented by one vector-matrix equation:

$$\underset{m\times l}{\left[y(0)\vdots y(1)\vdots \ldots \vdots y(l-1)\right]} = \underset{m\times rl}{\left[D\vdots CB\vdots CAB\vdots \ldots \vdots CA^{l-2}B\right]} \underset{rl\times l}{\begin{bmatrix} u(0) & u(1) & u(2) & \ldots & u(l-1) \\ & u(0) & u(1) & \ldots & u(l-2) \\ & & u(0) & \ldots & u(l-3) \\ & & & & \vdots \\ & & & & u(0) \end{bmatrix}} \qquad (2-11)$$

Equation (2-11), with the obvious denotations, can be rewritten in the following compact form:

$$\mathbf{y} = \mathbf{YU} \qquad (2\text{-}12)$$

As can be seen from equation (2-11), each row of the matrix **YU** is the product of the matrix **Y** and the proper u sequence of the length l (represented be a column of the matrix **U**), whose number of non-zero elements varies. In other words, each row of the matrix **YU** represents the convolution of the system's pulse response and the current input sequence, which changes its length from 1 to l.

The matrix **Y** contains all the Markov parameters to be determined. **U** and **y** are known matrices. However, the matrix **U** is square only in the case of a single input system, or r=1. This is the only case in which the solution for **Y** is unique. Conversely, it is known that **Y** must be unique for a finite-dimensional linear system. Even for r=1, the matrix **U** may become ill-conditioned and the solution $\mathbf{Y} = \mathbf{yU}^{-1}$ may not be attainable.

If we assume that the system is asymptotically stable, i.e., for sufficiently large p

$$A^k \approx 0 \quad \text{for} \quad k \geq p$$

then both **U** and **Y** can be truncated as follows:

$$\underset{m\times r(p+1)}{\mathbf{Y} = \left[D\vdots CB\vdots CAB\vdots \ldots \vdots CA^{p-1}B\right]}$$

$$\underset{r(p+1)\times l}{\mathbf{U} = \begin{bmatrix} u(0) & u(1) & \ldots & u(p) & \ldots & u(l-1) \\ & & & \vdots & & \vdots \\ & & & u(0) & \ldots & u(l-p-1) \end{bmatrix}}$$

Choosing the proper length l of the input-output sequence so that $l > r(p+1)$, we could approximately calculate p Markov parameters using pseudo-inverse $\mathbf{U}^{\dagger}$ of the matrix **U**:

$$\mathbf{Y} = \mathbf{y}\mathbf{U}^{\dagger} \tag{2-13}$$

Unfortunately, for lightly damped structures p and l are large and the size $r(p+1)\times l$ of $\mathbf{U}$ becomes impractically large to numerically solve for the pseudo-inverse.

A possible approach to solving this problem is to artificially increase system damping to obtain the solution for Markov parameters. The observer model of the system is used in this approach. The state equation (2-2) can be manipulated as follows:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + Gy(k) - Gy(k) \\ &= (A+GC)x(k) + (B+GD)u(k) - Gy(k) \end{aligned} \tag{2-14}$$

where G is an $n\times m$ arbitrary matrix chosen to make $A+GC$ as stable as desired. Equation (2-14) can be rewritten in a standard compact form:

$$x(k+1) = \overline{A}x(k) + \overline{B}v(k) \tag{2-15}$$

where

$$\overline{A} = A+GC, \quad \overline{B} = [B+GD \;\; -G], \quad v(k) = \begin{bmatrix} u(k) \\ y(k) \end{bmatrix}$$

The equation can now be written similar to (2-12), which involves observer Markov parameters:

$$\overline{\mathbf{Y}} = \left[D : C\overline{B} : C\overline{A}\overline{B} : \ldots : C\overline{A}^{p-1}\overline{B} : \ldots : C\overline{A}^{l-2}\overline{B}\right] \tag{2-16}$$

For an observable system, we can assign the eigenvalues of $\overline{A}$ arbitrarily through a proper choice of G. In the case of the dead-beat observer, or when all the eigenvalues of $\overline{A}$ are placed at the origin, $C\overline{A}^{k}\overline{B} \approx 0$ for $k \geq p$ where p is a sufficiently large integer. We can now solve for the observer Markov parameters

$$\overline{Y} = \left[D : C\overline{B} : C\overline{A}\overline{B} : \ldots : C\overline{A}^{p-1}\overline{B}\right] \tag{2-17}$$

using either a recursive or nonrecursive least-squares algorithm. Thus far, we have assumed zero initial conditions, which is never the case in practice. However, after the p steps, the initial conditions have a negligible effect on the measured data (because they are multipled by $\overline{A}^{p}$).

The observer Markov parameters in equation (2-17) include the system Markov parameters and the observer gain Markov parameters. The system Markov parameters are used to compute the system matrices A, B, C, and D, whereas the observer gain Markov parameters are used to determine the observer gain matrix G. The proper algorithm for

obtaining these Markov parameters has been introduced by Phan et al. [5] and is also discussed by Juang [2]. The algorithm uses input and output time histories as its input. Software implementation of this identification algorithm was developed at NASA Langley and is known as the Matlab program OKID. Finally, the state-space representation (A, B, C, D) of the system is obtained using the Eigensystem Realization Algorithm (ERA), based on system realization theory. An introduction to these topics is presented in Section 3.

In order to consider system uncertainties stochastically, one should add the process noise $w(k)$, which includes system uncertainties and input noise, and the measurement (or output) noise $v(k)$. This consideration leads to the following modification of equation (2-2):

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + w(k) \\ y(k) &= Cx(k) + Du(k) + v(k) \end{aligned} \tag{2-18}$$

The noises added can be specified either in terms of their statistics or in terms of the Kalman filter model which can be written as

$$\begin{aligned} \hat{x}(k+1) &= A\hat{x}(k) + Bu(k) + K\varepsilon_r(k) \\ \hat{y}(k) &= C\hat{x}(k) + Du(k) \end{aligned} \tag{2-19}$$

for the system (2-18) where $\hat{x}(k)$ is the estimated state, K is the Kalman filter gain, and $\varepsilon_r(k)$, called the residual, is defined as the difference between the real measurement $y(k)$ and the estimated measurement $\hat{y}(k)$. The Kalman filter representation of a process is valid if both $w(k)$ and $v(k)$ are Gaussian, zero-mean, and white with different covariance matrices and if they are statistically independent of each other. Equation (2-19) can be represented as

$$\begin{aligned} \hat{x}(k+1) &= \tilde{A}\hat{x}(k) + \tilde{B}v(k) \\ y(k) &= C\hat{x}(k) + Du(k) + \varepsilon_r(k) \end{aligned} \tag{2-20}$$

where

$$\tilde{A} = A - KC, \quad \tilde{B} = [B - KD \vdots K] \ , \quad v(k) = \begin{bmatrix} u(k) \\ y(k) \end{bmatrix}$$

Comparison of equations (2-15) and (2-20) reveals that they are identical if $G = -K$ and $\varepsilon_r(k)=0$, as are their Markov parameters. It can also be proven that the truncated observer model (2-17), obtained as a result of the dead-beat approximation of equation (2-15), produces the same input-output map as a Kalman filter if the data length is sufficient so that the truncation error is negligible. In this case, G computed from combined Markov parameters of equation (2-17) gives the steady-state Kalman filter gain $K = -G$.

In practice, because other factors such as disturbances, nonlinearities, non-whiteness of the process, and measurement noises are present, the resultant identified filter is not the Kalman filter. It is simply an observer that is computed from input-output data that minimize the filter residual in a least-squares sense.

3.0 Introduction to Eigensystem Realization

Consider Markov system parameters given in equation (2-7). Because $D = Y_0$, a realization will involve the computation of a triplet [A, B, C] from the Markov parameters, for which the discrete-time model (2-2) is satisfied. We look for a minimum realization, i.e., for a model with the smallest state space dimensions among all realizable systems that have the same input-output relations. All minimum realizations have the same set of eigenvalues.

If $(\psi_1, \psi_2, \ldots, \psi_n)$ is a complete set of the eigenvectors of the state matrix A of order n with corresponding eigenvalues $(\lambda_1, \lambda_2, \ldots, \lambda_n)$, then we can define two matrices $\Lambda = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$ and $\Psi = (\psi_1, \psi_2, \ldots, \psi_n)$. The realization [$A$, B, C] can be transformed to the modal realization [Λ, $\Psi^{-1}B$, $C\Psi$] with $\Lambda = \Psi^{-1}A\Psi$ and the modal state vector x_m defined by the equation $x = \Psi x_m$. In this realization, Λ specifies modal damping rates and damped natural frequencies, the matrix $B_m = \Psi^{-1}B$ defines the initial modal amplitudes, and the matrix $C_m = C\Psi$ defines the mode shapes at the sensor points. Let us analyze what are the modal contributions to the sequence of Markov parameters. We have

$$B_m = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}, \quad C_m = \begin{bmatrix} c_1 & c_2 & \ldots & c_n \end{bmatrix}$$

where b_i's are r-vectors and c_i's are m-vectors. Therefore, for modal state-space representation we have the following Markov parameter sequence:

$$\begin{aligned} Y &= [D \;\; C_m B_m \;\; C_m \Lambda B_m \ldots C_m \Lambda^{k-1} B_m] \\ &= D \;\; \sum_{i=1}^{n} c_i b_i \;\; \sum_{i=1}^{n} c_i \lambda_i b_i \;\; \sum_{i=1}^{n} c_i \lambda_i^2 b_i \;\; \ldots \;\; \sum_{i=1}^{n} c_i \lambda_i^{k-1} b_i \Big] \\ &= \begin{bmatrix} Y_0 & Y_1 & Y_2 & \ldots & Y_k \end{bmatrix} \end{aligned} \tag{3-1}$$

The sequence of each ith component in the sums above forms a function of time as defined by the

$$\begin{bmatrix} c_i b_i & c_i \lambda_i b & c_i \lambda_i^2 b_i & \ldots & c_i \lambda_i^{k-1} b_i \end{bmatrix} \tag{3-2}$$

where k is the number of samples and $i = 1,2,...,n$. This sequence represents the contribution of the *i*th mode to the system's Markov parameters. Note that all components in the sequences (3-1) and (3-2) are $m \times r$ matrices. Then

$$q_i \triangleq \begin{bmatrix} b_i & \lambda_i b_i & \lambda_i^2 b_i & \cdots & \lambda_i^{k-1} b_i \end{bmatrix}, \quad i = 1,2,...,n \tag{3-3}$$

is the modal amplitude time history. The sequence q_i represents the temporal contribution of the *i*th mode associated with the output martrix c_i to the Markov parameter sequence *Y*.

According to the relation (2-3)

$$\Lambda = e^{\Lambda_c T} \tag{3-4}$$

Therefore, the modal damping rates and damped natural frequencies of the continuous-time system are simply real and imaginary parts of the eigenvalues, which are the diagonal elements of the matrix

$$\Lambda_c = T^{-1} \ln(\Lambda) \tag{3-5}$$

System realization begins by forming the generalized $\alpha m \times \beta r$ Hankel matrix $H(k\text{-}1)$, composed of Markov parameters Y_i:

$$H(k-1) = \begin{bmatrix} Y_k & Y_{k+1} & \cdots & Y_{k+\beta-1} \\ Y_{k+1} & Y_{k+2} & \cdots & Y_{k+\beta} \\ \vdots & \vdots & \cdots & \vdots \\ Y_{k+\alpha-1} & Y_{k+\alpha} & \cdots & Y_{k+\beta+\alpha-2} \end{bmatrix} \tag{3-6}$$

If a system is controllable and observable, then for $\alpha \geq n$ and $\beta \geq n$ (the order of the system), the matrix $H(k\text{-}1)$ is of rank *n*. Recall that Y_k is an $m \times r$ matrix and that

$$Y_k = CA^{k-1}B$$

Consequently, we have the following decomposition:

$$H(k-1) = P_\alpha A^{k-1} Q_\beta \tag{3-7}$$

where

$$P_\alpha = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{\alpha-1} \end{bmatrix}, \quad Q_\beta = \begin{bmatrix} B & AB & A^2B & \cdots & A^{\beta-1}B \end{bmatrix} \tag{3-8}$$

This decomposition shows that $H(k\text{-}1)$ has the rank of n if the represented nth order system is controllable and observable. We can also note that, in particular

$$H(0) = P_\alpha Q_\beta \tag{3-9}$$

The ERA process starts with the factorization of $H(0)$, using singular value decomposition

$$H(0) = R\Sigma S^T \tag{3-10}$$

where

$$R^T R = I,\ S^T S = I,\ \Sigma = \begin{bmatrix} \Sigma_n & 0 \\ 0 & 0 \end{bmatrix} \tag{3-11}$$

with

$$\Sigma_n = \text{diag}[\sigma_1, \sigma_2, \ldots, \sigma_n],\ \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \tag{3-12}$$

If R_n and S_n are the matrices formed by first n columns of R and S, respectively, then

$$H(0) = R_n \Sigma_n S_n^T \quad \text{and} \quad R_n^T R_n = I = S_n^T S_n \tag{3-13}$$

Comparing equations (3-9) and (3-13), we can make the following obvious balanced choice:

$$P_\alpha = R_n \Sigma_n^{1/2} \quad \text{and} \quad Q_\beta = \Sigma_n^{1/2} S_n^T \tag{3-14}$$

Taking equation (3-8) into consideration, we can conclude that the first r columns of Q_β form the input matrix B and the first m rows of P_α form the output matrix C. Therefore, defining the matrix $E_i^T = \begin{bmatrix} I_i & O_i & \ldots & O_i \end{bmatrix}$ where O_i is a null matrix of order i and I_i is an identity matrix of order i, we can state that

$$B = \Sigma_n^{1/2} S_n^T E_r \quad \text{and} \quad C = E_m^T R_n \Sigma_n^{1/2} \tag{3-15}$$

where r is the number of inputs and m is the number of outputs. The matrix A is obtained from $H(1)$. Considering equations (3-7) and (3-14), we obtain

$$H(1) = P_\alpha A Q_\beta = R_n \Sigma_n^{1/2} A \Sigma_n^{1/2} S_n^T \tag{3-16}$$

and

$$A = \Sigma_n^{-1/2} R_n^T H(1) S_n \Sigma_n^{-1/2} \tag{3-17}$$

If there was no noise, the rank of $H(k)$ would be equal to the true order of the system under test. In practice, due to noise, nonlinearities, and computer round-off errors, $H(k)$ will be of full rank. However, some singular values, say $\sigma_{i+1}, \ldots, \sigma_n$, may be negligible.

Therefore, the reduced model of order i can be identified as a model of the system under test [2].

The above discussion suggests that the system realized from equations (3-15) and (3-16) is an approximation of the real system. That is, we are obtaining from these equations some estimates $\hat{A}, \hat{B}$, and $\hat{C}$ of the real matrices A, B, and C. Therefore, we need to distinguish true modes from noise modes. This can be done using the approach described below, which was first proposed by Juang [2].

Using the identified matrices $\hat{A}, \hat{B}$, and $\hat{C}$, we can determine the estimate $\hat{Y}$ of Y. According to equations (3-1) and (3-3), this has the following form:

$$\hat{Y} = \begin{bmatrix} D & \sum_{i=1}^{n} \hat{c}_i \hat{q}_i \end{bmatrix} \tag{3-18}$$

where the sequence $\hat{q}_i$ represents the identified modal amplitude time history. The sequence $\hat{q}_i$ is computed from the identified eigenvalue $\hat{\lambda}_i$ and the input matrix $\hat{b}_i$.. Another way to obtain modal amplitude time history is to form and then decompose the Hankel matrix

$$H(0) = \begin{bmatrix} Y_1 & Y_2 & \cdots & Y_{l-\alpha} \\ Y_2 & Y_3 & \cdots & Y_{l-\alpha+1} \\ \vdots & \vdots & \ddots & \vdots \\ Y_\alpha & Y_{\alpha+1} & \cdots & Y_{l-1} \end{bmatrix} \tag{3-19}$$

where the l Markov parameters involved are directly obtained from pulse response samples, and α is chosen such that $m\alpha$ is greater than or equal to the system order. The singular value decomposition is used to determine the matrix $\overline{Q}$, defined by the following equation:

$$H(0) = \left[R_n \Sigma_n^{1/2} \Psi \right] \left[\Psi^{-1} \Sigma_n^{1/2} S_n^T \right] \triangleq \overline{P}\,\overline{Q} \tag{3-20}$$

where Ψ is chosen to be the eigenvector matrix of the estimated state matrix $\hat{A}$ because the real state matrix A is unknown. Using modal state space representation (Λ, B_m, C_m), as defined above, we can make the following approximation:

$$\overline{Q} = \begin{bmatrix} \overline{q}_1 \\ \overline{q}_2 \\ \vdots \\ \overline{q}_n \end{bmatrix} \cong \begin{bmatrix} B_m & \Lambda B_m & \cdots & \Lambda^{l-\alpha-1} B_m \end{bmatrix} \tag{3-21}$$

or, in the expanded form:

$$\overline{Q} = \begin{bmatrix} \overline{q}_1 \\ \overline{q}_2 \\ \vdots \\ \overline{q}_n \end{bmatrix} \cong \begin{bmatrix} [b_1 & \lambda_1 b_1 & \cdots & \lambda_1^{l-\alpha-1} b_1] \\ [b_2 & \lambda_2 b_2 & \cdots & \lambda_2^{l-\alpha-1} b_2] \\ & & \vdots & \\ [b_n & \lambda_n b_n & \cdots & \lambda_n^{l-\alpha-1} b_n] \end{bmatrix} \tag{3-22}$$

It is clear that $\overline{q}_i$ and $\hat{q}_i$ $(i = 1,2,3,\ldots,n)$ are identical for the noise free case. With noise present and some small singular values truncated, $\hat{q}_i$ should be considered an approximation of $\overline{q}_i$. Therefore, the Modal Amplitude Coherence (MAC) can be defined as the following dot product of these two vectors:

$$MAC_i = \left|\overline{q}_i \hat{q}_i^*\right| / \left(\left|\overline{q}_i \overline{q}_i^*\right| \left|\hat{q}_i \hat{q}_i^*\right|\right)^{1/2} \quad , \quad i=1,2, \ldots n \tag{3-23}$$

where the superscript * denotes transpose and complex conjugate. If these two vectors coincide, then the model reproduces the pulse response data.

In addition, the system and noise modes can be quantified through the determination of each modal coordinate contribution to the pulse response of the model. Because the modal coordinates of the model are defined by

$$\left[\hat{c}_i \hat{b}_i \quad \hat{c}_i \hat{\lambda}_i \hat{b}_i \quad \hat{c}_i \hat{\lambda}_i^2 \hat{b}_i \quad \ldots \quad \hat{c}_i \hat{\lambda}_i^{k-1} \hat{b}_i\right]; \; i = 1,2,\ldots,n \tag{3-24}$$

their contributions can be determined by the Mode Singular Value given as

$$\begin{aligned} MSV_i &= \sqrt{\left|\hat{c}_i\right| \quad \left(1 + \left|\hat{\lambda}_i\right| + \left|\hat{\lambda}_i^2\right| + \ldots + \left|\hat{\lambda}_i^{l-1}\right|\right) \quad \left|\hat{b}_i\right|} \\ &\approx \sqrt{\frac{\left|\hat{c}_i\right| \quad \left|\hat{b}_i\right|}{1 - \left|\hat{\lambda}_i\right|}} \end{aligned} \tag{3-25}$$

The approximation is valid if all eigenvalues λ_i are inside the unit circle and the number of Markov parameters l is sufficiently large. The tests described here are used by the ERA developed at NASA Langley.

4.0 Frequency Response Function and Input-Output Correlation

Let us consider a system with no input forces before the starting time t=0 (implying that the input matrix **U** becomes an upper block triangular). In this case, we are considering the input-output relationship described for the input-output sequence of the length l by equation (2-11), represented by equation(2-12) in compact form. This representation is repeated here for convenience:

$$\underset{m\times l}{\mathbf{y}} = \underset{m\times rl}{\mathbf{Y}} \quad \underset{rl\times l}{\mathbf{U}} \tag{4-1}$$

Post-multiplying (4-1) by $\mathbf{U}^T$ and averaging both sides over the number of samples l, one obtains the following relation:

$$\underset{m\times rl}{R_{yU}} = \underset{m\times rl}{\mathbf{Y}} \quad \underset{rl\times rl}{R_{UU}} \tag{4-2}$$

where

$$R_{UU} = \begin{bmatrix} R_{uu}(0) & R_{uu}(1) & \cdots & R_{uu}(l-1) \\ R_{uu}(1) & R_{uu}(0) & \cdots & R_{uu}(l-2) \\ \vdots & \vdots & \ddots & \vdots \\ R_{uu}(l-1) & R_{uu}(l-2) & \cdots & R_{uu}(0) \end{bmatrix} \tag{4-3}$$

and

$$R_{yU} = \begin{bmatrix} R_{yu}(0) & R_{yu}(1) & \cdots & R_{yu}(l-1) \end{bmatrix} \tag{4-4}$$

with

$$R_{uu}(i) = \frac{1}{l}\sum_{k=0}^{l-1} u(k)u^T(k-i) \tag{4-5}$$

and

$$R_{yu}(i) = \frac{1}{l}\sum_{k=0}^{l-1} y(k)u^T(k-i) \tag{4-6}$$

Our goal is to solve equation (4-2) for $\mathbf{Y}$. However, this requires inversion of the matrix R_{UU}. If the input signal is not rich enough in frequencies, the matrix U and, consequently, the matrix R_{UU} become rank deficient and cannot be inverted. Averaging over many records (each of the length l) usually solves the problem [2]. Another possibility is to use a frequency- rich input signal, such as the PRBS signal of sufficient length l, which is discussed in detail in Section 5. For such a signal,

$$R_{UU} \cong \text{diag}(R_{uu}(0),\ R_{uu}(0),\ \ldots,\ R_{uu}(0)) \tag{4-7}$$

Because $\mathbf{Y} = [Y_0, Y_1, \ldots, Y_{l-1}]$, equation (4-2) can be represented by the following set of equations:

$$R_{yu}(i) = \sum_{k=0}^{l-1} Y_k R_{uu}(i-k)\ , \quad i = 0,1,2,\ldots,l-1 \tag{4-8}$$

For a PRBS input signal applied at all r inputs, equation (4-8) assumes the following form:

$$R_{yu}(k) \cong Y_k R_{uu}(0) \quad , \quad k = 0,1,2,...,l-1 \tag{4-9}$$

If we assume that for this input signal, the $r \times r$ diagonal matrix $R_{uu}(0) = \text{diag}(\rho,\rho,...,\rho)$, then for the PRBS input signal,

$$Y_k \cong \frac{1}{\rho} R_{yu}(k) \quad , \quad k = 0,1,2,...,l- \tag{4-10}$$

4.1 System with Stochastic Uncertainties

Thus far, we have considered the system model without process noise or input/output noise. Now, let us go back to the model represented by equation (2-18) and repeated here for convenience:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + w(k) \\ y(k) &= Cx(k) + Du(k) + v(k) \end{aligned} \tag{4-11}$$

where $v(k)$ and $w(k)$ are both zero-mean sequences, uncorrelated with the input signal $u(k)$. Assuming zero initial conditions, the system output is given by the following relationship:

$$y(k) = \sum_{i=0}^{\infty} \{g_i u(k-i) + h_i w(k-i)\} + v(k) \tag{4-12}$$

where

$$g_0 = D \ , \ g_i = CA^{i-1}B \ , \ h_0 = 0 \ , \ h_i = CA^{i-1} \ , \ i = 1,2,... \tag{4-13}$$

This equation can be converted to its correlation form (i.e., post-multiplied by $u^T(k-l)$ and summed over all samples k) to yield

$$R_{yu}(l) = \sum_{i=0}^{\infty} g_i R_{uu}(l-i) \tag{4-14}$$

If u(k) is a PRBS signal, this will lead again to the approximate result found in equations (4-9) or (4-10).

4.2 Frequency Response Function (FRF) from Correlation Discrete Fourier Transform (DFT)

Using DFT in equation (4-14) yields [2]

$$S_{yu}(k) = G(z_k)S_{uu}(k) - S_{eu}(k) \tag{4-15}$$

where

$$G(z_k) = \sum_{\tau=0}^{\infty} Y_\tau e^{-j\frac{2\pi k}{l}\tau}, \quad z_k = e^{-j\frac{2\pi k}{l}}$$

$$S_{yu}(k) = \frac{1}{l}\sum_{\tau=0}^{l-1} R_{yu}(\tau)e^{-j\frac{2\pi k}{l}\tau}$$

$$S_{uu}(k) = \frac{1}{l}\sum_{\tau=0}^{l-1} R_{uu}(\tau)e^{-j\frac{2\pi k}{l}\tau}$$

$S_{eu}(k) = 0$ when $R_{uu}(\tau)$ is periodic

For the single input and single output system,

$$G(z_k) = \frac{S_{yu}(k)}{S_{uu}(k)} \tag{4-16}$$

where $G(z_k)$ is the system transfer function at the frequency $\omega = \frac{2\pi k}{l}$.

The inverse transform of the frequency response function $G(z_k)$ is given by

$$Y_\tau = \sum_{\tau=0}^{\infty} G(z_k)e^{j\frac{2\pi\tau}{l}k} \tag{4-17}$$

where $Y_0 = D$, $Y_1 = CB$, $Y_2 = CAB$, ..., $Y_\tau = CA^{\tau-1}B$, ... form a sequence of system Markov parameters. In system identification, it is a common practice to compute the frequency response function and then convert it to system Markov parameters. In practice, we assume that N data records of the length l are available; therefore, the average values $\overline{S}_{yu}(k)$ and $\overline{S}_{uu}(k)$, defined as

$$\overline{S}_{yu} = \frac{1}{N}\sum_{i=0}^{N} S_{yu}^{(i)} \quad \text{and} \quad \overline{S}_{uu} = \frac{1}{N}\sum_{i=0}^{N} \overline{S}_{uu}^{(i)} \tag{4-18}$$

where the superscript (i) refers to the ith experiment, are used in equation (4-16).

It is necessary to qualify the frequency response function calculated using the procedure outlined above. The coherence function is commonly used for this purpose. The coherence function between the input u_ν and the output y_μ is defined as

$$\gamma^2_{y_\mu u_\nu}(k) = \frac{\overline{S}_{y_\mu u_\nu}(k)\overline{S}^*_{y_\mu u_\nu}(k)}{\overline{S}_{y_\mu y_\mu}(k)\overline{S}_{u_\nu u_\nu}(k)} \tag{4-19}$$

The following so called cross-spectrum inequality holds:

$$\overline{S}_{y_\mu y_\mu}(k)\overline{S}_{u_\nu u_\nu}(k) \geq \overline{S}_{y_\mu u_\nu}(k)\overline{S}^*_{y_\mu u_\nu}(k) \tag{4-20}$$

Therefore, the coherence function defined by equation (4-19) satisfies the following inequality:

$$\geq \gamma^2_{y_\mu u_\nu}(k) = \frac{\overline{S}_{y_\mu u_\nu}(k)\overline{S}^*_{y_\mu u_\nu}(k)}{\overline{S}_{y_\mu y_\mu}(k)\overline{S}_{u_\nu u_\nu}(k)} \geq 0 \tag{4-21}$$

for all k.

Because the system with input and output noise is described by the following equation:

$$y(k) = \sum_{\tau=0}^{\infty} Y_\tau u(k-\tau) + \varepsilon(k) \tag{4-22}$$

we can show that, for a single input and single output plant with the noise uncorrelated with the input signal, the following equality holds:

$$\gamma^2_{y_1 u_1}(k) = \frac{\overline{S}_{y_1 u_1}(k)\overline{S}^*_{y_1 u_1}(k)}{\overline{S}_{y_1 y_1}(k)\overline{S}_{u_1 u_1}(k)} = 1 - \frac{\overline{S}_{\varepsilon_1 \varepsilon_1}(k)}{\overline{S}_{y_1 y_1}(k)} \tag{4-23}$$

The ratio of $\overline{S}_{\varepsilon_1\varepsilon_1}(k)/\overline{S}_{y_1 y_1}(k)$ determines how close the coherence function is to unity. The stronger the output signal, the closer the coherence function is to unity (with the obvious assumption that the noise level remains unchanged). In addition, it can be shown that in the case of a plant with two inputs and one output, the sum of two coherence functions $\gamma^2_{y_1 u_1}$ and $\gamma^2_{y_1 u_2}$ is

$$\gamma^2_{y_1 u_1} + \gamma^2_{y_1 u_2} = -\frac{\overline{S}^*_{\varepsilon_1 \varepsilon_1}(k)}{\overline{S}_{y_1 y_1}(k)} \tag{4-24}$$

if inputs and noise are uncorrelated. This sum becomes unity when the output noise is negligible. In this ideal case, the unity will be equally shared by the two coherence

functions if the two inputs produce equal output spectral densities. If the input and output are absolutely uncorrelated, the coherence function will be zero.

Based on the discussion above, it is important to note that if the coherence function is greater than zero but less than unity, one or more of the following three situations exists:

1. Extraneous noise is present in the measurements.
2. The system has significant uncertainties and nonlinearities.
3. The output was generated by more than one input.

The coherence function is not as good a discriminator for multiple inputs as it is for the single input case; therefore, modal testing of flexible structures should be performed for each input with a collocated pair of an actuator and sensor. The coherence function for each input is made as close as possible to unity by moving around its corresponding actuator and sensor. In this way, modal testing with multiple inputs can be conducted with confidence in the coherence function.

5.0 Random Binary Process and the PRBS as a Test Signal for Process Identification

Good parameter identification requires the application of a frequency-rich or broad-band input. Such an input is provided by the PRBS signal. The sequence is generated by a digital waveform generator, which produces a binary signal by switching randomly between two output levels. It owes its name *pseudo-random* to the fact that it is characterized by a *sequence length* within which the pulse width varies randomly, while it is periodic over a larger time horizon. The period is defined by the sequence length. The random binary process can be considered a prototype of the periodic PRBS.

5.1 Properties of a Random Binary Process

Let us assume that $u(t)$ is a random binary process with the current value of a or $-a$. The value of $u(t)$ can change every T_{prbs} seconds; i.e, T_{prbs} is the switching period. The "*prbs*" subscript is used here because generation of a PRBS test signal with the same switching principle but with periodicity added is our ultimate goal. If u_1 is the value of $u(t)$ at time t, and u_2 is its value at time $t+\tau$, then we can characterize this two random variables by the joint probability density function $p(x_1,x_2)$. If $\tau=0$, then both random variables are either a or $-a$, or we can say that $p(x_1,x_2)$ consists of two delta functions, each of magnitude ½ at $u_1=u_2=a$ and $u_1=u_2=-a$. If $\tau > T_{prbs}$, then there is no correlation between u_1 and u_2 because u_2 is the result of a different random switch. Therefore, $p(x_1,x_2)$ will consist of four delta functions, each of magnitude ¼ at (a,a), $(-a,a)$, $(a,-a)$, and $(-a,-a)$. Now suppose that $\tau < T_{prbs}$. In this case, we must consider the initial phase shift ϕ between u_1 and u_2. If (because the process is stationary) we assume for convenience that $t=0$, then u_1 and u_2 will be correlated if $\phi > \tau$ (because the next random switch will not have occured at time

$t+\tau=\tau$) and uncorrelated if $\phi < \tau$. Because $0 \le \phi \le T_{prbs}$, we have $p(\phi) = 1/T_{prbs}\ \phi < \tau$. Then, the probability that $\phi < \tau$ is τ/T_{prbs} and the probability that $\phi > \tau$ is $1 - 1/T_{prbs}$. This leads to the following autocorrelation function:

$$R_{uu}(\tau) = \begin{cases} a^2(1-\tau/T_{prbs}) & \text{for } 0 \le \tau \le T_{prbs} \\ 0 & \text{for } \tau > T_{prbs} \end{cases} \tag{5-1}$$

The corresponding spectral density is

$$\begin{aligned} S_{uu}(\omega) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} R_{uu}(\tau)e^{-j\omega\tau}\,d\tau \\ &= \frac{a^2}{\pi}\frac{T_{prbs}}{2}\left(\frac{\sin(\omega T_{prbs}/2)}{\omega T_{prbs}/2}\right)^2 \end{aligned} \tag{5-2}$$

This spectral density can be assumed to be approximately flat up to a frequency of about $0.3f_{prbs}$. If f_{prbs} is sufficiently high (as compared to the bandwidth of a plant to be identified), then the random binary process has a spectrum corresponding to broad-band noise.

5.2 Generation of the Pseudo-Random Binary Sequence

The PRBS signal $u(k)$ is a periodic signal with the period LT_{prbs} (for example). Provided that the mean value $E[u]$ is zero, the autocorrelation of this sequence will be the autocorrelation function (5-1) repeated with period LT_{prbs} ; i.e.,

$$R_{uu}^{prbs}(\tau) = \sum_{m=-\infty}^{\infty} R_{uu}(\tau - mLT_{prbs}) \tag{5-3}$$

Then, it can be shown [6] that the spectrum of this type of signal is a comb of delta functions of varying height, each spaced $\Delta\omega = 2\pi/LT_{prbs}$ apart and each with magnitude $2\pi/LT_{prbs}$ times the ordinate of $S_{uu}(\omega)$ at the same frequency. In other words, we have

$$S_{uu}^{prbs}(\omega) = \frac{2\pi}{LT_{prbs}} S_{uu}(\omega) \sum_{k=-\infty}^{\infty} \delta(\omega - \frac{2\pi k}{LT_{prbs}}) \tag{5-4}$$

Provided that the sequence length L is large enough, the spacing between adjacent "spikes" ($2\pi/LT_{prbs}$ rad/s) is much less than the clock frequency ($2\pi/T_{prbs}$ rad/s) of the PRBS generator. Although only discrete frequencies occur, they are so closely spaced that the characteristics of broad-band noise are approximately preserved for frequencies less than about $0.3f_{prbs}=0.6\pi/\Delta t$.

The spectrum of the pseudo-random binary signal is therefore an approximation for broad band noise, provided that its clock frequency is fast enough and its sequence length is large enough.

The PRBSs are generated by means of shift registers with feedback (implemented in hardware or software). The maximum length L of a sequence is

$$L = 2^N - 1 \tag{5-5}$$

where N is the number of stages of the shift register. This equation reflects the fact that at least one of the N stages of the shift register should have an initial logic value of 1.

The PRBS generator includes an N-stage shift register, an adding device, and a clock. These components work in the following way. Each stage of the shift register stores a digit: 0 or 1. At every pulse from the clock, all digits are moved one place to the right. The digit in the last (right-hand) stage moves out to the output sequence register while the empty space in the first stage is filled by a new digit obtained by adding modulo 2 (before the clock pulse arrived) the digits previously present in the stages number l_i and l_h, specified for the selected N in Table 5.1. The pattern of every N adjacent digits of the output sequence is then unique, allowing us to approach the ideal of random switching. Typically, a shift register with N=10 stages gives an output sequence that is 2^{10}-1=1023 digits long, with every pattern of 10 adjacent digits being unique. This is not quite the same thing as a signal built up as a result of tossing more than one thousand coins (where more than 10 successive heads or tails may occur), but it is a good approximation. Such a sequence has one more state 1 than it has state 0. Therefore, we can say that the mean value $E(u) \cong 0$, as required.

Table 5.1 Generation of maximum-length PRBS

Number of Stages (N)	Sequence Length ($L = 2^N - 1$)	Stages Added (l_i and l_h)
4	15	3 and 4
5	31	3 and 5
6	63	5 and 6
7	127	4 and 7
8	255	3 and 8
9	511	5 and 9
10	1023	7 and 10

The Matlab program prbsm.m, generating PRBS, is included in Appendix A. Program users choose the magnitude a and the number of stages N of the shift register, which can be equal to any integer between 4 and 10 inclusive. A typical waveform generated is

shown in Figure 5-1 for N=5 and A=1. The Matlab program prbsm.m automatically plots the generated PRBS and stores the sequence in the PRB vector. The quality the PRBS as an input signal for process identification has been tested for all values of N through the calculation of its autocorrelation function. The results of this testing, i.e., the autocorrelation function of the PRBS for N=7, 8, 9, and 10 in terms of delay time in sampling periods, are shown in Figure 5-2.

5.3 Choosing a PRBS for Process Identification

If the process bandwidth considered is limited by the frequency $f_{\max}$, then (according to sampling theorem) discrete-time representation of this process requires a sampling frequency of $f_s > 2f_{\max}$. The rule of thumb is to choose $f_s = (6 \text{ to } 25) f_{\max}$ with f_s as high as possible. On the other hand, in order to correctly identify the steady-state gain of the process, the duration of at least one of the pulses in the PRBS must be greater than the rise time t_R of the process. Because the maximum duration of a pulse is NT_{prbs}, we have the following inequality:

$$NT_{prbs} > t_R \tag{5-6}$$

Therefore, the clock frequency f_{prbs} for the PRBS must be chosen as a submultiple of the sampling frequency f_s. If $f_{prbs}=f_s/p$ (p=1,2,...), then $T_{prbs} = T_s p$ and, combining this with the inequality (5-6), we obtain the following condition that must be fulfilled:

$$N > \frac{t_R}{T_s p} = \frac{t_R}{T_{prbs}} \tag{5-7}$$

Because lowering the clock frequency of the PRBS will reduce the frequency range corresponding to a constant spectral density, to choose $p \le 4$ [7] is recommended.

The proper signal/noise ratio should be maintained even when the magnitude a of PRBS is very low. The output variations due to PRBS should be larger than the residual noise level. Also, for a low signal/noise ratio, it is necessary to extend the test duration in order to obtain satisfactory parameter estimates. The significant increase in the PRBS level may be undesirable because of plant nonlinearities.

Let us assume that, for the process to be identified, $f_{\max}$=5 Hz. The time constant of such a process is approximately equal to 0.2 s and the rise time $t_R \cong 0.4$ s. Therefore, if we assume N=10, we should have $pT_s > 0.04$. Because we want the sampling interval to be as small as practical, we shall choose p=4 and T_s=0.01.

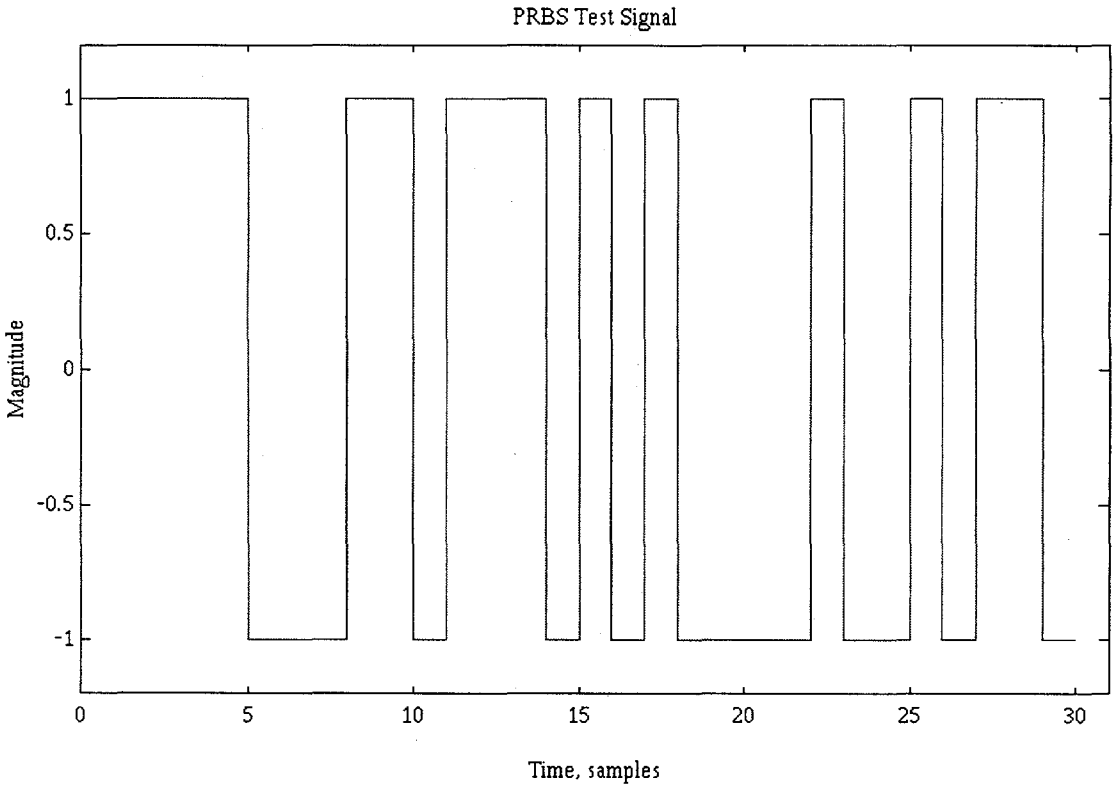


Figure 5-1 A typical PRBS waveform.

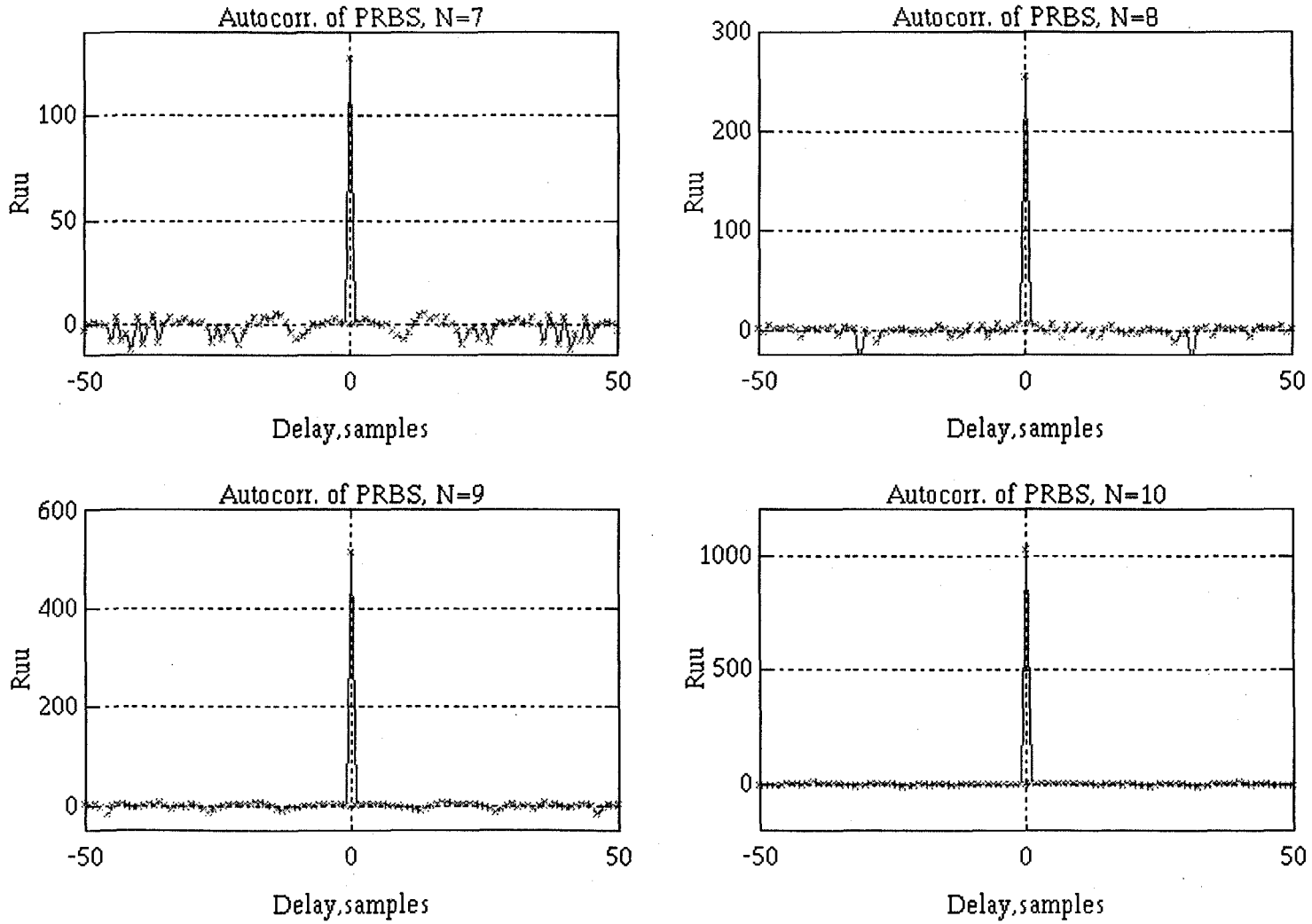


Figure 5-2 Autocorrelation function of the PRBS.

6.0 Process Simulation/Identification Programs and Test Results

Several process simulation/identification Matlab programs have been developed as a result of this research. Identification algorithms that have been implemented include the observer/Kalman filter algorithm and the state vector realization algorithm. Both of them identify state space representation of a process and also return modal parameters. They require a frequency-rich input/output history, which, in the simulation/identification programs, is obtained using PRBS sequences as inputs. The same program generating PRBS sequences can be used as an excitation signal in identification data acquisition from real processes. All programs developed in this research are described and listed in Appendix A. System/Observer/Controller Identification Toolbox [3], developed at the NASA Langley Research Center, is required to run these programs.

A step-by-step practical guide to system identification with a special emphasis on identification of modal parameters is presented in Appendix B. Two identification algorithms are used with the same numerical example. The results obtained and reported in detail in Appendix B confirm that both algorithms give reasonably good results.

REFERENCES

[1] James III,G.H.; Carne T.G.; and Lauffer J.P. 1993. "The Natural Excitation Technique (NExt) for Modal Parameter Extraction from Operating Wind Turbines," Report No. SAND92-1666, Sandia National Laboratories. February.

[2] Juang, J.-N. 1994. *Applied System Identification*, Englewood Cliffs, New Jersey,Prentice Hall.

[3] Juang, J.-N.; Horta, L.G.; and Phan M. 1992. "System/Observer/Controller Identification Toolbox", *NASA Technical Memorandum 107566*, NASA LaRC. February.

[4] Juang, J.-N.; Phan, M.; Horta, L.G.; and Longman, L.G. 1993. "Identification of Observer and Kalman Filter Markov Parameters: Theory and Experiments," *Journal of Guidance, Control and Dynamics*, vol.16, No. 2, pp.320-329.

[5] Phan, M.; Horta, L.G.; Juang, J.-N.; and Longman, R.W. 1992."Linear System Identification Via an Asymptotically Stable Observer," *NASA Technical Paper 3164*, June.

[6] Newland D.E. 1984. *An Introduction to Random Vibrations and Spectral Analysis*, Longman.

[7] Landau, I.D. 1990. *System Identification and Control Design*, Englewood Cliffs, New Jersey, Prentice Hall.

[8] Moonen, M.; Demoor, B.; Vandenberghe, L.; and Vandewalle, J. 1989. "On- and Off-Line Identification of Linear State-Space Models," *Intenational Journal of Control*, Vol. 49, pp. 219-232.

[9] Chen, C.W.; Huang, J.-K.; Phan, M.; and Juang, J.-N. 1992. "Integrated System Identification and Modal State Estimation for Control of Large Flexible Space Structures," *Journal of Guidance, Control and Dynamics*, vol.15, No.1, pp. 88-95.

[10] Lew, J.S.; Juang, J.-N.; and Longman, R.W. 1991. "Comparison of Several System Identification Methods for Flexible Structures," *Proceedings of the 32nd Structures, Structural Dynamics and Materials Conference*, Baltimore, MD, April, pp.2304-2318

APPENDIX A

Simulation/Identification Programs

Program prbsm.m

This program has been developed to generate the frequency-rich input test signal to be used for identification of dynamical systems. It has been discussed in detail in Section 4 of this report. This program is listed below. It is an interactive program. The log of user-Matlab interaction is as follows:

```
» prbsm
N (~=tr/Tprbs=4-10)=6
a(magnitude)=.1
»
```

The user is asked to determine the magnitude a and the number of stages of the shift register N, which determines the length L of the generated sequence. Also, the maximum pulse length in the sequence is equal to N clock (or sampling) periods. The sampling period has to be declared in the application program, which calls for execution of **prbsm.m**. The generated PRBS sequence is available as a row vector PRB. The response to **help prbsm** is as follows:

```
» help prbsm

 PROGRAM prbsm.m
 This program generates pseudo-random binary sequence using
 shift register with automatic modification of the shift
 register positions added mod2.
 If Tprbs is the generator clock period and tr is the rise
 time, then one should have N*Tprbs>tr. Roughly one can
 say that N indicates number of clock periods over the rise
 time
 Number of cells of the shift register SR is equal to N.
 Generated PRBS is stored in the Output Register PRB.
 It has a binary form, i.e., it assumes either the value of +a or -a.
 If Ts is the sample period, then Tprbs=p*Ts with p being an integer,
 but it is recommended to choose p<=4.

»
```

This information advises the user how the choice of N and the generator clock period should be related to the dynamics of the system to be identified.

```
%PROGRAM prbsm.m
%This program generates pseudo-random binary sequence using
%shift register with automatic modification of the shift
%register positions added mod2.
%If Tprbs is the generator clock period and tr is the rise
%time, then one should have N*Tprbs>tr. Roughly one can
%say that N indicates number of clock periods over the rise
%time
%Number of cells of the shift register SR is equal to N.
```

```
%Generated PRBS is stored in the Output Register PRB.
%It has a binary form, i.e., it assumes either the value of +a or -a.
%If Ts is the sample period, then Tprbs=p*Ts with p being an integer,
%but it is recommended to choose p<=4.
clf
N=input('N (~=tr/Tprbs=4-10)=');%User-defined parameter
a=input('a(magnitude)=');%User-defined parameter
L=2^N-1;%length of the sequence stored in the output register PRB
PRB=zeros(1,L); %initialization of the Output Register
SR=ones(1,N);     %initialization of the Shift Register
SR1=zeros(1,N); %mod 2 sum of the first and last digit of SR is
            %stored in SR1(1)
M=[3 4;3 5;5 6;4 7;3 8;5 9;7 10];  %bits added in PRBS generation
li=M(N-3,1); lh=M(N-3,2);  %determination of the positions
                    %of SR added modulo 2
for m=1:L
      SRSUM=SR(li)+SR(lh);
%MODULO 2 SUMMATION
            if SRSUM==0
               SR1(1)=0;
            elseif SRSUM==2
               SR1(1)=0;
            else
               SR1(1)=1;
            end
%OUTPUT GENERATION
            if SR(N)==1
               PRB(m)=a;
            else
               PRB(m)=-a;
            end
%SHIFTING SR TO THE RIGHT AND UPDATING
            for l=1:N-1
                SR(N+1-l)=SR(N-l);
            end
      SR(1)=0;
      SR=SR+SR1;
end
%
k=0:L-1;
[kk,PP]=stairs(k,PRB);
v=[0,L,-1.2*a,1.2*a];
plot(kk,PP),...
axis(v),...
title('PRBS Test Signal'),...
xlabel('Time, samples'),...
ylabel('Magnitude');
```

□

Identification Programs

There are two identification programs for single input multiple output systems developed in this research:

- **fiokuy.m** in which observer/Kalman filter algorithm is used;
- **fisvuy.m** in which state vector realization algorithm is used.

System/Observer/Controller Identification Toolbox [3], developed at NASA Langley Research Center, is required to run these programs. Both programs identify state space representation of a system and return modal parameters based on frequency-rich input/output history. Because the code of these programs is made a part of simulation/identification programs, presented in the next subsection, the test results presented in Appendix B can be considered as an evaluation of their performance.

The information on user-defined parameters and variables available to the user is included in the Matlab responses to **help fiokuy** and **help fisvuy**, as shown below. The listings of both programs immediately follow.

```
» help fiokuy

 PROGRAM fiokuy.m
 This is the IDENTIFIER program
 for single input multiple output system.
 When calling this program the user has to load
      (in the MATLAB format) frequency-rich input-output time histories
 OBSERVER/KALMAN FILTER IDENTIFICATION ALGORITHM is used.

 User-Defined Parameters:

      T - sampling interval used to obtain
          input-output time histories
      u - input sequence (single column vector)
      y - output sequence (a matrix with m columns)
      m - number of outputs
      p - number of observer parameters
            Note: Maximum order of the system which
                  can be identified is p*m.

 Variables Available:

      A1,B1,C1,D1 identified state space representation
      G - observer gain matrix
      y - response to u of the  system to be identified
      y1 - response to u of the identified system
      y2=[y y1]
      error - identification error.
```

```
» help fisvuy

 PROGRAM fisvuy.m
 This is the IDENTIFIER program
 for single input multiple output system.
 When calling this program the user has to load
      (in the MATLAB format) frequency-rich input-output time histories
 STATE VECTOR REALIZATION ALGORITHM is used for identification.

  User-Defined Parameters:

      n - chosen model order
      T - sampling interval used to obtain
          input-output time histories
      u - input sequence (single column vector)
      y - output sequence (a matrix with m columns)
      nr - length of the sample window or
           number of columns of the Hankel matrix
      nm - number of sample shifts or number of the
           input-output pairs of raws of the Hankel matrix
           (MUST BE EVEN!)
      ks - realization starts from the ks-th sampling instant

 Variables Available:

      A1,B1,C1,D1 identified state space representation
      y - response to u of the  system to be identified
      y1 - response to u of the identified system
      y2=[y y1]
      error - identification error
      eg - MODAL PARAMETERS MATRIX with damping ratios (%)
           in the first column, frequencies (Hz) in the
           second column, and the eigenvalues of the
           corresponding CONTINUOUS-TIME MODEL in the third
           column
      sg - vector of the singular values of the Hankel matrix
           - can be used to choose the model order.


»
```

```
%PROGRAM fiokuy.m
%This is the IDENTIFIER program
%for single input multiple output system.
%When calling this program the user has to load
%     (in the MATLAB format) frequency-rich input-output time histories
%OBSERVER/KALMAN FILTER IDENTIFICATION ALGORITHM is used.
%
%User-Defined Parameters:
%
%     T - sampling interval used to obtain
%         input-output time histories
%     u - input sequence (single column vector)
%     y - output sequence (a matrix with m columns)
%     m - number of outputs
%     p - number of observer parameters
%           Note: Maximum order of the system which
%                 can be identified is p*m.
%
%Variables Available:
%
%     A1,B1,C1,D1 identified state space representation
%     G - observer gain matrix
%     y - response to u of the  system to be identified
%     y1 - response to u of the identified system
%     y2=[y y1]
%     error - identification error.
pt=length(u); %number of points in the sample
t=[T:T:T*pt]';
m=input('m(number of outputs)=');%number of system outputs
%r=input('r(number of inputs)=');%number of system inputs
r=1;
disp('Declare p - number of observer Markov parameters.')
disp('Maximum order of the system to be identified is p*m')
p=input('p(number of observer Markov parameters)=');
      %number of observer Markov parameters
      %system of maximum order p*m can be identified
[A1,B1,C1,D1,G]=okid(m,r,T,u,y,'batch',p);%identification
y1=dlsim(A1,B1,C1,D1,u);%simulation to determine
                  %identification error
y2=[y y1];
clf
plot(y2),title('System and identified model response')
disp('Both real system output and identified system output')
disp('as response to the same input are now displayed in the Figure
Window.')
disp('To save the plot you can open New Figure now.')
disp('ENTER to continue.')
pause
error=y-y1;%identification error
clf
plot(error),title('Identification Error')
disp('Identification error is now displayed in the Figure Window')
disp('Identified system matrices are available as A1, B1, C1, D1')
disp('Observer gain matrix is available as G')
□
```

```
%PROGRAM fisvuy.m
%This is the IDENTIFIER program
%for single input multiple output system.
%When calling this program the user has to load
%     (in the MATLAB format) frequency-rich input-output time histories
%STATE VECTOR REALIZATION ALGORITHM is used for identification.
%
% User-Defined Parameters:
%
%     n - chosen model order
%     T - sampling interval used to obtain
%         input-output time histories
%     u - input sequence (single column vector)
%     y - output sequence (a matrix with m columns)
%     nr - length of the sample window or
%          number of columns of the Hankel matrix
%     nm - number of sample shifts or number of the
%          input-output pairs of raws of the Hankel matrix
%          (MUST BE EVEN!)
%     ks - realization starts from the ks-th sampling instant
%
%Variables Available:
%
%     A1,B1,C1,D1 identified state space representation
%     y - response to u of the  system to be identified
%     y1 - response to u of the identified system
%     y2=[y y1]
%     error - identification error
%     eg - MODAL PARAMETERS MATRIX with damping ratios (%)
%          in the first column, frequencies (Hz) in the
%          second column, and the eigenvalues of the
%          corresponding CONTINUOUS-TIME MODEL in the third
%          column
%     sg - vector of the singular values of the Hankel matrix
%          - can be used to choose the model order.
%
n=input('n(chosen model order)='); %user defined parameter
clf
nr=input('nr(window width)=');
nm=input('nm(number of window shifts)=');
ks=input('ks(first data used for realization)=');
L=length(u)
while L<nr+nm+ks-1
   disp('CHANGE PARAMETERS. IT IS REQUIRED THAT L>=nr+nm+ks-1.')
   nr=input('nr(window width)=');
   nm=input('nm(number of window shifts)=');
   ks=input('ks(first data used for realization)=');
end
pt=length(PRB); %number of points in the sample
t=[T:T:T*pt]';
[y,x]=lsim(A,B,C,D,u,t);
[A1,B1,C1,D1,sg,eg]=svra(u,y,n,nm,nr,T,ks);
y1=dlsim(A1,B1,C1,D1,u);
y2=[y y1];
clf
plot(y2),title('System and identified model response')
disp('Both real system output and identified system output')
disp('as response to the same input are now displayed in the Figure
Window.')
disp('To save the plot you can open New Figure now.')
```

```
disp('ENTER to continue.')
pause
error=y-y1;%identification error
clf
plot(error),title('Error')
disp('Identification error is now displayed in the Figure Window')
disp('Identified system matrices are available as A1, B1, C1, D1')
disp('Vector of the Hankel matrix singular values is')
disp('     available as sg')
□
```

Simulation/Identification Programs

All simulation/identification programs developed in this research use the PRBS signal as the input excitation of the simulated system. The following programs use the observer/Kalman filter identification algorithm:

- **fiokprbs.m** which when executed, requests the user to declare ABCD matrices of the system to be simulated;
- **miokprbs.m** which requires the user to define ABCD matrices before calling it.

The programs using the state vector realization algorithm for identification are the following:

- **fisvprbs.m** which when executed, requests the user to declare ABCD matrices of the system to be simulated;
- **misvprbs.m** which requires the user to define ABCD matrices before calling it.

Using Matlab **help** facility, the user can obtain the relevant information. The Matlab responses to **help** for all four programs are given below. The listings of all programs immediately follow.

```
» help fiokprbs

 PROGRAM fiokprbs.m
 This is the SIMULATOR/IDENTIFIER program
 for single input multiple output system.
 The system is first simulated to obtain input-output sequences.
 PRBS input is used for both simulation and identification.
 OBSERVER/KALMAN FILTER IDENTIFICATION ALGORITHM is used.
 Identified system is simulated using the same PRBS.
 Identification error is determined.
```

```
User-Defined Parameters:

      A,B,C,D state space representation
      tR - approximate value of the rise time
      N - maximum length of PRBS pulse
          (in switching periods Tprbs)
            - should be an integer: 4 to 10
      a - PRBS magnitude
      m - number of outputs
      p - number of observer parameters
            Note: Maximum order of the system which
                  can be identified is p*m.

 Variables Available:

      A1,B1,C1,D1 identified state space representation
      G - observer gain matrix
      PRB - PRBS input applied
      y - response of the simulated system to PRBS
      y1 - response of the identified system to PRBS
      y2=[y y1]
      error - identification error.


» help miokprbs

 PROGRAM miokprbs.m
 When calling this program the user has to define
      A, B, C, D matrices of the system to be simulated
 This is the SIMULATOR/IDENTIFIER program
 for single input multiple output system.
 The system is first simulated to obtain input-output sequences.
 PRBS input is used for both simulation and identification.
 OBSERVER/KALMAN FILTER IDENTIFICATION ALGORITHM is used.
 Identified system is simulated using the same PRBS.
 Identification error is determined.

 User-Defined Parameters:

      tR - approximate value of the rise time
      N - maximum length of PRBS pulse
          (in switching periods Tprbs)
            - should be an integer: 4 to 10
      a - PRBS magnitude
      m - number of outputs
      p - number of observer parameters
            Note: Maximum order of the system which
                  can be identified is p*m.

 Variables Available:

      A1,B1,C1,D1 identified state space representation
      G - observer gain matrix
      PRB - PRBS input applied
      y - response of the simulated system to PRBS
      y1 - response of the identified system to PRBS
      y2=[y y1]
      error - identification error.
```

```
» help fisvprbs

 PROGRAM fisvprbs.m
 This is the SIMULATOR/IDENTIFIER program
 for single input multiple output system
 The system is first simulated to obtain input-output sequences.
 PRBS input is used for both simulation and identification.
 STATE VECTOR REALIZATION ALGORITHM is used for identification.
 Identified system is simulated using the same PRBS.
 Identification error is determined.

  User-Defined Parameters:

      n - chosen model order
      A,B,C,D state space representation
      tR - approximate value of the rise time
      nr - length of the sample window or
           number of columns of the Hankel matrix
      nm - number of sample shifts or number of the
           input-output pairs of raws of the Hankel matrix
           (MUST BE EVEN!)
      ks - realization starts from the ks-th sampling instant
      N  - maximum length of PRBS pulse
           (in switching periods Tprbs)
            - should be an integer: 4 to 10
      a  - PRBS magnitude

 Variables Available:

      A1,B1,C1,D1 identified state space representation
      PRB - PRBS input applied
      y - response of the simulated system to PRBS
      y1 - response of the identified system to PRBS
      y2=[y y1]
      error - identification error
      eg - MODAL PARAMETERS MATRIX with damping ratios (%)
           in the first column, frequencies (Hz) in the
           second column, and the eigenvalues of the
           corresponding CONTINUOUS-TIME MODEL in the third
           column
      sg - vector of the singular values of the Hankel matrix
           - can be used to choose the model order.
```

```
» help misvprbs

 PROGRAM misvprbs.m
 This is the SIMULATOR/IDENTIFIER program
 for single input multiple output system. STATE VECTOR REALIZATION
ALGORITHM is used for  identification.
 Identified system is simulated using the same PRBS.
 When calling this program the user has to define
      A, B, C, D matrices of the system to be simulated.
 The system is first simulated to obtain input-output sequences.
 PRBS input is used for both simulation and identification.

 Identification error is determined.

  User-Defined Parameters:

      n - chosen model order
      tR - approximate value of the rise time
      nr - length of the sample window or
           number of columns of the Hankel matrix
      nm - number of sample shifts or number of the
           input-output pairs of raws of the Hankel matrix
           (MUST BE EVEN!)
      ks - realization starts from the ks-th sampling instant
      N  - maximum length of PRBS pulse
           (in switching periods Tprbs)
            - should be an integer: 4 to 10
      a  - PRBS magnitude

 Variables Available:

      A1,B1,C1,D1 identified state space representation
      PRB - PRBS input applied
      y - response of the simulated system to PRBS
      y1 - response of the identified system to PRBS
      y2=[y y1]
      error - identification error
      eg - MODAL PARAMETERS MATRIX with damping ratios (%)
           in the first column, frequencies (Hz) in the
           second column, and the eigenvalues of the
           corresponding CONTINUOUS-TIME MODEL in the third
           column
      sg - vector of the singular values of the Hankel matrix
           - can be used to choose the model order.


»
```

```
%PROGRAM fiokprbs.m
%This is the SIMULATOR/IDENTIFIER program
%for single input multiple output system.
%The system is first simulated to obtain input-output sequences.
%PRBS input is used for both simulation and identification.
%OBSERVER/KALMAN FILTER IDENTIFICATION ALGORITHM is used.
%Identified system is simulated using the same PRBS.
%Identification error is determined.
%
%User-Defined Parameters:
%
%     A,B,C,D state space representation
%     tR - approximate value of the rise time
%     N - maximum length of PRBS pulse
%         (in switching periods Tprbs)
%           - should be an integer: 4 to 10
%     a - PRBS magnitude
%     m - number of outputs
%     p - number of observer parameters
%           Note: Maximum order of the system which
%                 can be identified is p*m.
%
%Variables Available:
%
%     A1,B1,C1,D1 identified state space representation
%     G - observer gain matrix
%     PRB - PRBS input applied
%     y - response of the simulated system to PRBS
%     y1 - response of the identified system to PRBS
%     y2=[y y1]
%     error - identification error.
%
A=input('A(state matrix)='); %user defined parameter
B=input('B(input matrix)='); %user defined parameter
C=input('C(output matrix)='); %user defined parameter
D=input('D(direct coupling matrix)='); %user defined parameter
tR=input('tR(rise time)=');%used (together with N)
      %to determine the switching period Tprbs.
      %N is the maximum length of PRBS pulse.
prbsm %generation of input sequence
disp('PRBS input is now displayed in the Figure Window.')
disp('To save the plot you can open New Figure now')
disp('ENTER to continue.')
pause
pt=length(PRB); %number of points in the sample
Tprbs=tR/N;
u=PRB';
t=[Tprbs:Tprbs:Tprbs*pt]';
[y,x]=lsim(A,B,C,D,u,t);%system simulation:to generate
                  %output time history
m=input('m(number of outputs)=');%number of system outputs
%r=input('r(number of inputs)=');%number of system inputs
r=1;
disp('Declare p - number of observer Markov parameters.')
disp('Maximum order of the system to be identified is p*m')
disp('ENTER to continue')
p=input('p(number of observer Markov parameters)=');
      %number of observer Markov parameters
      %system of maximum order p*m can be identified
[A1,B1,C1,D1,G]=okid(m,r,Tprbs,u,y,'batch',p);%identification
```

```
y1=dlsim(A1,B1,C1,D1,u);%simulation to determine
                    %identification error
y2=[y y1];
clf
plot(y2),title('PRBS input response')
disp('Both simulated system output and identified system output')
disp('with PRBS as input are now displayed in the Figure Window.')
disp('To save the plot you can open New Figure now.')
disp('ENTER to continue.')
pause
error=y-y1;%identification error
clf
plot(error),title('Identification Error')
disp('Identification error is now displayed in the Figure Window')
disp('Identified system matrices are available as A1, B1, C1, D1')
disp('Observer gain matrix is available as G')
disp('Use "help okid" for more information.')
□
```

```
%PROGRAM miokprbs.m
%When calling this program the user has to define
%     A, B, C, D matrices of the system to be simulated
%This is the SIMULATOR/IDENTIFIER program
%for single input multiple output system.
%The system is first simulated to obtain input-output sequences.
%PRBS input is used for both simulation and identification.
%OBSERVER/KALMAN FILTER IDENTIFICATION ALGORITHM is used.
%Identified system is simulated using the same PRBS.
%Identification error is determined.
%
%User-Defined Parameters:
%
%     tR - approximate value of the rise time
%     N - maximum length of PRBS pulse
%         (in switching periods Tprbs)
%           - should be an integer: 4 to 10
%     a - PRBS magnitude
%     m - number of outputs
%     p - number of observer parameters
%           Note: Maximum order of the system which
%                 can be identified is p*m.
%
%Variables Available:
%
%     A1,B1,C1,D1 identified state space representation
%     G - observer gain matrix
%     PRB - PRBS input applied
%     y - response of the simulated system to PRBS
%     y1 - response of the identified system to PRBS
%     y2=[y y1]
%     error - identification error.
%
tR=input('tR(rise time)=');%used (together with N)
      %to determine the switching period Tprbs.
      %N is the maximum length of PRBS pulse.
prbsm %generation of input sequence
disp('PRBS input is now displayed in the Figure Window.')
disp('To save the plot you can open New Figure now')
disp('ENTER to continue.')
pause
pt=length(PRB); %number of points in the sample
Tprbs=tR/N;
u=PRB';
t=[Tprbs:Tprbs:Tprbs*pt]';
[y,x]=lsim(A,B,C,D,u,t);%system simulation:to generate
                  %output time history
m=input('m(number of outputs)=');%number of system outputs
%r=input('r(number of inputs)=');%number of system inputs
r=1;
disp('Declare p - number of observer Markov parameters.')
disp('Maximum order of the system to be identified is p*m')
disp('ENTER to continue')
p=input('p(number of observer Markov parameters)=');
```

```
      %number of observer Markov parameters
      %system of maximum order p*m can be identified
[A1,B1,C1,D1,G]=okid(m,r,Tprbs,u,y,'batch',p);%identification
y1=dlsim(A1,B1,C1,D1,u);%simulation to determine
                  %identification error
y2=[y y1];
clf
plot(y2),title('PRBS input response')
disp('Both simulated system output and identified system output')
disp('with PRBS as input are now displayed in the Figure Window.')
disp('To save the plot you can open New Figure now.')
disp('ENTER to continue.')
pause
error=y-y1;%identification error
clf
plot(error),title('Identification Error')
disp('Identification error is now displayed in the Figure Window')
disp('Identified system matrices are available as A1, B1, C1, D1')
disp('Observer gain matrix is available as G')
□
```

```
%PROGRAM fisvprbs.m
%This is the SIMULATOR/IDENTIFIER program
%for single input multiple output system
%The system is first simulated to obtain input-output sequences.
%PRBS input is used for both simulation and identification.
%STATE VECTOR REALIZATION ALGORITHM is used for identification.
%Identified system is simulated using the same PRBS.
%Identification error is determined.
%
% User-Defined Parameters:
%
%     n - chosen model order
%     A,B,C,D state space representation
%     tR - approximate value of the rise time
%     nr - length of the sample window or
%          number of columns of the Hankel matrix
%     nm - number of sample shifts or number of the
%          input-output pairs of raws of the Hankel matrix
%          (MUST BE EVEN!)
%     ks - realization starts from the ks-th sampling instant
%     N  - maximum length of PRBS pulse
%          (in switching periods Tprbs)
%           - should be an integer: 4 to 10
%     a  - PRBS magnitude
%
%Variables Available:
%
%     A1,B1,C1,D1 identified state space representation
%     PRB - PRBS input applied
%     y - response of the simulated system to PRBS
%     y1 - response of the identified system to PRBS
%     y2=[y y1]
%     error - identification error
%     eg - MODAL PARAMETERS MATRIX with damping ratios (%)
%          in the first column, frequencies (Hz) in the
%          second column, and the eigenvalues of the
%          corresponding CONTINUOUS-TIME MODEL in the third
%          column
%     sg - vector of the singular values of the Hankel matrix
%          - can be used to choose the model order.
%
n=input('n(chosen model order)='); %user defined parameter
A=input('A(state matrix)='); %user defined parameter
B=input('B(input matrix)='); %user defined parameter
C=input('C(output matrix)='); %user defined parameter
D=input('D(direct coupling matrix)='); %user defined parameter
tR=input('tR(rise time)=');
clf
nr=input('nr(window width)=');
nm=input('nm(number of window shifts)=');
ks=input('ks(first data used for realization)=');
prbsm %generation of input sequence
while L<nr+nm+ks-1
   disp('CHANGE PARAMETERS. IT IS REQUIRED THAT L>=nr+nm+ks-1.')
   nr=input('nr(window width)=');
   nm=input('nm(number of window shifts)=');
   ks=input('ks(first data used for realization)=');
   prbsm %generation of input sequence
end
disp('PRBS input is now displayed in the Figure Window.')
```

```
disp('To save the plot you can open New Figure now')
disp('ENTER to continue.')
pause
pt=length(PRB); %number of points in the sample
Tprbs=tR/N;
u=PRB';
t=[Tprbs:Tprbs:Tprbs*pt]';
[y,x]=lsim(A,B,C,D,u,t);
[A1,B1,C1,D1,sg,eg]=svra(u,y,n,nm,nr,Tprbs,ks);
y1=dlsim(A1,B1,C1,D1,u);
y2=[y y1];
clf
plot(y2),title('PRBS input response')
disp('Both simulated system output and identified system output')
disp('with PRBS as input are now displayed in the Figure Window.')
disp('To save the plot you can open New Figure now.')
disp('ENTER to continue.')
pause
error=y-y1;
clf
plot(error),title('Error')
disp('Identification error is now displayed in the Figure Window')
disp('Identified system matrices are available as A1, B1, C1, D1')
disp('Matrix containing modal parameters is available as eg')
disp('Vector of the Hankel matrix singular values is')
disp('     available as sg')
disp('Use "HELP SVRA" for more information.')
□
```

```
%PROGRAM misvprbs.m
%This is the SIMULATOR/IDENTIFIER program
%for single input multiple output system.
%When calling this program the user has to define
%     A, B, C, D matrices of the system to be simulated.
%The system is first simulated to obtain input-output sequences.
%PRBS input is used for both simulation and identification.
%STATE VECTOR REALIZATION ALGORITHM is used for identification.
%Identified system is simulated using the same PRBS.
%Identification error is determined.
%
% User-Defined Parameters:
%
%     n - chosen model order
%     tR - approximate value of the rise time
%     nr - length of the sample window or
%          number of columns of the Hankel matrix
%     nm - number of sample shifts or number of the
%          input-output pairs of raws of the Hankel matrix
%          (MUST BE EVEN!)
%     ks - realization starts from the ks-th sampling instant
%     N  - maximum length of PRBS pulse
%          (in switching periods Tprbs)
%           - should be an integer: 4 to 10
%     a  - PRBS magnitude
%
%Variables Available:
%
%     A1,B1,C1,D1 identified state space representation
%     PRB - PRBS input applied
%     y - response of the simulated system to PRBS
%     y1 - response of the identified system to PRBS
%     y2=[y y1]
%     error - identification error
%     eg - MODAL PARAMETERS MATRIX with damping ratios (%)
%          in the first column, frequencies (Hz) in the
%          second column, and the eigenvalues of the
%          corresponding CONTINUOUS-TIME MODEL in the third
%          column
%     sg - vector of the singular values of the Hankel matrix
%          - can be used to choose the model order.
%
n=input('n(chosen model order)='); %user defined parameter
tR=input('tR(rise time)=');
clf
nr=input('nr(window width)=');
nm=input('nm(number of window shifts)=');
ks=input('ks(first data used for realization)=');
prbsm %generation of input sequence
while L<nr+nm+ks-1
   disp('CHANGE PARAMETERS. IT IS REQUIRED THAT L>=nr+nm+ks-1.')
   nr=input('nr(window width)=');
   nm=input('nm(number of window shifts)=');
```

```
ks=input('ks(first data used for realization)=');
   prbsm %generation of input sequence
end
disp('PRBS input is now displayed in the Figure Window.')
disp('To save the plot you can open New Figure now')
disp('ENTER to continue.')
pause
pt=length(PRB); %number of points in the sample
Tprbs=tR/N;
u=PRB';
t=[Tprbs:Tprbs:Tprbs*pt]';
[y,x]=lsim(A,B,C,D,u,t);
[A1,B1,C1,D1,sg,eg]=svra(u,y,n,nm,nr,Tprbs,ks);
y1=dlsim(A1,B1,C1,D1,u);
y2=[y y1];
clf
plot(y2),title('PRBS input response')
disp('Both simulated system output and identified system output')
disp('with PRBS as input are now displayed in the Figure Window.')
disp('To save the plot you can open New Figure now.')
disp('ENTER to continue.')
pause
error=y-y1;
clf
plot(error),title('Error')
disp('Identification error is now displayed in the Figure Window')
disp('Identified system matrices are available as A1, B1, C1, D1')
disp('Vector of the Hankel matrix singular values is')
disp('     available as sg')
□
```

APPENDIX B

Application Notes on Two Identification Algorithms and a Numerical Example

This appendix is a practical step-by-step guide to system identification with a special emphasis on identification of modal parameters. Two identification algorithms are used with the same numerical example:

- The observer/Kalman filter identification algorithm, developed and implemented at NASA Langley as Matlab function **okid** [3];
- The state vector realization algorithm, introduced in [8] and implemented at NASA Langley as a Matlab function **svra** [3].

Both algorithms identify a state-space model of a multi-input and multi-output, linear, time-invariant system from a set of frequency-rich input response data. A PRBS input is used, which is generated by the Matlab program **prbsm.m**, developed in this research. The application simulator/identifier programs **miokprbs.m** (okid identification) and **misvprbs.m** (svra identification), presented in Appendix A, are used in the numerical example. The simulation/identification example is implemented by the program **exstrok.m** (using the okid identification algorithm) and by the program **exstrsv.m** (using the svra identification algorithm). Matlab responses to **help exstrok** and **help exstrsv** are listed below.

```
» help exstrok

 PROGRAM exstrok.m
 This is the SIMULATION/IDETIFICATION example,
 the lumped mass beamlike system with three masses,
 force applied at one of the masses is the single input
 and the responses (displacements) at the other three masses
 are the two outputs. OKID identification algorithm is used.
 The original state space model
 is discrete-time, obtained at the sampling frequency of 10 Hz.
 The continuous model (A,B,C,D) is obtained.
 Then SIMULATION/IDENTIFICATION is performed by miokprbs.m.

» help exstrsv

 PROGRAM exstrsv.m
 This is the SIMULATION/IDETIFICATION example,
 the lumped mass beamlike system with tree masses,
 force applied at one of the masses is the single input
 and the responses (displacements) at the other three masses
 are the two outputs. SVRA identification algorithm is used.
 The original state space model
 is discrete-time, obtained at the sampling frequency of 10 Hz.
 The continuous model (A,B,C,D) is obtained.
 Then SIMULATION/IDENTIFICATION is performed by misvprbs.m.
```

Numerical Example

This is the example used in [4] and [9] to study the numerical properties of the integrated system identification and state estimation algorithm. In this example, the lumped-mass, beam-like simulated system with three masses has three modes (six states). The system is excited by PRBS force u at one of the three nodes (masses), and the responses are measured at two other nodes. This system is described in the state space by the following matrices:

```
A =
 -1.0455e-002  1.6370e+000          0          0          0          0
 -1.6370e+000 -1.0455e-002          0          0          0          0
            0           0 -4.5123e-002  4.4720e+000          0          0
            0           0 -4.4720e+000 -4.5123e-002          0          0
            0           0          0          0 -7.9416e-002  6.1082e+000
            0           0          0          0 -6.1082e+000 -7.9416e-002
B =
  9.4195e-006
  1.3467e-001
  3.5517e-004
 -7.4542e-002
  2.8787e-004
  3.6440e-002
C =
  1.5112e+000           0  2.0000e+000           0  1.5119e+000           0
  1.3093e+000           0           0           0 -1.3093e+000           0
D =
     0
     0
```

and has the following modal parameters:

Mode No.	Frequency (Hz)	Damping (%)
1	0.261	0.63
2	0.712	1.01
3	0.972	1.30

Assuming a rise time of tR=1 s and N=10, the sampling frequency (or clock frequency of the PRBS generator) is 10 Hz, which is sufficient for identification of the highest modal frequency (0.97 Hz) of the system. This results in 1023 data points in the simulated u and y sequences. However, because the system considered has two outputs, each data point of y consists of two values.

***Interactive execution of* exstrok.m**

The program **exstrok.m** is listed below. The log of the user-Matlab interaction, while this program is being executed, immediately follows. The user is asked to declare tR, N, and a (the magnitude of PRBS input). PRBS input is generated and displayed in the Figure

Window. Then, the user is asked to declare p, which is the number of observer Markov parameters. What happens when p is too small is illustrated in Table B-1 and Fig. B-1., both showing the identification results for p=4. In reality, we do not know the true frequency response. But the negative damping in Table B-1 (generated by the identification program), indicates that something is wrong and, therefore, the value of p must be increased. For p=5 the results are shown in Table B-2 and Fig. B-2. We can see that, in the bandwidth of interest, the identified frequency response is right on top of the true frequency response. This picture does not change for higher p values and, consequently, is not repeated for p=10, 15, and 20 for which modal parameters, identified by exstrok.m are shown in Tables B-3, B-4, and B-5, respectively. In analyzing those tables, we first notice that as p increases, the number of rows in the Modal Parameter Table increases. Increasing p, we increase the maximum order of the system to be identified, which equals $p*m$. The Matlab program analyzing singular values of the Hankel matrix reduces this order. As a result, in the interactive execution listing for p=8 (shown below), the user gets the following statement:

```
The initial order is set to 16.
It is now set to 7.
```

Then, the Modal Parameter Table follows. Note that in the third column of this table shows mode singular values and the fourth column specifies Modal Amplitude Coherence (MAC). Comparing Modal Parameter Tables for p greater than 5 we see that, in the last six rows, modal frequencies hardly change, there are some changes in damping coefficients, and, most importantly, we have simultaneously MAC=1 and high mode singular values (much higher than in the other rows). Therefore, we can assume that the modes of the system identified are adequately represented by the last six rows. Comparing the identified modal parameters with the true values shoes that the results obtained are reasonably good. Also, the eigenvalues of the original and identified system, printed out for comparison at the end of the interactive execution log (for p=8), coincide reasonably well.

```
%PROGRAM exstrok.m
%This is the SIMULATION/IDETIFICATION example,
%the lumped mass beamlike system with tree masses,
%force applied at one of the masses is the single input
%and the responses (displacements) at the other three masses
%are the two outputs. OKID identification algorithm is used.
%The original state space model
%is discrete-time, obtained at the sampling frequency of 10 Hz.
%The continuous model (A,B,C,D) is obtained.
%Then SIMULATION/IDENTIFICATION is performed by miokprbs.m.
s1=[.9856 .1628 0 0 0 0;-.1628 .9856 0 0 0 0];
s2=[0 0 .8976 .4305 0 0;0 0 -.4305 .8976 0 0];
s3=[0 0 0 0 .8127 .5690;0 0 0 0 -.5690 .8127];
AA=[s1;s2;s3];
BB=[.0011 .0134 -.0016 -.0072 .0011 .0034]';
s4=[1.51119 0 2 0 1.5119 0];
s5=[1.3093 0 0 0 -1.3093 0];
C=[s4;s5];
D=[0 0]';
[A,B]=d2c(AA,BB,.1);%original continuous system is obtained.
disp('Set tR=1, and N=10, which gives the sampling period')
disp('Tprbs=0.1. Set m=2 (number of outputs).')
disp('Choose p (the number of observer Markov')
disp('parameters) so that p*m is much larger than')
disp('the effective order of the system for identification.')
disp('Set, say,  p=10.')
miokprbs
disp('To save the plot you can open New Figure now.')
disp('ENTER to continue.')
pause
[a1,b1]=d2c(A1,B1,Tprbs);%identified continuous system is obtained.
[num1,den1]=ss2tf(a1,b1,C1,D1,1);%identified trnsfer function
[num,den]=ss2tf(A,B,C,D,1);%original transfer function
disp('ORIGINAL SYSTEM EIGENVALUES')
r=roots(den)%original eigenvalues
disp('IDENTIFIED SYSTEM EIGENVALUES')
r1=roots(den1)%identified eigenvalues
[mag,phase,w]=dbode(AA,BB,C,D,.1,1);
[mag1,phase1,w]=dbode(A1,B1,C1,D1,Tprbs,1,w);
lmag=20*log(mag(:,1));
v=[.1,10,-100,50];
lmag1=20*log(mag1(:,1));
subplot(2,1,1)
semilogx(w,lmag,'r',w,lmag1,'*g'),grid,...
axis(v),...
xlabel('Frequency (rad/sec)'),...
ylabel('Gain dB'),...
title('FREQUENCY RESPONSE FOR OUTPUT 1'),...
text(.11,40,'____ original system'),...
text(.11,10,'****** identified system')
subplot(2,1,2)
semilogx(w,phase(:,1),'r',w,phase1(:,1),'*g'),grid,...
  v1=[.1,10,-600,0];,...
  axis(v1),...
```

```
  xlabel('Frequency (rad/sec)'),...
  ylabel('Phase deg')
disp('Both original system and identified system')
disp('frequency responses for OUTPUT 1 are now displayed in the Figure
Window.')
disp('To save the plot you can open New Figure now.')
disp('ENTER to continue.')
pause
lmag=20*log(mag(:,2));
%v=[.1,10,-100,50];
lmag1=20*log(mag1(:,2));
subplot(2,1,1)
semilogx(w,lmag,'r',w,lmag1,'*g'),grid,...
axis(v),...
xlabel('Frequency (rad/sec)'),...
ylabel('Gain dB'),...
title('FREQUENCY RESPONSE FOR OUTPUT 2'),...
text(.11,40,'____ original system'),...
text(.11,10,'****** identified system')
subplot(2,1,2)
semilogx(w,phase(:,2),'r',w,phase1(:,2),'*g'),grid,...
  v1=[.1,10,-400,0];,...
  axis(v1),...
  xlabel('Frequency (rad/sec)'),...
  ylabel('Phase deg')
disp('Both original system and identified system')
disp('frequency responses for OUTPUT 2 are now displayed in the Figure
Window.')
```

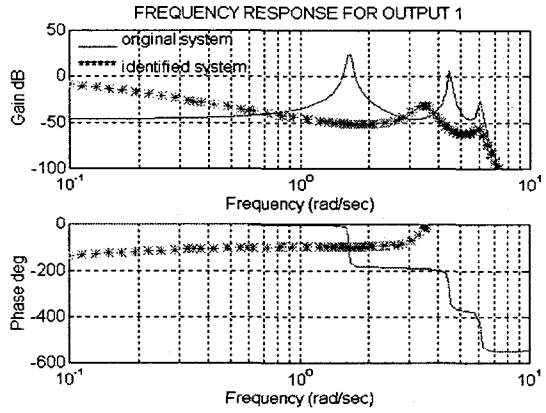
Interactive execution of exstrok.m for p=8

```
» exstrok
Set tR=1, and N=10, which gives the sampling period
Tprbs=0.1. Set m=2 (number of outputs).
Choose p (the number of observer Markov
parameters) so that p*m is much larger than
the effective order of the system for identification.
Set, say,  p=10.

tR(rise time)=1

N (~=tr/Tprbs=4-10)=10

a(magnitude)=.5
PRBS input is now displayed in the Figure Window.
To save the plot you can open New Figure now
ENTER to continue.

m(number of outputs)=2
Declare p - number of observer Markov parameters.
Maximum order of the system to be identified is p*m
ENTER to continue

p(number of observer Markov parameters)=8
batch
 Total number of sample  points = 1023
 Number of experiments in file = 1
 Number of inputs = 1
 Number of outputs = 2
 Compute Parameters For Data Set Number 1
 Time (min) to compute observer parameters 0.08317
 Time (min) to compute Markov parameters 0.0045
ERADC is used now.
The Hankel matrix size for ERADC is 16 by 57.
Maximum Hankel singular value = 4.826635e+006
Minimum Hankel singular value = 4.367265e-011
The initial order is set to 16.
It is now set to 7.
   Damping(%)     Freq(HZ)       Mode SV        MAC
  1.7684e+001  5.0801e+000  6.4661e-004  9.3453e-001
  1.2652e+000  9.7271e-001  2.2193e-001  1.0000e+000
  1.2652e+000  9.7271e-001  2.2193e-001  1.0000e+000
  9.3900e-001  7.1139e-001  3.9919e-001  1.0000e+000
  9.3900e-001  7.1139e-001  3.9919e-001  1.0000e+000
  8.8257e-001  2.6055e-001  1.0000e+000  1.0000e+000
  8.8257e-001  2.6055e-001  1.0000e+000  1.0000e+000
Both simulated system output and identified system output
with PRBS as input are now displayed in the Figure Window.
To save the plot you can open New Figure now.
ENTER to continue.
Identification error is now displayed in the Figure Window
Identified system matrices are available as A1, B1, C1, D1
```

```
Observer gain matrix is available as G
To save the plot you can open New Figure now.
ENTER to continue.
Warning: Accuracy of d2c conversion may be poor.
ORIGINAL SYSTEM EIGENVALUES
r =
-7.9416e-002+ 6.1082e+000i
-7.9416e-002- 6.1082e+000i
-4.5123e-002+ 4.4720e+000i
-4.5123e-002- 4.4720e+000i
-1.0455e-002+ 1.6370e+000i
-1.0455e-002- 1.6370e+000i
IDENTIFIED SYSTEM EIGENVALUES
r1 =
-5.6447e+000
-7.7323e-002+ 6.1113e+000i
-7.7323e-002- 6.1113e+000i
-4.1972e-002+ 4.4696e+000i
-4.1972e-002- 4.4696e+000i
-1.4449e-002+ 1.6370e+000i
-1.4449e-002- 1.6370e+000i
Both original system and identified system
frequency responses for OUTPUT 1 are now displayed in the Figure Window.
To save the plot you can open New Figure now.
ENTER to continue.
Both original system and identified system
frequency responses for OUTPUT 2 are now displayed in the Figure Window.
»
```

Table B-1 Modal parameters identified by exstrok.m, p=4

Damping(%)	Freq(HZ)	Mode SV	MAC
5.6540e+001	3.5023e+000	1.4643e-002	9.9949e-001
5.6540e+001	3.5023e+000	1.4643e-002	9.9949e-001
-1.0000e+002	1.6194e-002	1.0000e+000	1.0000e+000
-8.9882e+000	5.5175e-001	3.9447e-001	1.0000e+000
-8.9882e+000	5.5175e-001	3.9447e-001	1.0000e+000
-5.5469e+000	9.6481e-001	2.4311e-001	1.0000e+000
-5.5469e+000	9.6481e-001	2.4311e-001	1.0000e+000



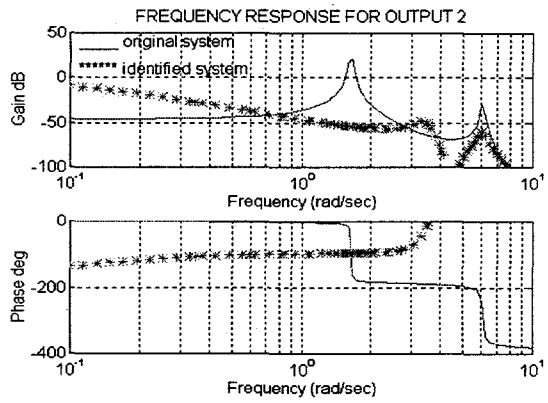


Figure B-1 Frequency responses for p=4.

Table B-2 Modal parameters identified by exstrok.m, p=5

Damping(%)	Freq(HZ)	Mode SV	MAC
1.3120e+000	9.7177e-001	1.7498e-001	1.0000e+000
1.3120e+000	9.7177e-001	1.7498e-001	1.0000e+000
1.3111e+000	7.1176e-001	2.7104e-001	1.0000e+000
1.3111e+000	7.1176e-001	2.7104e-001	1.0000e+000
5.5778e-001	2.6307e-001	1.0000e+000	1.0000e+000
5.5778e-001	2.6307e-001	1.0000e+000	1.0000e+000

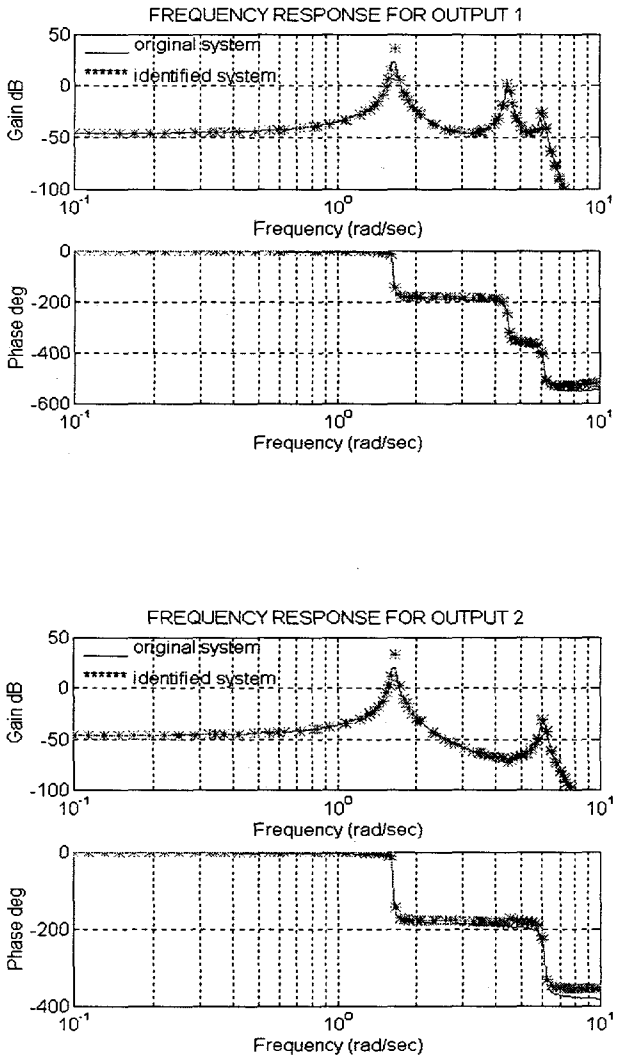


Figure B-2 Frequency responses for p=5.

Table B-3 Modal parameters identified by exstrok.m, p=10

Damping(%)	Freq(HZ)	Mode SV	MAC
1.0000e+002	5.9101e-001	3.4620e-003	9.9980e-001
1.2881e+000	9.7223e-001	1.7662e-001	1.0000e+000
1.2881e+000	9.7223e-001	1.7662e-001	1.0000e+000
9.8865e-001	7.1146e-001	3.1199e-001	1.0000e+000
9.8865e-001	7.1146e-001	3.1199e-001	1.0000e+000
5.6211e-001	2.6099e-001	1.0000e+000	1.0000e+000
5.6211e-001	2.6099e-001	1.0000e+000	1.0000e+000

Table B-4 Modal parameters identified by exstrok.m, p=15

Damping(%)	Freq(HZ)	Mode SV	MAC
9.0604e+001	3.4851e-001	1.0072e-002	1.0000e+000
9.0604e+001	3.4851e-001	1.0072e-002	1.0000e+000
1.5308e+001	1.6035e+000	1.5254e-003	9.9902e-001
1.5308e+001	1.6035e+000	1.5254e-003	9.9902e-001
1.2812e+000	9.7185e-001	1.6441e-001	1.0000e+000
1.2812e+000	9.7185e-001	1.6441e-001	1.0000e+000
1.1260e+000	7.1184e-001	2.7119e-001	1.0000e+000
1.1260e+000	7.1184e-001	2.7119e-001	1.0000e+000
4.7782e-001	2.6133e-001	1.0000e+000	1.0000e+000
4.7782e-001	2.6133e-001	1.0000e+000	1.0000e+000

Table B-5 Modal parameters identified by exstrok.m, p=20

Damping(%)	Freq(HZ)	Mode SV	MAC
3.4098e+001	8.3804e-001	7.4628e-003	9.9999e-001
3.4098e+001	8.3804e-001	7.4628e-003	9.9999e-001
1.0000e+002	1.4218e-001	9.8751e-003	1.0000e+000
9.2890e+000	1.4129e+000	1.7146e-003	9.9851e-001
9.2890e+000	1.4129e+000	1.7146e-003	9.9851e-001
1.3100e+000	9.7236e-001	1.8977e-001	1.0000e+000
1.3100e+000	9.7236e-001	1.8977e-001	1.0000e+000
1.0681e+000	7.1162e-001	3.2672e-001	1.0000e+000
1.0681e+000	7.1162e-001	3.2672e-001	1.0000e+000
6.5891e-001	2.6031e-001	1.0000e+000	1.0000e+000
6.5891e-001	2.6031e-001	1.0000e+000	1.0000e+000

***Interactive execution of* exstrsv.m**

The program **exstrsv.m** is listed below. The log of user-Matlab interaction, while this program is being executed for the chosen model order n=12, immediately follows. The parameter n is the key parameter (as p, as shown above, is the key parameter for the okid

algorithm). For each set of input-output data, the identification will give consistent results for n greater than some value. As for **exstrok.m,** the user will be asked to declare tR, N, and a. Then, the user will be asked to declare "window width" nr, which is the number of columns in the measurement Hankel matrix whose rows consist of input and output data. The "number of window shifts" nm is the next parameter to be declared by the user. It must be even and it specifies the number of sample shifts in constructing the rows of the Hankel matrix. The following general form of this Hankel matrix clarifies the meaning of those parameters:

$$H=\begin{bmatrix} u_k & u_{k+1} & \cdots & u_{k+nr-1} \\ y_k & y_{k+1} & \cdots & y_{k+nr-1} \\ u_{k+1} & u_{k+2} & \cdots & u_{k+nr} \\ y_{k+1} & y_{k+2} & \cdots & y_{k+nr} \\ \vdots & \vdots & \ddots & \vdots \\ u_{k+nm-1} & u_{k+nm} & \cdots & u_{k+nm+nr-2} \\ y_{k+nm-1} & y_{k+nm} & \cdots & y_{k+nm+nr-2} \end{bmatrix}$$

where u_i and y_i denote the r-dimensional input vector and m-dimensional output vector at time i, respectively. Integers $nm*m$ (m is the number of outputs) and nr are required to be not smaller than the chosen model order n. Also, $n+nm*r$ (r is the number of inputs) needs to be smaller than nr. The data used for realization starts with the ks-th discrete time to be declared. PRBS input is generated and displayed in the Figure Window.

As already discussed, the chosen model order n is the key parameter. To correctly identify the system, n must be greater than the real order of the system, represented by the input-output data. The results when n is too small are illustrated by Table B-6 and Figure B-3, which both show the identification results for n=5 (the true order of our system is 6) and are both automatically generated by **exstrsv.m**. In reality, we do not know the true frequency response. Fortunately, for an oversize model the **svra** function is not able to identify the nonexisting modes. Therefore, a series of warnings will be displayed for the user, but the program will run to the end and will correctly identify the existing modes displaying at the end NaN (Not a Number) as a value of parameters of the non-existing modes. This is evident from both the log of the interactive execution of exstrsv.m for n=12 and from the identified parameters for n=7, 10, and 15, as illustrated by Table B-7 and Figure B-4, Table B-8, and Table B-9, respectively. As can be seen, in all these cases modal parameters are identified exactly. Also, the frequency responses identified are right on top of the true ones. The frequency responses for n=7 only are shown. The true and the identified system eigenvalues are identical.

```
%PROGRAM exstrsv.m
%This is the SIMULATION/IDETIFICATION example,
%the lumped mass beamlike system with three masses,
%force applied at one of the masses is the single input
%and the responses (displacements) at the other three masses
%are the two outputs. SVRA identification algorithm is used.
%The original state space model
%is discrete-time, obtained at the sampling frequency of 10 Hz.
%The continuous model (A,B,C,D) is obtained.
%Then SIMULATION/IDENTIFICATION is performed by misvprbs.m.
s1=[.9856 .1628 0 0 0 0;-.1628 .9856 0 0 0 0];
s2=[0 0 .8976 .4305 0 0;0 0 -.4305 .8976 0 0];
s3=[0 0 0 0 .8127 .5690;0 0 0 0 -.5690 .8127];
AA=[s1;s2;s3];
BB=[.0011 .0134 -.0016 -.0072 .0011 .0034]';
s4=[1.51119 0 2 0 1.5119 0];
s5=[1.3093 0 0 0 -1.3093 0];
C=[s4;s5];
D=[0 0]';
[A,B]=d2c(AA,BB,.1);%original continuous system is obtained.
disp('Set tR=1, and N=10, which gives the sampling period Tprbs=0.1')
misvprbs
disp('To save the plot you can open New Figure now.')
disp('ENTER to continue.')
pause
[a1,b1]=d2c(A1,B1,Tprbs);%identified continuous system is obtained.
[num1,den1]=ss2tf(a1,b1,C1,D1,1);%identified trnsfer function
[num,den]=ss2tf(A,B,C,D,1);%original transfer function
disp('ORIGINAL SYSTEM EIGENVALUES')
r=roots(den)%original eigenvalues
disp('IDENTIFIED SYSTEM MODAL PARAMETERS:')
disp('- damping ratios (%) in the first column')
disp('- frequencies (Hz) in the second column')
disp('- eigenvalues of the corresponding continuous-time model')
disp('      in the third column')
eg%identified system modal parameters
[mag,phase,w]=dbode(AA,BB,C,D,.1,1);
[mag1,phase1,w]=dbode(A1,B1,C1,D1,Tprbs,1,w);
lmag=20*log(mag(:,1));
v=[.1,10,-100,50];
lmag1=20*log(mag1(:,1));
subplot(2,1,1)
semilogx(w,lmag,'r',w,lmag1,'*g'),grid,...
axis(v),...
xlabel('Frequency (rad/sec)'),...
ylabel('Gain dB'),...
title('FREQUENCY RESPONSE FOR OUTPUT 1'),...
text(.11,40,'____ original system'),...
text(.11,10,'****** identified system')
subplot(2,1,2)
semilogx(w,phase(:,1),'r',w,phase1(:,1),'*g'),grid,...
  v1=[.1,10,-600,0];,...
  axis(v1),...
```

```
  xlabel('Frequency (rad/sec)'),...
  ylabel('Phase deg')
disp('Both original system and identified system')
disp('frequency responses for OUTPUT 1 are now displayed in the Figure
Window.')
disp('To save the plot you can open New Figure now.')
disp('ENTER to continue.')
pause
lmag=20*log(mag(:,2));
%v=[.1,10,-100,50];
lmag1=20*log(mag1(:,2));
subplot(2,1,1)
semilogx(w,lmag,'r',w,lmag1,'*g'),grid,...
axis(v),...
xlabel('Frequency (rad/sec)'),...
ylabel('Gain dB'),...
title('FREQUENCY RESPONSE FOR OUTPUT 2'),...
text(.11,40,'____ original system'),...
text(.11,10,'****** identified system')
subplot(2,1,2)
semilogx(w,phase(:,2),'r',w,phase1(:,2),'*g'),grid,...
  v1=[.1,10,-400,0];,...
  axis(v1),...
  xlabel('Frequency (rad/sec)'),...
  ylabel('Phase deg')
disp('Both original system and identified system')
disp('frequency responses for OUTPUT 2 are now displayed in the Figure
Window.')
```

Interactive execution of exstrsv.m for n=12

```
» exstrsv
Set tR=1, and N=10, which gives the sampling period Tprbs=0.1

n(chosen model order)=12

tR(rise time)=1

nr(window width)=40

nm(number of window shifts)=20

ks(first data used for realization)=5

N (~=tr/Tprbs=4-10)=10

a(magnitude)=.5
PRBS input is now displayed in the Figure Window.
To save the plot you can open New Figure now
ENTER to continue.

Warning: Rank deficient, rank = 7  tol =   2.8087e-014

Warning: Log of zero

Warning: Divide by zero
Both simulated system output and identified system output
with PRBS as input are now displayed in the Figure Window.
To save the plot you can open New Figure now.
ENTER to continue.
Identification error is now displayed in the Figure Window
Identified system matrices are available as A1, B1, C1, D1
Vector of the Hankel matrix singular values is
     available as sg
To save the plot you can open New Figure now.
ENTER to continue.
Warning: Accuracy of d2c conversion may be poor.
ORIGINAL SYSTEM EIGENVALUES

r =

  -0.0794 + 6.1082i
  -0.0794 - 6.1082i
  -0.0451 + 4.4720i
  -0.0451 - 4.4720i
  -0.0105 + 1.6370i
  -0.0105 - 1.6370i
```

```
IDENTIFIED SYSTEM MODAL PARAMETERS:
- damping ratios (%) in the first column
- frequencies (Hz) in the second column
- eigenvalues of the corresponding continuous-time model
-    in the third column

eg =

      NaN                NaN                NaN       NaN
      NaN                NaN                NaN       NaN
      NaN                NaN                NaN       NaN
      NaN                NaN                NaN       NaN
      NaN                NaN                NaN       NaN
      NaN                NaN                NaN       NaN
   0.6386             0.2605          -0.0105 + 1.6370i
   0.6386             0.2605          -0.0105 - 1.6370i
   1.3000             0.9722          -0.0794 + 6.1082i
   1.3000             0.9722          -0.0794 - 6.1082i
   1.0089             0.7118          -0.0451 + 4.4720i
   1.0089             0.7118          -0.0451 - 4.4720i

Both original system and identified system
frequency responses for OUTPUT 1 are now displayed in the Figure Window.
To save the plot you can open New Figure now.
ENTER to continue.
Both original system and identified system
frequency responses for OUTPUT 2 are now displayed in the Figure Window.
»
```

Table B-6 Modal parameters identified by exstrsv.m, n=5

```
IDENTIFIED SYSTEM MODAL PARAMETERS:
- damping ratios (%) in the first column
- frequencies (Hz) in the second column
- eigenvalues of the corresponding continuous-time model
   in the third column

eg =

  1.0e+002 *

   0.0011            0.0077            -0.0001 + 0.0485i
   0.0011            0.0077            -0.0001 - 0.0485i
   0.0312            0.0025            -0.0005 + 0.0160i
   0.0312            0.0025            -0.0005 - 0.0160i
   1.0000            0.0022            -0.0138
```

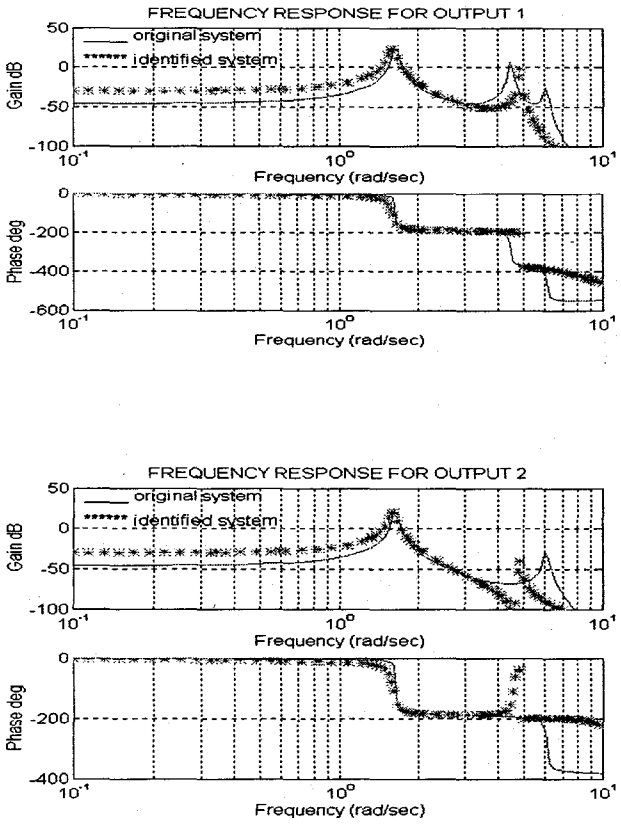
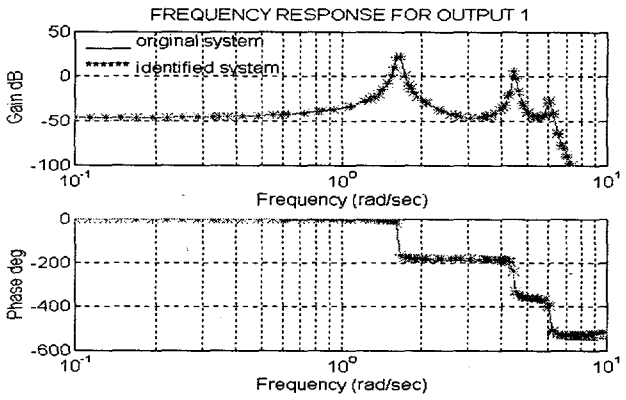


Figure B-3 Frequency response for n=5.

Table B-7 Modal parameters identified by exstrsv.m, n=7

```
IDENTIFIED SYSTEM MODAL PARAMETERS:
- damping ratios (%) in the first column
- frequencies (Hz) in the second column
- eigenvalues of the corresponding continuous-time model
  in the third column

eg =

     NaN          NaN              NaN          NaN
  0.6386       0.2605          -0.0105 + 1.6370i
  0.6386       0.2605          -0.0105 - 1.6370i
  1.3000       0.9722          -0.0794 + 6.1082i
  1.3000       0.9722          -0.0794 - 6.1082i
  1.0089       0.7118          -0.0451 + 4.4720i
  1.0089       0.7118          -0.0451 - 4.4720i
```



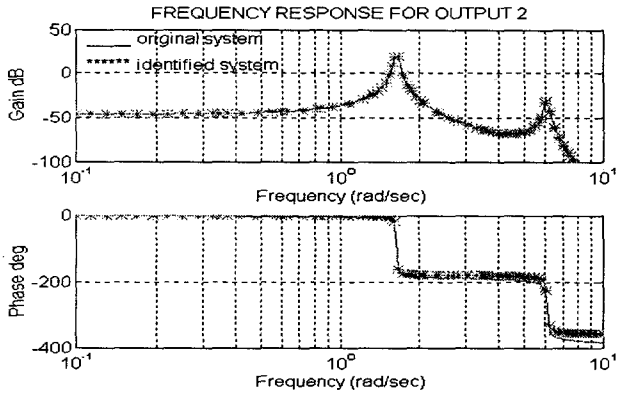


Figure B-4 Frequency response for n=7.

Table B-8 Modal parameters identified by exstrsv.m, n=10

```
IDENTIFIED SYSTEM MODAL PARAMETERS:
- damping ratios (%) in the first column
- frequencies (Hz) in the second column
- eigenvalues of the corresponding continuous-time model
   in the third column
eg =
      NaN                NaN                NaN      NaN
      NaN                NaN                NaN      NaN
      NaN                NaN                NaN      NaN
      NaN                NaN                NaN      NaN
   1.3000             0.9722          -0.0794 + 6.1082i
   1.3000             0.9722          -0.0794 - 6.1082i
   1.0089             0.7118          -0.0451 + 4.4720i
   1.0089             0.7118          -0.0451 - 4.4720i
   0.6386             0.2605          -0.0105 + 1.6370i
   0.6386             0.2605          -0.0105 - 1.6370i
```

Table B-9 Modal parameters identified by exstrsv.m, n=15

```
ORIGINAL SYSTEM EIGENVALUES

r =
  -0.0794 + 6.1082i
  -0.0794 - 6.1082i
  -0.0451 + 4.4720i
  -0.0451 - 4.4720i
  -0.0105 + 1.6370i
  -0.0105 - 1.6370i

IDENTIFIED SYSTEM MODAL PARAMETERS:
- damping ratios (%) in the first column
- frequencies (Hz) in the second column
- eigenvalues of the corresponding continuous-time model
      in the third column
eg =
      NaN                NaN                NaN      NaN
      NaN                NaN                NaN      NaN
      NaN                NaN                NaN      NaN
      NaN                NaN                NaN      NaN
      NaN                NaN                NaN      NaN
      NaN                NaN                NaN      NaN
      NaN                NaN                NaN      NaN
      NaN                NaN                NaN      NaN
      NaN                NaN                NaN      NaN
   1.3000             0.9722          -0.0794 + 6.1082i
   1.3000             0.9722          -0.0794 - 6.1082i
   0.6386             0.2605          -0.0105 + 1.6370i
   0.6386             0.2605          -0.0105 - 1.6370i
   1.0089             0.7118          -0.0451 + 4.4720i
   1.0089             0.7118          -0.0451 - 4.4720i
```

REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1.	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	July 1995	Subcontractor Report

4. TITLE AND SUBTITLE
Advanced System Identification Techniques for Wind Turbine Structures with Special Emphasis on Modal Parameters

5. FUNDING NUMBERS
C:
TA: WE429170

6. AUTHOR(S)
Jan T. Bialasiewicz

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Jan T. Bialasiewicz
University of Colorado, Denver

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Renewable Energy Laboratory
1617 Cole Blvd.
Golden, CO 80401-3393

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
TP-442-7110
DE95009251

11. SUPPLEMENTARY NOTES
NREL technical monitor: Richard M. Osgood

12a. DISTRIBUTION/AVAILABILITY STATEMENT

12b. DISTRIBUTION CODE
UC-1213

13. ABSTRACT *(Maximum 200 words)*

The U. S. Department of Energy (DOE), in conjunction with the U.S. wind industry, is supporting the development of technology for advanced, higher efficiency wind energy conversion systems. As part of the Turbine Development Program, the DOE, through the National Renewable Energy Laboratory, is conducting several full system modal tests of advanced wind turbine systems. This report supplements the field of full system modal testing by developing an advanced system identifications method which excites the structure with a measured frequency-rich input and then derives a complete state-space representation of the structure. After computing a state-space model of the wind turbine structure, the system identification technique calculates the inherent modal parameters such as natural frequencies, damping ratios, and mode shapes. Also, the system identification technique can be used to adjust an analytical model with input-output time series obtained from experimental structural tests.

14. SUBJECT TERMS
wind turbines, modal parameters

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL