

# State-Space System Realization With Input- and Output-Data Correlation

---

*Jer-Nan Juang*  
*Langley Research Center • Hampton, Virginia*

Available electronically at the following URL address: <http://techreports.larc.nasa.gov/ltrs/ltrs.html>

Printed copies available from the following:

NASA Center for AeroSpace Information  
800 Elkridge Landing Road  
Linthicum Heights, MD 21090-2934  
(301) 621-0390

National Technical Information Service (NTIS)  
5285 Port Royal Road  
Springfield, VA 22161-2171  
(703) 487-4650

## Abstract

*This paper introduces a general version of the information matrix consisting of the autocorrelation and cross-correlation matrices of the shifted input and output data. Based on the concept of data correlation, a new system realization algorithm was developed to create a model directly from input and output data. The algorithm starts by computing a special type of correlation matrix derived from the information matrix. The special correlation matrix provides information on the system-observability matrix and the state-vector correlation. A system model was then developed from the observability matrix in conjunction with other algebraic manipulations. This approach leads to several different algorithms for computing system matrices for use in representing the system model. The relationship of the new algorithms with other realization algorithms in the time and frequency domains is established with matrix factorization of the information matrix. Several examples are given to illustrate the validity and usefulness of these new algorithms.*

## 1. Introduction

Recently, system identification has gained much support for active control of flexible structures, including acoustic noise reduction, jitter-induced vibration suppression, and fine pointing of spacecraft antenna. In practice, control designs based on analytical models will not work when used the first time because often the analytical models used in the control designs are not accurate enough to meet specified performance requirements. As a result, most practicing engineers conduct experiments to either tune the control parameters or identify accurate mathematical models from input and output data. In addition to identifying system models, most robust control methods require information about model uncertainties.

System identification encompasses many approaches, perspectives, and techniques (refs. 1–9) and can be divided into two groups. One group of techniques uses the nonparametric approach (refs. 1–8) with the least-squares method to determine the input-output map. The input-output map is characterized by a system model which may not have any explicit physical meaning. These techniques are generally referred to as black box approaches and are noniterative. A second group of techniques uses the parametric approach (ref. 9) to determine system model parameters. The parameters may represent physical quantities such as structural stiffness or mass. Nonlinear mathematical optimization techniques are used to search for the optimal value of each parameter.

Both parametric and nonparametric approaches have many successful applications. While development and enhancement of the many individual techniques continue, a comprehensive yet coherent unification of the different methods is still needed. In the past decade, researchers have successfully provided frameworks to unify different techniques via system realization theory (ref. 1).

This paper is based on the need to improve system identification techniques such as those discussed in references 6–8. Computational time and numerical accuracy are an issue when measurement data are substantial. Section 3 of this paper describes an alternate procedure used to perform system identification more efficiently with the concept of data correlations presented in references 9 and 10. A new algorithm called “system realization using information matrix” (SRIM) is developed. The information matrix is similar to the matrix defined in reference 9 for

the frequency-domain analysis, but has a general form consisting of shifted input- and output-data correlations in the time domain or frequency domain. A special correlation matrix is introduced and computed from the information matrix. The special correlation matrix reduces to the shifted data correlation of the pulse response if the output is a free-decay response generated by a pulse input. To form a discrete time model, the SRIM algorithm includes several methods of different merit for computing system matrices, including the state matrix, the input and output matrices, and the direct-transmission matrix.

The eigensystem realization algorithm with data correlation (ERA/DC) (ref. 10) uses the shifted data correlation of the pulse response and factors the correlation matrix via singular-value decomposition to obtain the system matrices. The pulse response may be obtained with either a pulse input or computed from input and output data with the observer/Kalman filter identification (OKID) technique (ref. 11). Thus, the SRIM algorithm presented in this paper may be considered an extension of the ERA/DC for system identification directly from input and output data.

Section 4 of this paper describes how the mathematical framework developed for the SRIM algorithm is used to establish the relationship among different realization techniques and how the different algorithms may be derived from the information matrix. The basis vectors of the column space and the null space of the information matrix are the key elements that provide the link among the different realization algorithms. Matrix factorization of the information matrix is presented as the most important step in the theoretical development and computational procedure that allows unification of many different techniques.

Each section of this paper starts with background information and ends with numerical and experimental examples that illustrate the validity and usefulness of the algorithms presented in the paper. Comparisons of algorithms are also discussed to demonstrate the merit of different techniques for computing system matrices.

## 2. Symbols

$A$	state matrix, $n \times n$
$A_c$	continuous-time state matrix, $n \times n$
$B$	input-influence matrix, $n \times r$
$B_c$	continuous-time input-influence matrix
$\underline{b}_i$	$i$ th column of $B$ , $n \times 1$
$C$	output-influence matrix, $m \times n$
$C_a$	accelerometer output-influence matrix
$D$	direct-transmission matrix, $m \times r$
$\underline{d}_i$	$i$ th column of $D$ , $m \times 1$
$G(z_k)$	frequency-response function, $m \times r$ (at frequency variable $z_k$ )
$I_i$	identity matrix of order $i$
$K$	stiffness matrix
$k$	time index
$k_i$	$i$ th spring constant
$\ell$	data length
$M$	mass matrix

$m$	number of outputs
$m_i$	$i$ th mass
$n$	order of first-order system
$n_o$	number of zero singular values
$\mathcal{O}_p$	observability matrix, $pm \times n$
$\mathcal{O}_{pA}, \mathcal{O}_{p\Gamma}$	working matrices
$p$	integer determining maximum order of system
$\mathcal{R}, \bar{\mathcal{R}}$	information matrix, $p(m+r) \times p(m+r)$
$\mathcal{R}_{hh}$	fundamental SRIM correlation matrix, $pm \times pm$
$\mathcal{R}_{uu}, \tilde{\mathcal{R}}_{uu}$	autocorrelation of input vector, $pr \times pr$
$\mathcal{R}_{xu}$	cross correlation of state and input vectors, $n \times pr$
$\mathcal{R}_{xx}$	autocorrelation of state vector, $n \times n$
$\tilde{\mathcal{R}}_{xx}$	state-related correlation matrix, $n \times n$
$\mathcal{R}_{yu}, \tilde{\mathcal{R}}_{yu}$	cross correlation of output and input vectors, $pm \times pr$
$\mathcal{R}_{yx}$	cross correlation of output and state vectors, $pm \times n$
$\mathcal{R}_{yy}$	autocorrelation of output vector, $pm \times pm$
$r$	number of inputs
$\mathcal{T}_p$	Toeplitz matrix, $pm \times pr$
$\mathcal{U}$	left singular matrix
$\underline{\mathcal{U}}_m(k)$	special force matrix at time index $k$ , $m \times r$
$\mathcal{U}_n$	columns of left singular matrix corresponding to nonzero singular values
$\mathcal{U}_o$	columns of left singular matrix corresponding to zero singular values
$\mathcal{U}_{on}, \mathcal{U}_{o\mathcal{T}}$	working matrices
$\mathcal{U}_p(k)$	output matrix containing $u_p(k)$ up to $u_p(k+N-1)$ , $pr \times N$
$u(k)$	input-force vector at time index $k$ , $r \times 1$
$u_i(k)$	$i$ th input force at time index $k$
$u_p(k)$	vector containing $u(k)$ up to $u(k+p-1)$ , $rp \times 1$
$u'_N(k)$	vector containing $u(k)$ up to $u(k+N-1)$ , $pN \times 1$
$\mathcal{V}$	right singular matrix
$\mathcal{V}(k)$	process-noise vector at time index $k$
$w$	displacement vector
$w_i$	$i$ th element of $w$
$X(k)$	matrix containing $x(k)$ up to $x(k+N-1)$ , $n \times N$
$x(k)$	state vector at time index $k$ , $n \times 1$
$Y_p(k)$	output matrix containing $y_p(k)$ to $y_p(k+N-1)$ , $pm \times N$

$y(k)$	output-measurement vector at time index $k$ , $m \times 1$
$y_p(k)$	vector containing $y(k)$ up to $y(k + p - 1)$ , $pm \times 1$
$z_k$	frequency-domain variable
$\alpha, \beta, \Theta$	parameter matrices
$\alpha_i$	$i$ th ARX-coefficient matrix, $m \times m$ (associated with output vector $y(k + i)$ )
$\alpha(z_k)$	ARX-coefficient matrix, $m \times m$ (associated with output vector $y(z_k)$ at frequency variable $z_k$ )
$\beta_i$	$i$ th ARX-coefficient matrix, $m \times r$ (associated with input vector $u(k + i)$ )
$\beta(z_k)$	ARX-coefficient matrix, $m \times r$ (associated with input vector $u(z_k)$ at frequency variable $z_k$ )
$\Gamma$	working matrix
$\epsilon(k)$	measurement noise vector at time index $k$
$\zeta_i$	$i$ th damping term
$\Xi$	damping matrix
$\Sigma$	singular-value matrix
$\Sigma_n$	diagonal matrix containing nonzero singular values
$\sigma_i$	$i$ th singular value
$\Phi$	working matrix
$0_i$	zero-square matrix of order $i$
$0_{i \times j}$	zero-rectangular matrix of dimension $i$ by $j$
$\dagger$	pseudoinverse

### Abbreviations:

ARX	autoregressive exogeneous
ERA/DC	eigensystem realization algorithm with data correlation
FRF	frequency-response function
IDM	indirect method
OEM	output-error minimization method
OKID	observer/Kalman filter identification
SMI	subspace model identification
SRIM	system realization using information matrix
SV	singular value

### 3. System Realization Using Information Matrix (SRIM)

This section presents a system realization algorithm for computing system matrices to characterize the map from input to output. The system matrices are the state matrix  $A$ , the input matrix  $B$ , the output matrix  $C$ , and the direct-transmission matrix  $D$ . The key to the

SRIM algorithm is computation of the information matrix, which consists of autocorrelation and cross-correlation matrices of the shifted input and output data.

Section 3 starts with a description of a discrete-time state-space model and gives key definitions, such as the observability matrix and the Toeplitz matrix formed from the system matrices. The development of the state-space model realization used to compute the set of system matrices  $[A, B, C, D]$  follows. Next, the computational steps are provided for the algorithm. Finally, simulation and experimental examples are described.

### 3.1. State-Space Model

A deterministic, linear, time-invariant system is commonly represented by the following discrete-time state-space model:

$$\left. \begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned} \right\} \quad (1)$$

where  $x(k)$  is an  $n \times 1$  state vector at time index  $k$ ,  $u(k)$  is an  $r \times 1$  input vector corresponding to  $r$  inputs, and  $y(k)$  is an  $m \times 1$  output vector associated with  $m$  sensor measurements. The system matrices  $A, B, C$ , and  $D$  are unknown and will be determined from given input and output data, that is,  $u(k)$  and  $y(k)$  for  $k = 0, 1, 2, \dots, \ell$ .

Equation (1) can be written for various time shifts in a matrix form as

$$\begin{bmatrix} x(1) & x(2) & \cdots & x(\ell+1) \\ y(0) & y(1) & \cdots & y(\ell) \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x(0) & x(1) & \cdots & x(\ell) \\ u(0) & u(1) & \cdots & u(\ell) \end{bmatrix} \quad (2)$$

If the quantities  $u(k), y(k)$  for  $k = 0, 1, \dots, \ell$ , and  $x(k)$  for  $k = 0, 1, \dots, \ell + 1$  are known, then  $A, B, C$ , and  $D$  may be determined from equation (2) with the least-squares technique. Unfortunately, the time sequence of the state vector  $x(k)$  for  $k = 0, 1, \dots, \ell + 1$  is generally unknown and must either be estimated or measured, if possible. In practice, usually only the input and output sequences  $u(k)$  and  $y(k)$  for  $k = 1, \dots, \ell$ , respectively, are available. Therefore, other forms of system equations must be found for use in system identification.

With several algebraic manipulations, equations (1) produce

$$\begin{aligned} \begin{bmatrix} y(k) \\ y(k+1) \\ y(k+2) \\ \vdots \\ y(k+p-1) \end{bmatrix} &= \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{p-1} \end{bmatrix} x(k) \\ &+ \begin{bmatrix} D & & & & \mathbf{0} \\ CB & D & & & \\ CAB & CB & D & & \\ \vdots & \vdots & \vdots & \ddots & \\ CA^{p-2}B & CA^{p-3}B & CA^{p-4}B & \cdots & D \end{bmatrix} \begin{bmatrix} u(k) \\ u(k+1) \\ u(k+2) \\ \vdots \\ u(k+p-1) \end{bmatrix} \quad (3) \end{aligned}$$

where  $p$  is an integer depending on the size of the system model (the dimension of  $A$ ). The choice of  $p$  will be shown later. Let  $y_p(k)$ ,  $\mathcal{O}_p$ ,  $u_p(k)$ , and  $\mathcal{T}_p$  be defined as

$$\left. \begin{aligned} y_p(k) &= \begin{bmatrix} y(k) \\ y(k+1) \\ y(k+2) \\ \vdots \\ y(k+p-1) \end{bmatrix} & \mathcal{O}_p &= \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{p-1} \end{bmatrix} & u_p(k) &= \begin{bmatrix} u(k) \\ u(k+1) \\ u(k+2) \\ \vdots \\ u(k+p-1) \end{bmatrix} \\ \mathcal{T}_p &= \begin{bmatrix} D & & & & \mathbf{0} \\ CB & D & & & \\ CAB & CB & D & & \\ \vdots & \vdots & \vdots & \ddots & \\ CA^{p-2}B & CA^{p-3}B & CA^{p-4}B & \dots & D \end{bmatrix} \end{aligned} \right\} \quad (4)$$

Equation (3) thus becomes

$$y_p(k) = \mathcal{O}_p x(k) + \mathcal{T}_p u_p(k) \quad (5)$$

The matrix  $\mathcal{O}_p$  of dimension  $pm \times n$  is commonly called the observability matrix. The matrix  $\mathcal{T}_p$  of dimension  $pm \times pr$  is a generalized Toeplitz matrix. Note that  $\mathcal{T}_p$  is unique even though matrices  $A, B, C$ , and  $D$  are not unique because the system Markov parameters  $D$  and  $CA^k B$  are unique. The use of equation (3) to develop a system model is presented in section 3.2.

### 3.2. State-Space Model Realization

The goal of state-space system identification is to determine the unknown matrices  $A, B, C$ , and  $D$  which are embedded in matrices  $\mathcal{O}_p$  and  $\mathcal{T}_p$ . One approach starts with computing  $\mathcal{O}_p$  and  $\mathcal{T}_p$  from known input and output data. With  $\mathcal{O}_p$  computed, the output matrix  $C$  is the first  $m$  rows of  $\mathcal{O}_p$ . Define  $\mathcal{O}_p(m+1 : pm, :)$  as the matrix consisting of the last  $(p-1)m$  rows (from  $(m+1)$ th row to the  $(pm)$ th row) and all columns of  $\mathcal{O}_p$ . Similarly, define  $\mathcal{O}_p(1 : (p-1)m, :)$  as the matrix formed with the first  $(p-1)m$  rows and all columns of  $\mathcal{O}_p$  as follows:

$$\mathcal{O}_p(m+1 : pm, :) = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{p-1} \end{bmatrix} \quad \mathcal{O}_p(1 : (p-1)m, :) = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{p-2} \end{bmatrix} \quad (6)$$

Form the following equality:

$$\mathcal{O}_p(m+1 : pm, :) = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{p-1} \end{bmatrix} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{p-2} \end{bmatrix} A = \mathcal{O}_p(1 : (p-1)m, :) A \quad (7)$$

The colon in place of a subscript denotes the entire corresponding row or column. The state matrix can then be computed by

$$A = \mathcal{O}_p^\dagger(1 : (p-1)m, :)\mathcal{O}_p(m+1 : pm, :) \quad (8)$$

where  $\dagger$  means the pseudoinverse. The integer  $p$  should be chosen so that the matrix  $\mathcal{O}_p(m+1 : pm, :)$  of dimension  $(p-1)m \times n$  has rank larger than or equal to  $n$  as follows:

$$(p-1)m \geq n \implies p \geq \frac{n}{m} + 1 \quad (9)$$

where  $n$  is the order of the system.

Similarly, the first  $m$  rows and the first  $r$  columns of the matrix  $\mathcal{T}_p$  (eqs. (4)) constitute the direct-transmission matrix  $D$ . Define  $\mathcal{T}(m+1 : (p-1)m, 1 : r)$  as the matrix formed by deleting the first  $m$  rows of the first  $r$  columns of  $\mathcal{T}_p$  as follows:

$$\mathcal{T}(m+1 : (p-1)m, 1 : r) = \begin{bmatrix} CB \\ CAB \\ \vdots \\ CA^{p-2}B \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{p-2} \end{bmatrix} B = \mathcal{O}_p(1 : (p-1)m, :)B \quad (10)$$

Equation (10) produces

$$B = \mathcal{O}_p^\dagger(1 : (p-1)m, :)\mathcal{T}(m+1 : (p-1)m, 1 : r) \quad (11)$$

To determine  $\mathcal{O}_p$  and  $\mathcal{T}_p$ , first expand the vector equation (eq. (5)) to a matrix equation as follows:

$$Y_p(k) = \mathcal{O}_p X(k) + \mathcal{T}_p U_p(k) \quad (12)$$

where

$$\left. \begin{aligned} X(k) &= [x(k) \quad x(k+1) \quad \cdots \quad x(k+N-1)] \\ Y_p(k) &= [y_p(k) \quad y_p(k+1) \quad \cdots \quad y_p(k+N-1)] \\ &= \begin{bmatrix} y(k) & y(k+1) & \cdots & y(k+N-1) \\ y(k+1) & y(k+2) & \cdots & y(k+N) \\ \vdots & \vdots & \ddots & \vdots \\ y(k+p-1) & y(k+p) & \cdots & y(k+p+N-2) \end{bmatrix} \\ U_p(k) &= [u_p(k) \quad u_p(k+1) \quad \cdots \quad u_p(k+N-1)] \\ &= \begin{bmatrix} u(k) & u(k+1) & \cdots & u(k+N-1) \\ u(k+1) & u(k+2) & \cdots & u(k+N) \\ \vdots & \vdots & \ddots & \vdots \\ u(k+p-1) & u(k+p) & \cdots & u(k+p+N-2) \end{bmatrix} \end{aligned} \right\} \quad (13)$$

The integer  $N$  must be sufficiently large so that the rank of  $Y_p(k)$  and  $U_p(k)$  is at least equal to the rank of  $\mathcal{O}_p$ . Equation (12) is the key equation used to solve for  $\mathcal{O}_p$  and  $\mathcal{T}_p$  and includes the input- and output-data information up to the data point  $k+p+N-2$ . Because the data matrices  $Y_p(k)$  and  $U_p(k)$  are the only information given, it is necessary to focus on these two matrices to extract information necessary to determine the system matrices  $A, B, C$ , and  $D$ .

The following quantities are defined as:

$$\left. \begin{aligned} \mathcal{R}_{yy} &= \frac{1}{N} Y_p(k) Y_p^T(k) \\ \mathcal{R}_{yu} &= \frac{1}{N} Y_p(k) U_p^T(k) \\ \mathcal{R}_{uu} &= \frac{1}{N} U_p(k) U_p^T(k) \\ \mathcal{R}_{xx} &= \frac{1}{N} X(k) X^T(k) \\ \mathcal{R}_{yx} &= \frac{1}{N} Y_p(k) X^T(k) \\ \mathcal{R}_{xu} &= \frac{1}{N} X(k) U_p^T(k) \end{aligned} \right\} \quad (14)$$

where  $N = \ell - p$ , with  $\ell$  being the data length and  $p$  the data shift. The quantities  $\mathcal{R}_{yy}$ ,  $\mathcal{R}_{uu}$ , and  $\mathcal{R}_{xx}$  are symmetric matrices. The square matrices  $\mathcal{R}_{yy}$  ( $mp \times mp$ ),  $\mathcal{R}_{uu}$  ( $rp \times rp$ ), and  $\mathcal{R}_{xx}$  ( $n \times n$ ) are the autocorrelations of the output data  $y$  with time shifts, the input data  $u$  with time shifts, and the state vector  $x$ , respectively. The rectangular matrices  $\mathcal{R}_{yu}$  ( $mp \times rp$ ),  $\mathcal{R}_{yx}$  ( $mp \times n$ ), and  $\mathcal{R}_{xu}$  ( $n \times rp$ ) represent the cross correlations of the output data  $y$  and the input data  $u$ , the output data  $y$  and the state vector  $x$ , and the state vector  $x$  and input data  $u$ , respectively. When the integer  $N$  is sufficiently large, the quantities defined in equations (14) approximate expected values in the statistical sense if the input and output data are stationary processes satisfying the ergodic property.

When considering equations (14), postmultiplying equation (12) by  $U_p^T(k)$  and then dividing the result by  $N$  will yield

$$\mathcal{R}_{yu} = \mathcal{O}_p \mathcal{R}_{xu} + \mathcal{I}_p \mathcal{R}_{uu} \quad (15)$$

which, if  $\mathcal{R}_{uu}^{-1}$  exists, yields

$$\mathcal{I}_p = (\mathcal{R}_{yu} - \mathcal{O}_p \mathcal{R}_{xu}) \mathcal{R}_{uu}^{-1} \quad (16)$$

The matrix inverse  $\mathcal{R}_{uu}^{-1}$  exists only when integers  $p$  and  $N$  are chosen so that  $\mathcal{R}_{uu}$  at least has rank  $rp$ . Similarly, postmultiplying equation (12) by  $Y_p^T(k)$  yields

$$\mathcal{R}_{yy} = \mathcal{O}_p \mathcal{R}_{yx}^T + \mathcal{I}_p \mathcal{R}_{yu}^T \quad (17)$$

and postmultiplying equation (12) by  $X^T(k)$  gives

$$\mathcal{R}_{yx} = \mathcal{O}_p \mathcal{R}_{xx} + \mathcal{I}_p \mathcal{R}_{xu}^T \quad (18)$$

Substituting equation (16) for  $\mathcal{I}_p$  into equations (17) and (18), and then substituting the resulting equation for  $\mathcal{R}_{yx}$  into equation (17) will produce

$$\mathcal{R}_{yy} - \mathcal{R}_{yu} \mathcal{R}_{uu}^{-1} \mathcal{R}_{yu}^T = \mathcal{O}_p \mathcal{R}_{xx} \mathcal{O}_p^T - \mathcal{O}_p \mathcal{R}_{xu} \mathcal{R}_{uu}^{-1} \mathcal{R}_{xu}^T \mathcal{O}_p^T \quad (19)$$

Now define

$$\mathcal{R}_{hh} = \mathcal{R}_{yy} - \mathcal{R}_{yu} \mathcal{R}_{uu}^{-1} \mathcal{R}_{yu}^T \quad (20)$$

and

$$\tilde{\mathcal{R}}_{xx} = \mathcal{R}_{xx} - \mathcal{R}_{xu} \mathcal{R}_{uu}^{-1} \mathcal{R}_{xu}^T \quad (21)$$

Equation (19) becomes

$$\mathcal{R}_{hh} = \mathcal{O}_p \tilde{\mathcal{R}}_{xx} \mathcal{O}_p^T \quad (22)$$

Equation (22) is the key equation for determining system matrices  $A$  and  $C$ . The quantity  $\mathcal{R}_{hh}$  is determined from the output-autocorrelation matrix  $\mathcal{R}_{yy}$  minus the product of the cross-correlation matrix  $\mathcal{R}_{yu}$  and its transpose weighted by the inverse of the input-autocorrelation matrix  $\mathcal{R}_{uu}$ . The quantity  $\mathcal{R}_{hh}$  exists only if the input-autocorrelation matrix  $\mathcal{R}_{uu}$  is invertible. The symmetric matrix  $\mathcal{R}_{uu}$  is invertible if the input signal  $u(k)$  for  $k = 1, 2, \dots, \ell$  is rich and persistent, which results in a matrix  $U_p(k)$  of full rank, that is,  $rp$ . Assume that the input signal  $u(i)$  for  $i \geq k$  is uncorrelated with the state vector  $x(k)$  at time step  $k$ . Stated differently, the current and future input data are uncorrelated with the current state. In this case, the cross-correlation matrix  $\mathcal{R}_{xu}$  becomes an  $n \times rp$  zero matrix and the matrix  $\tilde{\mathcal{R}}_{xx}$  (eq. (21)) reduces to  $\mathcal{R}_{xx}$ . For example, if the input  $u$  is a zero-mean, white, random Gaussian signal, then  $\tilde{\mathcal{R}}_{xx} = \mathcal{R}_{xx}$  when the data length is sufficiently long ( $N \rightarrow \infty$  in theory).

Define  $\mathcal{R}$  as

$$\mathcal{R} = \begin{bmatrix} \mathcal{R}_{yy} & \mathcal{R}_{yu} \\ \mathcal{R}_{yu}^T & \mathcal{R}_{uu} \end{bmatrix} = \begin{bmatrix} Y_p(k) \\ U_p(k) \end{bmatrix} [Y_p^T(k) \quad U_p^T(k)] \quad (23)$$

The matrix  $\mathcal{R}$  is defined here as the *information matrix* and is formed by the correlation matrices  $\mathcal{R}_{yy}$ ,  $\mathcal{R}_{yu}$ , and  $\mathcal{R}_{uu}$  of shifted input and output data. The information matrix contains all information necessary to compute the system matrices  $A$ ,  $B$ ,  $C$ , and  $D$ . Factoring  $\mathcal{R}$  yields

$$\mathcal{R} = \begin{bmatrix} \mathcal{R}_{yy} & \mathcal{R}_{yu} \\ \mathcal{R}_{yu}^T & \mathcal{R}_{uu} \end{bmatrix} = \begin{bmatrix} I_{pm} & \mathcal{R}_{yu} \mathcal{R}_{uu}^{-1} \\ 0_{pr \times pm} & I_{pr} \end{bmatrix} \begin{bmatrix} \mathcal{R}_{hh} & 0_{pm \times pr} \\ 0_{pr \times pm} & \mathcal{R}_{uu} \end{bmatrix} \begin{bmatrix} I_{pm} & 0_{pm \times pr} \\ \mathcal{R}_{uu}^{-1} \mathcal{R}_{yu} & I_{pr} \end{bmatrix} \quad (24)$$

where  $I_{pm}$  (or  $I_{pr}$ ) is an identity matrix of order  $pm$  (or  $pr$ ), and  $0_{pm \times pr}$  (or  $0_{pr \times pm}$ ) is a  $pm \times pr$  (or  $pr \times pm$ ) zero matrix. The product of a matrix and its transpose is either a positive-semidefinite or a positive-definite matrix, depending on the rank of the matrix itself. Therefore, the matrix product on the left side of equation (24) is a positive-semidefinite or a positive-definite matrix. In the matrix triple product on the right side of equation (24), the left matrix and its transpose (i.e., the right matrix) are both of full rank. This means that  $\mathcal{R}_{hh}$  must be a positive-semidefinite or a positive-definite matrix as follows:

$$\mathcal{R}_{hh} \geq 0 \quad (25)$$

for the case when  $\mathcal{R}_{uu} > 0$  (positive definite), which is required for the existence of  $\mathcal{R}_{uu}^{-1}$ .

The left side of equation (22) (the symmetric matrix  $\mathcal{R}_{hh}$ ) is known from input and output data and the right side is formed from the product of the rectangular matrix  $\mathcal{O}_p$  of dimension  $mp \times n$ , which is the symmetric matrix  $\tilde{\mathcal{R}}_{xx}$  of dimension  $n \times n$  and the transpose of  $\mathcal{O}_p$ . It is clear that the matrix  $\mathcal{R}_{hh}$  must be factored into three matrices to solve for the observability matrix  $\mathcal{O}_p$ , and this approach is taken in section 3.2.1.

### 3.2.1. Computation of $A$ and $C$

Section 3.2.1 shows two methods for computing  $A$  and  $C$ . One method decomposes the full matrix  $\mathcal{R}_{hh}$  and is called the full decomposition method. The other method decomposes a portion of  $\mathcal{R}_{hh}$  and is called the partial decomposition method.

*3.2.1.1. Full decomposition method.* Given matrix  $\mathcal{R}_{hh}$  computed from the input and output data, the matrix decomposition method starts with factoring  $\mathcal{R}_{hh}$  into the product of three matrices. Singular-value decomposition is the obvious choice to perform the matrix factorization.

Taking the singular-value decomposition of the symmetric matrix  $\mathcal{R}_{hh}$  yields

$$\mathcal{R}_{hh} = \mathcal{U}\Sigma^2\mathcal{U}^T = [\mathcal{U}_n \quad \mathcal{U}_o] \begin{bmatrix} \Sigma_n^2 & 0_{n \times n_o} \\ 0_{n_o \times n} & 0_{n_o} \end{bmatrix} \begin{bmatrix} \mathcal{U}_n^T \\ \mathcal{U}_o^T \end{bmatrix} = \mathcal{U}_n \Sigma_n^2 \mathcal{U}_n^T \quad (26)$$

The integer  $n_o = pm - n$  is the number of dependent columns in  $\mathcal{R}_{hh}$ ,  $0_{n \times n_o}$  is an  $n \times n_o$  zero matrix, and  $0_{n_o}$  is a square-zero matrix of order  $n_o$ . The  $pm \times n$  matrix  $\mathcal{U}_n$  corresponds to the  $n$  nonzero singular values in the diagonal matrix  $\Sigma_n$ , and the  $pm \times n_o$  matrix  $\mathcal{U}_o$  is associated with the  $n_o$  zero singular values. Combining equations (22) and (26) produces

$$\mathcal{R}_{hh} = \mathcal{O}_p \tilde{\mathcal{R}}_{xx} \mathcal{O}_p^T = \mathcal{U}_n \Sigma_n^2 \mathcal{U}_n^T \quad (27)$$

The last equality produces one solution each for  $\mathcal{O}_p$  and  $\tilde{\mathcal{R}}_{xx}$  as follows:

$$\mathcal{O}_p = \mathcal{U}_n \quad (28)$$

and

$$\tilde{\mathcal{R}}_{xx} = \Sigma_n^2 \quad (29)$$

Equation (28) implies that the  $pm \times n$  matrix  $\mathcal{U}_n$  computed from the correlation matrix  $\mathcal{R}_{hh}$  is a representation of the observability matrix  $\mathcal{O}_p$  and can be used to solve for the output matrix  $C$  and the state matrix  $A$  with equation (8). The first  $m$  rows of  $\mathcal{U}_n$  constitute the output matrix  $C$ .

Equation (29) gives the correlation  $\tilde{\mathcal{R}}_{xx}$  (eq. (21)) as the singular-value matrix  $\Sigma_n^2$  of the correlation matrix  $\mathcal{R}_{hh}$ . For the case where the input  $u$  is a zero-mean white-noise sequence, the correlation  $\tilde{\mathcal{R}}_{xx}$  reduces to  $\mathcal{R}_{xx}$  (eqs. (14)), which is the correlation of the state vector  $x$ . The diagonal nature of  $\Sigma_n^2$  implies that all individual elements of the state vector  $x$  are linearly independent and orthogonal (uncoupled). Each individual element of the state vector  $x(k)$  represents one coordinate. The importance of each coordinate can then be measured by the magnitude of the corresponding singular value.

Let the diagonal matrix  $\Sigma_n$  be denoted by

$$\Sigma_n = \text{diagonal}[\sigma_1, \sigma_2, \dots, \sigma_n] = \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_n \end{bmatrix} \quad (30)$$

with monotonically nonincreasing  $\sigma_i$  ( $i = 1, 2, \dots, n$ ),  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n \geq 0$ . Accordingly, the strength of the elements (coordinates) in the state vector  $x(k)$  can be quantified by the singular values. Assume that the singular values  $\sigma_{i+1}, \dots, \sigma_n$  are relatively small and negligible in the sense that they contain more noise information than system information. As a result, the coordinates corresponding to the singular values  $\sigma_{i+1}, \dots, \sigma_n$  are negligible compared with the other coordinates. The order of the system may then be reduced from  $n$  to  $i$  by deleting singular values  $\sigma_{i+1}, \dots, \sigma_n$ .

In practice, none of the singular values will be identically zero because of system uncertainties and measurement noise, implying that  $n_o = 0$ ,  $n = pm$ , and  $\mathcal{U}_o = []$  (empty) without singular-values truncation. The observability matrix  $\mathcal{O}_p$  (eq. (28)) becomes a square matrix because  $\mathcal{U}_n$  obtained from equation (27) is a  $pm \times pm$  matrix. The equality (eq. (9)) is violated, indicating that equation (8) cannot be used to solve for the state matrix  $A$ . If none of the singular

values are zero, at least  $m$  smallest singular values must be considered as zero before using the full decomposition method. In other words, the last  $m$  columns of  $\mathcal{U}_n$  must be truncated and treated as  $\mathcal{U}_o$ . To overcome this problem, a different method is presented in section 3.2.1.2.

*3.2.1.2. Partial decomposition method.* Regardless of which integer  $p$  is chosen, the minimum value of  $n_o$  must be  $m$  (the number of outputs) to make  $n < pm$ , which will then satisfy the equality constraint in equation (27). There is one way of avoiding any singular-values truncation. Instead of taking the singular-value decomposition of the  $pm \times pm$  square matrix  $\mathcal{R}_{hh}$ , factor only part of the matrix as follows:

$$\mathcal{R}_{hh}(:, 1 : (p-1)m) = \mathcal{U}\Sigma^2\mathcal{V}^T = [\mathcal{U}'_n \quad \mathcal{U}'_o] \begin{bmatrix} \Sigma_n^2 & 0_{n \times n_o} \\ 0_{n'_o \times n} & 0_{n'_o \times n_o} \end{bmatrix} \begin{bmatrix} \mathcal{V}_n^T \\ \mathcal{V}'_o{}^T \end{bmatrix} = \mathcal{V}_n \Sigma_n^2 \mathcal{V}_n^T \quad (31)$$

The dimension of  $\mathcal{R}_{hh}(:, 1 : (p-1)m)$  is  $pm \times (p-1)m$ , meaning there are more rows than columns. The integer  $n_o$  indicates the number of zero singular values and also the number of columns of  $\mathcal{V}_o$ . The integer  $n'_o$  is the number of columns of  $\mathcal{U}'_o$  that are orthogonal to the columns of  $\mathcal{U}'_n$ . For noisy data, there are no zero singular values, that is,  $n_o = 0$ . If no singular values are truncated,  $n'_o = m$  is obtained. If some small singular values are truncated,  $n'_o$  becomes the sum of  $m$  and the number of truncated singular values. Stated differently, there are at least  $m$  columns of  $\mathcal{U}'_o$  that are orthogonal to the columns of  $\mathcal{U}'_n$  in equation (31). From equation (27), it is easy to show that

$$\mathcal{R}_{hh}(:, 1 : (p-1)m) = \mathcal{O}_p \tilde{\mathcal{R}}_{xx} \mathcal{O}_p^T(:, 1 : (p-1)m) = \mathcal{U}'_n \Sigma_n^2 \mathcal{V}_n^T \quad (32)$$

which yields the following equations:

$$\mathcal{O}_p = \mathcal{U}'_n \quad (33a)$$

and

$$\tilde{\mathcal{R}}_{xx} \mathcal{O}_p^T(:, 1 : (p-1)m) = \Sigma_n^2 \mathcal{V}_n^T \quad (33b)$$

Equation (33b) does not imply that  $\tilde{\mathcal{R}}_{xx} = \Sigma_n^2$  or  $\mathcal{O}_p^T(:, 1 : (p-1)m) = \mathcal{V}_n^T$  if equation (33a) is satisfied. One important feature of this approach is that there are always enough columns of  $\mathcal{U}'_o$  available for computing  $B$  and  $D$  with or without singular-values truncation. The disadvantage is that the singular values no longer represent the correlation of the state vector.

### 3.2.2. Computation of $B$ and $D$

Similar to computing  $A$  and  $C$ , three methods are available for computing  $B$  and  $D$ . The first method, called the indirect method, uses the column vectors that are orthogonal to the column vectors of the observability matrix. The second method makes direct use of the observability matrix and is referred to as the direct method. The third method minimizes output error between the measured output and the reconstructed output. The reconstructed output is the output time history obtained by using the input time history to drive the identified system model represented by the computed matrices  $A$ ,  $B$ ,  $C$ , and  $D$ .

*3.2.2.1. Indirect method.* With matrices  $A$  and  $C$  known, the input matrix  $B$  and the direct transmission matrix  $D$  can be computed from the Toeplitz matrix  $\mathcal{T}_p$  (eqs. (4)). To formulate an equation to solve for  $\mathcal{T}_p$ , the term associated with the observability matrix  $\mathcal{O}_p$  (eq. (15)) must be eliminated.

After considering equations (26) and (28), premultiplying equation (15) by  $\mathcal{U}_o^T$  and using the orthogonality property of  $\mathcal{U}_n$  and  $\mathcal{U}_o$  yield

$$\mathcal{U}_o^T \mathcal{R}_{yu} = \mathcal{U}_o^T \mathcal{T}_p \mathcal{R}_{uu}$$

Postmultiplying the above equation by  $\mathcal{R}_{uu}^{-1}$  results in

$$\mathcal{U}_o^T \mathcal{T}_p = \mathcal{U}_o^T \mathcal{R}_{yu} \mathcal{R}_{uu}^{-1} \quad (34)$$

Equation (34) is the fundamental equation used to solve for the input matrix  $B$  and the direct transmission matrix  $D$ . Note that equation (34) does not imply that  $\mathcal{T}_p = \mathcal{R}_{yu} \mathcal{R}_{uu}^{-1}$  because  $\mathcal{U}_o^T$  is a rectangular matrix of dimension  $n_o \times mp$  with  $n_o < mp$ . The right side of equation (34) is a known quantity and the left side contains the matrix  $\mathcal{T}_p$ , which is partially known to include  $A$  and  $C$  and partially unknown to include  $B$  and  $D$ . Therefore, the matrix  $\mathcal{T}_p$  must be partitioned into two parts to extract matrices  $B$  and  $D$ .

Let  $\mathcal{T}_p$  be partitioned as

$$\mathcal{T}_p = [\mathcal{T}_p(:, 1:r) \quad \mathcal{T}_p(:, r+1:2r) \quad \cdots \quad \mathcal{T}_p(:, (p-1)r+1:pr)] \quad (35)$$

From equations (4) (for definitions of  $\mathcal{T}_p$  and  $\mathcal{O}_p$ ) and equation (28) obtain

$$\left. \begin{aligned} \mathcal{T}_p(:, 1:r) &= \begin{bmatrix} D \\ \mathcal{U}_n(1:(p-1)m, :)B \end{bmatrix} \\ \mathcal{T}_p(:, r+1:2r) &= \begin{bmatrix} 0_{m \times r} \\ D \\ \mathcal{U}_n(1:(p-2)m, :)B \end{bmatrix} \\ \mathcal{T}_p(:, 2r+1:3r) &= \begin{bmatrix} 0_{2m \times r} \\ D \\ \mathcal{U}_n(1:(p-3)m, :)B \end{bmatrix} \\ &\vdots \\ \mathcal{T}_p(:, (p-1)r+1:pr) &= \begin{bmatrix} 0_{(p-1)m \times r} \\ D \end{bmatrix} \end{aligned} \right\} \quad (36)$$

with  $0_{i \times j}$  being a zero matrix of dimension  $i \times j$ . The product of  $\mathcal{U}_o^T \mathcal{T}_p$  becomes

$$\left. \begin{aligned} \mathcal{U}_o^T \mathcal{T}_p(:, 1:r) &= \mathcal{U}_o^T(:, 1:m)D \\ &\quad + \mathcal{U}_o^T(:, m+1:pm)\mathcal{U}_n(1:(p-1)m, :)B \\ \mathcal{U}_o^T \mathcal{T}_p(:, r+1:2r) &= \mathcal{U}_o^T(:, m+1:2m)D \\ &\quad + \mathcal{U}_o^T(:, 2m+1:pm)\mathcal{U}_n(1:(p-2)m, :)B \\ \mathcal{U}_o^T \mathcal{T}_p(:, 2r+1:3r) &= \mathcal{U}_o^T(:, 2m+1:3m)D \\ &\quad + \mathcal{U}_o^T(:, 3m+1:pm)\mathcal{U}_n(1:(p-3)m, :)B \\ &\vdots \\ \mathcal{U}_o^T \mathcal{T}_p(:, (p-1)r+1:pr) &= \mathcal{U}_o^T(:, (p-1)m+1:pm)D \end{aligned} \right\} \quad (37)$$

Equations (37) can be rewritten in the following matrix form:

$$\mathcal{U}_{o\mathcal{T}} = \mathcal{U}_{on} \begin{bmatrix} D \\ B \end{bmatrix} \quad (38)$$

where

$$\mathcal{U}_{o\mathcal{T}} = \begin{bmatrix} \mathcal{U}_o^T \mathcal{T}_p(:, 1:r) \\ \mathcal{U}_o^T \mathcal{T}_p(:, r+1:2r) \\ \mathcal{U}_o^T \mathcal{T}_p(:, 2r+1:3r) \\ \vdots \\ \mathcal{U}_o^T \mathcal{T}_p(:, (p-1)r+1:pr) \end{bmatrix}$$

$$\mathcal{U}_{on} = \begin{bmatrix} \mathcal{U}_o^T(:, 1:m) & \mathcal{U}_o^T(:, m+1:pm) \mathcal{U}_n(1:(p-1)m,:) \\ \mathcal{U}_o^T(:, m+1:2m) & \mathcal{U}_o^T(:, 2m+1:pm) \mathcal{U}_n(1:(p-2)m,:) \\ \mathcal{U}_o^T(:, 2m+1:3m) & \mathcal{U}_o^T(:, 3m+1:pm) \mathcal{U}_n(1:(p-3)m,:) \\ \vdots & \vdots \\ \mathcal{U}_o^T(:, (p-1)m+1:pm) & 0_{n_o \times n} \end{bmatrix}$$

The dimension of  $\mathcal{U}_{o\mathcal{T}}$  is  $pn_o \times pr$  and the dimension of  $\mathcal{U}_{on}$  is  $pn_o \times (m+n)$ . Let the right side of equation (34) be denoted by

$$\mathcal{U}_{o\mathcal{R}} = \mathcal{U}_o^T \mathcal{R}_{yu} \mathcal{R}_{uu}^{-1} \quad (39)$$

where  $\mathcal{U}_{o\mathcal{R}}$  is an  $n_o \times pr$  matrix. Equation (38) shows that  $\mathcal{U}_{o\mathcal{T}}$  is thus given by

$$\mathcal{U}_{o\mathcal{T}} = \begin{bmatrix} \mathcal{U}_{o\mathcal{R}}(:, 1:r) \\ \mathcal{U}_{o\mathcal{R}}(:, r+1:2r) \\ \mathcal{U}_{o\mathcal{R}}(:, 2r+1:3r) \\ \vdots \\ \mathcal{U}_{o\mathcal{R}}(:, (p-1)r+1:pr) \end{bmatrix} \quad (40)$$

and matrices  $B$  and  $D$  can be computed by

$$\begin{bmatrix} D \\ B \end{bmatrix} = \mathcal{U}_{on}^\dagger \mathcal{U}_{o\mathcal{T}} \quad (41)$$

The first  $m$  rows of  $\mathcal{U}_{on}^\dagger \mathcal{U}_{o\mathcal{T}}$  form the matrix  $D$ , and the last  $n$  rows produce the matrix  $B$ .

Equation (41) has a unique least-squares solution for matrices  $B$  and  $D$  only if the matrix  $\mathcal{U}_{on}$  has more rows than columns. Since the dimension of  $\mathcal{U}_{on}$  is  $pn_o \times (m+n)$ , the integer  $p$  must be chosen so that  $pn_o \geq (m+n)$ , where  $n_o = pm - n$ , with  $n$  being the order of the system. For example, if  $p$  is chosen to be  $(p-1)m \geq n$ , then the minimum requirement for  $n_o$  is  $n_o = m$ , which indicates that the order of the system must be determined so that  $pn_o \geq (m+n)$  is satisfied, particularly for the case where all singular values beyond  $\sigma_n$  (eq. (30)), that is,  $\sigma_{n+1}, \dots, \sigma_{pm}$ , are small quantities rather than zeros.

For small  $n_o$ , computing matrices  $B$  and  $D$  from equation (41) is efficient in time. In practice, the integer  $n_o$  results from truncating small but nonzero singular values. The truncation error may in turn introduce considerable error in the computed results for  $B$  and  $D$ . An alternate

method for computing  $B$  and  $D$  without the matrix  $\mathcal{U}_o$  associated with the zero singular values is presented in section 3.2.2.2.

*3.2.2.2. Direct method.* Instead of using  $\mathcal{U}_o$  to derive equation (34), the direct method depends on the observability matrix  $\mathcal{O}_p$  to formulate an equation to solve for matrices  $B$  and  $D$ . The approach used to derive the direct method is similar to the approach used for the indirect method.

First, use the notation  $X(k)$  defined in equations (13) and the state equations (eqs. (1)) to form

$$\begin{aligned} X(k+1) &= [x(k+1) \quad x(k+2) \quad \cdots \quad x(k+N)] \\ &= AX(k) + Bu'_N(k) \end{aligned} \quad (42)$$

where  $u'_N(k)$  is defined as

$$u'_N(k) = [u(k) \quad u(k+1) \quad \cdots \quad u(k+N-1)] \quad (43)$$

Substituting equation (42) into equation (12) yields

$$\begin{aligned} Y_p(k+1) &= \mathcal{O}_p X(k+1) + \mathcal{T}_p U_p(k+1) \\ &= \mathcal{O}_p AX(k) + \mathcal{O}_p Bu'_N(k) + \mathcal{T}_p U_p(k+1) \\ &= \mathcal{O}_p AX(k) + [\mathcal{O}_p B \quad \mathcal{T}_p] \begin{bmatrix} u'_N(k) \\ U_p(k+1) \end{bmatrix} \\ &= \mathcal{O}_p AX(k) + [\mathcal{O}_p B \quad \mathcal{T}_p] U_{p+1}(k+1) \end{aligned} \quad (44)$$

From equation (12), the least-squares solution for  $X(k)$  is

$$X(k) = \mathcal{O}_p^\dagger [Y_p(k) - \mathcal{T}_p U_p(k)] \quad (45)$$

With equation (45) for  $X(k)$ , equation (44) produces

$$\begin{aligned} Y_p(k+1) - \mathcal{O}_p A \mathcal{O}_p^\dagger Y_p(k) &= -\mathcal{O}_p A \mathcal{O}_p^\dagger \mathcal{T}_p U_p(k) + [\mathcal{O}_p B \quad \mathcal{T}_p] U_{p+1}(k+1) \\ &= \{[\mathcal{O}_p B \quad \mathcal{T}_p] - [\mathcal{O}_p A \mathcal{O}_p^\dagger \mathcal{T}_p \quad 0_{pm \times r}]\} U_{p+1}(k+1) \\ &= \Gamma U_{p+1}(k+1) \end{aligned} \quad (46)$$

where  $0_{pm \times r}$  is a  $pm \times r$  zero matrix and

$$\Gamma = [\mathcal{O}_p B \quad \mathcal{T}_p] - [\mathcal{O}_p A \mathcal{O}_p^\dagger \mathcal{T}_p \quad 0_{pm \times r}] \quad (47)$$

Note that  $\mathcal{O}_p B$  is a  $pm \times r$  matrix, whereas  $\mathcal{O}_p A \mathcal{O}_p^\dagger \mathcal{T}_p$  is a  $pm \times pr$  matrix because  $\mathcal{T}_p$  is a  $pm \times pr$  matrix. Since the dimensions of  $\mathcal{O}_p B$  and  $\mathcal{O}_p A \mathcal{O}_p^\dagger \mathcal{T}_p$  are different, they may not be

directly added to become a single matrix. Similarly,  $\mathcal{T}_p$  and  $0_{pm \times r}$  cannot be directly subtracted. However, either  $[\mathcal{O}_p B \quad \mathcal{T}_p]$  or  $[\mathcal{O}_p A \mathcal{O}_p^\dagger \mathcal{T}_p \quad 0_{pm \times r}]$  is a  $pm \times (pr + r)$  matrix. Postmultiplying equation (46) by  $U_{p+1}^T(k+1)$  thus results in

$$\tilde{\mathcal{R}}_{yu}(m+1 : (p+1)m, :) - \mathcal{O}_p A \mathcal{O}_p^\dagger \tilde{\mathcal{R}}_{yu}(1 : pm, :) = \Gamma \tilde{\mathcal{R}}_{uu} \quad (48)$$

where

$$\left. \begin{aligned} \tilde{\mathcal{R}}_{uu} &= \frac{1}{N} U_{p+1}(k+1) U_{p+1}^T(k+1) \\ \tilde{\mathcal{R}}_{yu} &= \frac{1}{N} Y_{p+1}(k+1) U_{p+1}^T(k+1) \end{aligned} \right\} \quad (49)$$

Now, premultiplying equations (49) by  $\mathcal{O}_p^\dagger$  and postmultiplying by  $\tilde{\mathcal{R}}_{uu}^{-1}$  yields

$$\mathcal{O}_p^\dagger \tilde{\mathcal{R}}_{yu}(m+1 : (p+1)m, :) \tilde{\mathcal{R}}_{uu}^{-1} - A \mathcal{O}_p^\dagger \tilde{\mathcal{R}}_{yu}(1 : pm, :) \tilde{\mathcal{R}}_{uu}^{-1} = \mathcal{O}_p^\dagger \Gamma \quad (50)$$

where  $\mathcal{O}_p^\dagger \mathcal{O}_p = I_n$  has been applied. With input and output data given, and matrices  $A$  and  $C$  determined, all quantities on the left side of equation (50) are computable. The unknown matrices  $B$  and  $D$  are embedded in the matrix  $\Gamma$ . Similar to equations (37) for computing the product of  $U_o^T \mathcal{T}_p$ , compute  $\mathcal{O}_p^\dagger \Gamma$  as follows:

$$\mathcal{O}_p^\dagger \Gamma = [B \quad \mathcal{O}_p^\dagger \mathcal{T}_p] - [A \mathcal{O}_p^\dagger \mathcal{T}_p \quad 0_{pm \times r}] \quad (51)$$

Similar to equations (37), with the use of the alternate expression for the  $pm \times pr$  matrix  $\mathcal{T}_p$  (eqs. (36)), one obtains

$$\left. \begin{aligned} \mathcal{O}_p^\dagger \Gamma(:, 1 : r) &= -A \mathcal{O}_p^\dagger(:, 1 : m) D \\ &\quad + B - A \mathcal{O}_p^\dagger(:, m+1 : pm) \mathcal{O}_p(1 : (p-1)m, :) B \\ \mathcal{O}_p^\dagger \Gamma(:, r+1 : 2r) &= [\mathcal{O}_p^\dagger(:, 1 : m) - A \mathcal{O}_p^\dagger(:, m+1 : 2m)] D \\ &\quad + [\mathcal{O}_p^\dagger(:, m+1 : pm) \mathcal{O}_p(:, 1 : (p-1)m) \\ &\quad - A \mathcal{O}_p^\dagger(:, 2m+1 : pm) \mathcal{O}_p(1 : (p-2)m, :)] B \\ \mathcal{O}_p^\dagger \Gamma(:, 2r+1 : 3r) &= [\mathcal{O}_p^\dagger(:, m+1 : 2m) - A \mathcal{O}_p^\dagger(:, 2m+1 : 3m)] D \\ &\quad + [\mathcal{O}_p^\dagger(:, 2m+1 : pm) \mathcal{O}_p(:, 1 : (p-2)m) \\ &\quad - A \mathcal{O}_p^\dagger(:, 3m+1 : pm) \mathcal{O}_p(1 : (p-3)m, :)] B \\ &\quad \vdots \\ \mathcal{O}_p^\dagger \Gamma(:, pr+1 : (p+1)r) &= \mathcal{O}_p^\dagger(:, (p-1)m+1 : pm) D \end{aligned} \right\} \quad (52)$$

Similar to equation (38), equations (52) can be rewritten in the following matrix form:

$$\mathcal{O}_p \Gamma = \mathcal{O}_p A \begin{bmatrix} D \\ B \end{bmatrix} \quad (53)$$

where

$$\mathcal{O}_{p\Gamma} = \begin{bmatrix} \mathcal{O}_p^\dagger \Gamma(:, 1 : r) \\ \mathcal{O}_p^\dagger \Gamma(:, r + 1 : 2r) \\ \mathcal{O}_p^\dagger \Gamma(:, 2r + 1 : 3r) \\ \vdots \\ \mathcal{O}_p^\dagger \Gamma(:, pr + 1 : (p + 1)r) \end{bmatrix}$$

$$\mathcal{O}_{pA} = \begin{bmatrix} -A\mathcal{O}_p^\dagger(:, 1 : m) & [I_n - \\ & A\mathcal{O}_p^\dagger(:, m + 1 : pm)\mathcal{O}_p(1 : (p - 1)m, :)] \\ [\mathcal{O}_p^\dagger(:, 1 : m) - \\ & A\mathcal{O}_p^\dagger(m + 1 : 2m, :)] & [\mathcal{O}_p^\dagger(:, m + 1 : pm)\mathcal{O}_p(:, 1 : (p - 1)m) - \\ & A\mathcal{O}_p^\dagger(:, 2m + 1 : pm)\mathcal{O}_p(1 : (p - 2)m, :)] \\ [\mathcal{O}_p^\dagger(:, m + 1 : 2m) - \\ & A\mathcal{O}_p^\dagger(2m + 1 : 3m, :)] & [\mathcal{O}_p^\dagger(:, 2m + 1 : pm)\mathcal{O}_p(:, 1 : (p - 2)m) - \\ & A\mathcal{O}_p^\dagger(:, 3m + 1 : pm)\mathcal{O}_p(1 : (p - 3)m, :)] \\ \vdots & \vdots \\ \mathcal{O}_p^\dagger(:, (p - 1)m + 1 : pm) & 0_n \end{bmatrix}$$

Here,  $I_n$  is an identity matrix of order  $n$  and  $0_n$  is a zero matrix of order  $n$ . The quantity  $\mathcal{O}_{p\Gamma}$  is a  $pn \times r$  matrix and  $\mathcal{O}_{pA}$  is a  $(p + 1)n \times (m + n)$  matrix. Let the left side of equation (50) be denoted by

$$\mathcal{O}_{p\mathcal{R}} = \mathcal{O}_p^\dagger \tilde{\mathcal{R}}_{yu}(m + 1 : (p + 1)m, :)\tilde{\mathcal{R}}_{uu}^{-1} - A\mathcal{O}_p^\dagger \tilde{\mathcal{R}}_{yu}(1 : pm, :)\tilde{\mathcal{R}}_{uu}^{-1} \quad (54)$$

Equation (50) implies that

$$\mathcal{O}_{p\Gamma} = \begin{bmatrix} \mathcal{O}_{p\mathcal{R}}(:, 1 : r) \\ \mathcal{O}_{p\mathcal{R}}(:, r + 1 : 2r) \\ \mathcal{O}_{p\mathcal{R}}(:, 2r + 1 : 3r) \\ \vdots \\ \mathcal{O}_{p\mathcal{R}}(:, pr + 1 : (p + 1)r) \end{bmatrix} \quad (55)$$

Matrices  $B$  and  $D$  can then be determined from equation (53) by

$$\begin{bmatrix} D \\ B \end{bmatrix} = \mathcal{O}_{pA}^\dagger \mathcal{O}_{p\Gamma} \quad (56)$$

The first  $m$  rows of  $\mathcal{O}_{pA}^\dagger \mathcal{O}_{p\Gamma}$  form the matrix  $D$ , and the last  $n$  rows produce the matrix  $B$ .

Equation (56) has a unique least-squares solution for matrices  $B$  and  $D$  only if the matrix  $\mathcal{O}_{pA}$  has more rows than columns. Since the size of  $\mathcal{O}_{pA}$  is  $pn \times (m + n)$ , the integer  $p$  must be chosen so that  $pn \geq (m + n)$ , where  $n$  is the order of the system. The direct method for computing  $B$  and  $D$  is generally more computationally intensive than the indirect method. For example, when  $n = 10$  and  $p = 20$ , the number of rows in  $\mathcal{O}_{pA}$  becomes  $(p + 1)n = 210$ . However, it is unnecessary to use all the rows of  $\mathcal{O}_{pA}$  to solve for  $B$  and  $D$ . It is sufficient to use the number

of rows in  $\mathcal{O}_{pA}$  and  $\mathcal{O}_{p\Gamma}$  so that the rank is larger than  $m + n$ . It should also be noted that more rows in  $\mathcal{O}_{pA}$  and  $\mathcal{O}_{p\Gamma}$  may improve the solution, particularly when considerable system uncertainties are present.

The indirect and direct methods minimize the equation errors of equations (38) and (53), respectively. This does not imply that the output error between the real output and the reconstructed output is minimized. Here, the reconstructed output is the output time history obtained with the input time history to drive the identified system model represented by the computed matrices  $A, B, C$ , and  $D$ . An alternate method that minimizes the output error is given in section 3.2.2.3.

*3.2.2.3. Output-error minimization method.* The output-error minimization method starts with rearranging the output equation (eq. (5)). The rearrangement is performed to formulate a linear equation that explicitly relates the output vector to the elements of matrices  $B$  and  $D$ . The least-squares solution for matrices  $B$  and  $D$  will then minimize the output error between the real output and the reconstructed output.

Use equation (5) with  $p = N$  and  $k = 0$  to obtain

$$y_N(0) = \mathcal{O}_N x(0) + \mathcal{T}_N u_p(0) \quad (57)$$

The matrices  $B$  and  $D$  are embedded in the term  $\mathcal{T}_N u_p(0)$  on the right side of equation (57). The method for extracting matrices  $B$  and  $D$  from equation (57) will now be shown.

Let the column vectors in  $B$  and  $D$  be expressed as

$$\left. \begin{aligned} B &= [\underline{b}_1 \quad \underline{b}_2 \quad \cdots \quad \underline{b}_r] \\ D &= [\underline{d}_1 \quad \underline{d}_2 \quad \cdots \quad \underline{d}_r] \end{aligned} \right\} \quad (58)$$

Each column vector  $\underline{b}_i$  ( $i = 1, 2, \dots, r$ ) has  $n$  elements, with  $n$  being the length of the state vector, and each column vector  $\underline{d}_i$  ( $i = 1, 2, \dots, r$ ) has  $m$  elements, with  $m$  being the number of outputs. Let the vectors  $\underline{b}$  and  $\underline{d}$  be defined as

$$\left. \begin{aligned} \underline{b} &= \left[ \begin{array}{c} \underline{b}_1 \\ \underline{b}_2 \\ \vdots \\ \underline{b}_r \end{array} \right] \\ \underline{d} &= \left[ \begin{array}{c} \underline{d}_1 \\ \underline{d}_2 \\ \vdots \\ \underline{d}_r \end{array} \right] \end{aligned} \right\} \quad (59)$$

The column vector  $\underline{b}$  is the result of stacking together all the column vectors of the input matrix  $B$  and column vector  $\underline{d}$  includes all the column vectors of the transmission matrix  $D$ . Similarly, let the input vector  $u(k)$  be explicitly written as

$$u(k) = \left[ \begin{array}{c} u_1(k) \\ u_2(k) \\ \vdots \\ u_r(k) \end{array} \right] \quad (60)$$

where the quantities  $u_i(k)$  for  $i = 1, 2, \dots, r$  are scalar, with  $r$  being the number of inputs.

With  $\underline{b}$  and  $\underline{d}$  (eqs. (59)) and  $\mathcal{T}_N$  (eq. (3)),  $\mathcal{T}_N u_p(0)$  may now be rewritten as

$$\begin{aligned} \mathcal{T}_N u_p(0) &= \begin{bmatrix} D & & & & \\ CB & D & & & \\ CAB & CB & D & & \\ \vdots & \vdots & \vdots & \ddots & \\ CA^{N-2}B & CA^{N-3}B & CA^{N-4}B & \cdots & D \end{bmatrix} \begin{bmatrix} u(0) \\ u(1) \\ u(2) \\ \vdots \\ u(N-1) \end{bmatrix} \\ &= \begin{bmatrix} \underline{\mathcal{U}}_m(0) \\ \underline{\mathcal{U}}_m(1) \\ \underline{\mathcal{U}}_m(2) \\ \vdots \\ \underline{\mathcal{U}}_m(N-1) \end{bmatrix} \underline{d} + \begin{bmatrix} 0_{m \times n} \\ C\underline{\mathcal{U}}_n(0) \\ CA\underline{\mathcal{U}}(0) + C\underline{\mathcal{U}}_n(1) \\ \vdots \\ \sum_{k=0}^{N-2} CA^{N-k-2}\underline{\mathcal{U}}_n(k) \end{bmatrix} \underline{b} \end{aligned} \quad (61)$$

where  $0_{m \times n}$  is an  $m \times n$  zero matrix, and

$$\left. \begin{aligned} \underline{\mathcal{U}}_m(k) &= [I_m u_1(k) \quad I_m u_2(k) \quad \cdots \quad I_m u_r(k)] \\ \underline{\mathcal{U}}_n(k) &= [I_n u_1(k) \quad I_n u_2(k) \quad \cdots \quad I_n u_r(k)] \end{aligned} \right\} \quad (62)$$

and  $I_m$  and  $I_n$  are identity matrices of order  $m$  and  $n$ , respectively. Appendix A presents a simple method of computing the summation term in equation (61). The matrix size of  $\underline{\mathcal{U}}_m(k)$  is  $m \times mr$  and  $\underline{\mathcal{U}}_n(k)$  is  $n \times nr$ . The purpose of rewriting the expression for  $\mathcal{T}_N u_p(0)$  is to move the unknown quantities  $B$  and  $D$  outside the brackets (eq. (61)).

Substituting equations (62) into equation (57) yields

$$y_N(0) = \Phi \Theta \quad (63)$$

where

$$\Theta = \begin{bmatrix} x(0) \\ \underline{d} \\ \underline{b} \end{bmatrix} \quad \Phi = \begin{bmatrix} C & \underline{\mathcal{U}}_m(0) & 0_{m \times n} \\ CA & \underline{\mathcal{U}}_m(1) & C\underline{\mathcal{U}}_n(0) \\ CA^2 & \underline{\mathcal{U}}_m(2) & CA\underline{\mathcal{U}}(0) + C\underline{\mathcal{U}}_n(1) \\ \vdots & \vdots & \vdots \\ CA^{N-1} & \underline{\mathcal{U}}_m(N-1) & \sum_{k=0}^{N-2} CA^{N-k-2}\underline{\mathcal{U}}_n(k) \end{bmatrix} \quad (64)$$

The vector size  $\Theta$  is  $(n + mr + nr) \times 1$  and the matrix size  $\Phi$  is  $mN \times (n + mr + nr)$ . The unknown vector  $\Theta$  can then be solved by

$$\Theta = \Phi^\dagger y_N(0) \quad (65)$$

where  $\dagger$  denotes the pseudoinverse. The least-squares solution  $\Theta$  does not actually satisfy equation (63) when the system has input and output noises. However,  $\Theta$  minimizes the error between the actual output vector  $y_N(0)$  and the computed output vector  $\hat{y}_N(0) = \Phi \Theta$ . Solving for the least-squares solution  $\Theta$  can be very time consuming because the number of rows in  $\Phi$  is  $m$  times the integer  $N$  (data length). For example, the row number can be as large as 10 000 for a system with  $m = 5$  outputs and  $N = 2000$  data points.

The SRIM algorithm was developed to compute system matrices  $A$ ,  $B$ ,  $C$ , and  $D$ . This section presented two methods for computing matrices  $A$  and  $C$  and three methods for calculating matrices  $B$  and  $D$ . Computational steps for programming the SRIM algorithm are given in section 3.3.

### 3.3. Computational Steps

To better understand the computational procedure for the SRIM algorithm, the computational steps are summarized as follows:

1. Choose an integer  $p$  so that  $p \geq \frac{n}{m} + 1$ , where  $n$  is the desired order of the system and  $m$  is the number of outputs.
  2. Compute correlation matrices  $\mathcal{R}_{yy}$  of dimension  $pm \times pm$ ,  $\mathcal{R}_{yu}$  of dimension  $pm \times pr$ , and  $\mathcal{R}_{uu}$  of dimension  $pr \times pr$  (eqs. (14)) with the matrices  $Y_p(k)$  of dimension  $pm \times N$  and  $U_p(k)$  of dimension  $pr \times N$  (eqs. (13)). The integer  $r$  is the number of inputs. The index  $k$  is the data point used as the starting point for system identification. The integer  $N$  must be chosen so that  $\ell - k - p + 2 \geq N \gg \min(pm, pr)$ , where  $\ell$  is the length of the data.
  3. Calculate the correlation matrix  $\mathcal{R}_{hh}$  of dimension  $pm \times pm$  (eqs. (14)), that is,  $\mathcal{R}_{hh} = \mathcal{R}_{yy} - \mathcal{R}_{yu}\mathcal{R}_{uu}^{-1}\mathcal{R}_{yu}^T$ .
  4. Factor  $\mathcal{R}_{hh}$  with singular-value decomposition for the full decomposition method (eq. (26)) or a portion of  $\mathcal{R}_{hh}$  for the partial decomposition method (eq. (31)).
  5. Determine the order  $n$  of the system by examining the singular values of  $\mathcal{R}_{hh}$ , and obtain  $\mathcal{U}_n$  of dimension  $pm \times n$  (eq. (26)) and  $\mathcal{U}_o$  of dimension  $pm \times n_o$ , where  $n_o = pm - n$  is the number of truncated small singular values. The integer  $n_o$  must satisfy the condition  $pn_o \geq (m+n)$  for the full decomposition method. For the partial decomposition method,  $\mathcal{U}_n$  is replaced by  $\mathcal{U}'_n$  (eq. (31)) and the integer  $n_o$  is the sum of  $m$  and the number of truncated singular values.
  6. Let  $\mathcal{U}_n = \mathcal{O}_p$  or  $\mathcal{U}'_n = \mathcal{O}_p$ . Use equation (8) to determine the state matrix  $A$ . The output matrix  $C$  is the first  $m$  rows of  $\mathcal{U}_n$ .
  7. Compute  $\mathcal{U}_o\mathcal{R} = \mathcal{U}_o^T\mathcal{R}_{yu}\mathcal{R}_{uu}^{-1}$  (eq. (39)) for the indirect method, and construct  $\mathcal{U}_{on}$  and  $\mathcal{U}_{oT}$  (eqs. (38) and (40)). Determine the input matrix  $B$  and the direct transmission matrix  $D$  from equation (41) (i.e., the first  $m$  rows of  $\mathcal{U}_{on}^{\dagger}\mathcal{U}_{oT}$  form  $D$  and the last  $n$  rows produce  $B$ ).  
For the direct method, construct  $\mathcal{O}_{p\Gamma}$  and  $\mathcal{O}_{pA}$  from equation (53) and solve for matrices  $B$  and  $D$  by computing  $\mathcal{O}_{pA}^{\dagger}\mathcal{O}_{p\Gamma}$ . The first  $m$  rows of  $\mathcal{O}_{pA}^{\dagger}\mathcal{O}_{p\Gamma}$  form matrix  $D$ , and the last  $n$  rows produce matrix  $B$ .
- For the output-error minimization method, construct  $y_N(0)$  and  $\Phi$  from equations (64) and solve for matrices  $B$  and  $D$  by computing  $\Phi^{\dagger}y_N(0)$ . The first  $n$  elements of  $\Phi^{\dagger}y_N(0)$  form the initial state vector  $x(0)$ , the second  $mr$  elements give the  $r$  column vectors of  $D$ , and the last  $nr$  elements produce the  $r$  column vectors of  $B$ .
8. Find the eigenvalues and eigenvectors of the realized state matrix and transform the realized model into modal coordinates to compute system damping and frequencies. This step is needed only if modal parameters identification is desired.
  9. Calculate mode singular values (ref. 1) to quantify and distinguish the system and noise modes. This step provides a way for model reduction with modal truncation.

The computational steps reduce to the steps for the ERA/DC method (ref. 1) when the output data are the pulse-response-time history. Assume that a pulse is given to excite the system at the time step zero. Let  $k = 1$  in step 2. The correlation matrices  $\mathcal{R}_{yu}$  and  $\mathcal{R}_{uu}$  become null, and  $\mathcal{R}_{hh} = \mathcal{R}_{yy}$  is obtained. Theoretically, the formulation  $\mathcal{R}_{hh} = \mathcal{R}_{yy} - \mathcal{R}_{yu}\mathcal{R}_{uu}^{-1}\mathcal{R}_{yu}^T$  should not be used for computation of  $\mathcal{R}_{hh}$  if  $\mathcal{R}_{uu}$  is not invertible. For special cases such as free decay and pulse responses,  $\mathcal{R}_{hh}$  reduces to  $\mathcal{R}_{yy}$  when the integer  $k$  is chosen at the point where the input signal vanishes.

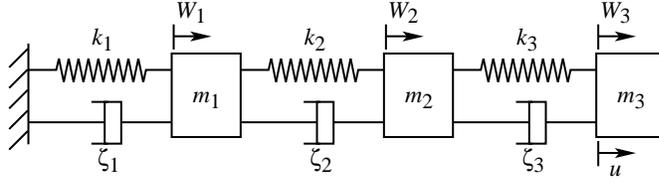


Figure 1. Mass-spring-dashpot system

The SRIM algorithm with the computational steps incorporated with the recursive formula (appendix B) is more efficient computationally than subspace model identification (SMI) techniques (refs. 6–8). The SMI techniques require a  $QR$  (refs. 8 and 9) factorization of a large matrix  $[U_p^T(k) Y_p^T(k)]^T$ , followed by a singular-value decomposition and the solution of an overdetermined set of equations. Furthermore, the proposed method with the concept of data correlation permits more physical insight than the SMI techniques.

### 3.4. Examples

To illustrate the SRIM algorithm, a numerical example and an experimental example are given. The numerical example uses a three-degree-of-freedom mass-spring-dashpot system. The experimental example uses a truss structure tested at Langley Research Center. Different methods of computing system matrices are compared and discussed relating to system frequencies and dampings.

#### 3.4.1. Numerical Example

Figure 1 shows the mass-spring-dashpot system where  $m_i$ ,  $\zeta_i$ , and  $k_i$  ( $i = 1, 2, 3$ ) are masses, damping coefficients, and spring constants, respectively  $w_i(t)$  are absolute displacements for the respective masses, and  $u_i$  are input forces. The second-order differential equation for this system is

$$M\ddot{w} + \Xi\dot{w} + Kw = bu$$

where

$$M = \begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix} \quad \Xi = \begin{bmatrix} \zeta_1 + \zeta_2 & -\zeta_2 & 0 \\ -\zeta_2 & \zeta_2 + \zeta_3 & -\zeta_3 \\ 0 & -\zeta_3 & \zeta_3 \end{bmatrix}$$

$$K = \begin{bmatrix} k_1 + k_2 & -k_2 & 0 \\ -k_2 & k_2 + k_3 & -k_3 \\ 0 & -k_3 & k_2 + k_3 \end{bmatrix} \quad w = \begin{Bmatrix} w_1 \\ w_2 \\ w_3 \end{Bmatrix} \quad b = \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix}$$

The corresponding first-order differential equation is

$$\dot{x} = A_c x + B_c u$$

where

$$x = \begin{bmatrix} w \\ \dot{w} \end{bmatrix} \quad A_c = \begin{bmatrix} 0_3 & I_3 \\ -M^{-1}K & -M^{-1}\Xi \end{bmatrix} \quad B_c = \begin{bmatrix} 0_{3 \times 1} \\ b \end{bmatrix}$$

with  $0_3$  being a zero matrix of order 3,  $I_3$  an identity matrix of order 3, and  $0_{3 \times 1}$  a  $3 \times 1$  zero vector. For simplicity, let the mass  $m_1 = m_2 = m_3 = 1$ ,  $k_1 = 1, k_2 = 2, k_3 = 3$ , and  $\Xi = 0.05\sqrt{K}$ . The damping matrix  $\Xi = 0.05\sqrt{K}$  is chosen so that each mode has a proportional damping of 0.5 percent. With a sampling rate of 1 Hz, the system is excited by a discretized random force  $u(k)$  for  $k = 1, 2, \dots, 3000$  (normally distributed with signal of unit strength). Two output signals are obtained from the accelerometers located at positions 1 and 2 as follows:

$$y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \ddot{w}_1 \\ \ddot{w}_2 \\ \ddot{w}_3 \end{bmatrix} = C_a \ddot{w}$$

where

$$C_a = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

The measurement vector  $y$  after substitution of  $\ddot{w}$  becomes

$$y = C_a M^{-1}(bu - \Xi \dot{w} - Kw) = Cx + Du$$

where

$$C = -C_a M^{-1}[K \quad \Xi] \quad \text{and} \quad D = C_a b$$

The discrete state-space model is

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

where

$$\begin{aligned} A &= 10^{-2} \times \begin{bmatrix} -4.0315 & 47.981 & 17.092 & 59.137 & 22.195 & 3.9627 \\ 47.981 & -26.375 & 71.972 & 22.195 & 42.886 & 33.292 \\ 17.092 & 71.972 & 10.211 & 3.9627 & 33.292 & 62.440 \\ -133.02 & 19.188 & 54.697 & -4.8427 & 47.944 & 17.343 \\ 19.188 & -70.165 & 28.782 & 47.944 & -26.773 & 71.916 \\ 54.697 & 28.782 & -87.442 & 17.343 & 71.916 & 9.6095 \end{bmatrix} \\ B &= 10^{-2} \times \begin{bmatrix} 0.72516 \\ 9.6335 \\ 39.563 \\ 3.9627 \\ 33.292 \\ 62.440 \end{bmatrix} \\ C &= 10^{-2} \begin{bmatrix} -300 & 200 & 0 & -1.6119 & 0.60778 & 0.18028 \\ 200 & -500 & 300 & 0.60778 & -1.9492 & 0.91167 \end{bmatrix} \\ D &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{aligned}$$

and  $k$  is the time index. The quantities  $\nu(k)$  and  $\varepsilon(k)$  are added to the model to represent the process noise and the measurement noise, respectively. In practice, both noises are generally assumed random-normally distributed. The process noise is set at approximately 10 percent of the input force, and the measurement noise is set at about 10 percent of the output force, both as standard deviation ratios.

The initial values for  $p$  are arbitrarily selected as  $p = 6, 12, 25, 50$ , and 100 to make the maximum system order  $pm = 12, 24, 100$ , and 200, which is higher than the anticipated system

order of  $n = 6$ . For illustration, the partial decomposition method is used to compute the state matrix  $A$  and the output matrix  $C$ . The indirect method and the output-error minimization method are used to compute the input matrix  $B$  and the direct-transmission matrix  $D$ .

In practical applications, the correct system order is unknown and the maximum system order selected by  $pm$  provides an upper bound, allowing the solution approach to proceed. From these upper bounds, there are two ways to reduce the model to the order of 6. One way is to truncate the singular values of  $\mathcal{R}_{hh}$  and retain only the six largest values, resulting in system matrices  $A$  of  $6 \times 6$ ,  $B$  of  $6 \times 1$ ,  $C$  of  $2 \times 6$ , and  $D$  of  $2 \times 1$ . The eigenvalues of system matrix  $A$  lead to the frequencies and dampings, and the matrices  $C$  and  $B$  can be used to estimate the mode shapes and modal participation factors (ref. 1).

Table 1 lists the true modal frequencies and damping ratios. Table 2 shows the identified damping ratios for different values of  $p$ . The identified frequencies are not shown because they are identical to the true frequencies up to three digits. The last column, "Error max SV," gives the largest singular value of the error matrix between the real output and the output reconstructed from the identified system matrices with the same input signal. The error matrix has the size of  $m \times \ell$ , where  $m$  is the number of outputs and  $\ell$  is the length of the data.

Another way to reduce the model order is to obtain the full-size system matrices first (i.e., no singular-values truncation) and then perform the modal truncation. Each identified mode is weighted by its mode-singular value (ref. 1, pp. 139–143). The three modes with largest mode-singular values are then chosen to represent the system. Table 3 shows the identified damping ratios for different values of  $p$ . The identified frequencies are identical to the true frequencies up to three digits. (See table 1.)

Table 1. True Modal Parameters

Mode	True damping ratio, percent	True frequency, Hz
1	0.50	0.80
2	.50	.28
3	.50	.44

Table 2. Identified Damping Ratio Obtained by Partial Decomposition Method With Singular Values Truncation

$p$	Mode 1 damping, percent	Mode 2 damping, percent	Mode 3 damping, percent	IDM error, max SV	OEM error, max SV
6	3.06	0.93	1.01	180.97	167.5
12	1.13	.53	.50	136.60	129.77
25	.55	.43	.45	128.28	126.23
50	.40	.41	.43	129.99	127.04
100	.37	.41	.37	132.00	128.01

Table 3. Identified Damping Ratio Obtained by Partial Decomposition Method With Modal Truncation

$p$	Mode 1 damping, percent	Mode 2 damping, percent	Mode 3 damping, percent	Error, max SV
6	3.80	1.34	1.12	194.95
12	1.65	.72	.60	152.73
25	.66	.50	.54	126.79
50	.46	.47	.50	126.18
100	.42	.65	.39	167.35

Table 2 shows that all damping ratios are underestimated by singular-values truncation of  $\mathcal{R}_{hh}$  when the integer  $p$  is chosen sufficiently large compared with the minimum requirement of  $p$ , which is 3 in this example. An integer  $p$  exists that produces a minimal output error. Increasing the value of  $p$  beyond the optimal value does not necessarily reduce the output error. Both the indirect method and the output-error minimization method show similar trends on the output error. The output-error minimization method gives better output-error solutions than the indirect method, but takes three orders of magnitude longer for computation when compared with the indirect method.

On the other hand, table 3 shows that some damping ratios may still be overestimated by modal truncation for a large  $p$ . Also, there exists an integer  $p$  that produces a minimal output error. Comparing tables 2 and 3 indicates that the modal-truncation method performs somewhat better at  $p = 50$  than both the output-error minimization method and the indirect method. The computational time for the modal-truncation method is about the same as the indirect method, but much less than the output-error minimization method. It should be noted that the output-error minimization method only minimizes the output error relative to  $\mathbf{x}(0)$  and matrices  $B$  and  $D$  (given matrices  $A$  and  $C$ ), and therefore, does not globally minimize the output error relative to all matrices  $A$ ,  $B$ ,  $C$ ,  $D$ , and  $\mathbf{x}(0)$ .

### 3.4.2. Experimental Example

The experimental results given in this section illustrate the usefulness of the proposed techniques when used in practice. Figure 2 shows the L-shaped truss structure used. This structure consists of nine bays on its vertical section and one bay on its horizontal section which extend 90 in. and 20 in., respectively. The shorter section is clamped to a steel plate that is rigidly attached to the wall. The square cross section is 10 in. by 10 in. Two cold air jet thrusters located at the beam tip serve as actuators for excitation and control. Each thruster has a maximum thrust of 2.2 lb. Two servoaccelerometers located at a corner of the square cross section provide the in-plane tip acceleration measurements. In addition, a ~~offset~~ weight of 30 lb is added to enhance dynamic coupling between the two principal axes and to lower the structure fundamental frequency. For identification, the truss is excited with random inputs to both thrusters. The input-output signals are sampled at 250 Hz and recorded for system identification. A data record of 2000 points is used for identification.

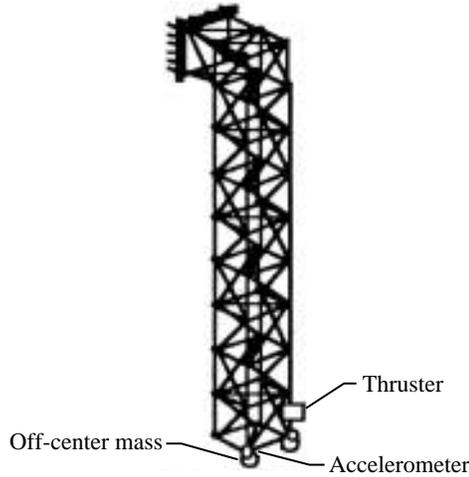


Figure 2. Truss structure test configuration.

Table 4 lists the modal frequencies and damping ratios identified with the partial decomposition method for determining matrices  $A$  and  $C$  in conjunction with the output-error minimization method for computing matrices  $B$  and  $D$ . The initial index  $p$  is arbitrarily set to make the maximum system order  $pm = 10, 20, 30, 40, 50, 100$ , and 200. The singular-values truncation is used to reduce the order of the system model to 6. The output error decreases continuously as  $p$  increases from 5 to 100. The speed of decreasing the output error is slow from  $p = 50$  to  $p = 100$ . The frequencies identified for all different  $p$  values are close and the damping ratios range from 2.5 percent to 0.4 percent for the first mode, from 1.5 percent to 0.4 percent for the second mode, and from 1.2 percent to 0.07 percent for the third mode.

Results from the indirect and direct methods for computing matrices  $B$  and  $D$  are not shown because these methods produce output errors several orders of magnitude higher than the output errors shown in table 4. Both methods work well for the simulation data, with input and output noises assumed to be white, random, Gaussian, and zero-mean. Therefore, it is believed that noise nonlinearities are the major causes of the problem of introducing significant errors in matrices  $B$  and  $D$ . This example shows that the indirect and direct methods should not be used in practice for computing matrices  $B$  and  $D$  if matrices  $A$  and  $C$  are obtained by the reduced model with singular-values truncation.

Table 4. Identified Modal Parameters Obtained by Partial Decomposition Method With Singular-Values Truncation

$p$	Mode 1		Mode 2		Mode 3		OEM error, max SV
	Frequency, Hz	Damping, percent	Frequency, Hz	Damping, percent	Frequency, Hz	Damping, percent	
5	5.89	2.48	7.29	1.54	48.5	1.19	621.77
10	5.88	2.08	7.30	.99	48.2	.71	545.58
15	5.88	1.10	7.30	.51	48.0	.99	365.04
20	5.88	.62	7.29	.38	47.5	2.06	264.57
25	5.87	.46	7.29	.42	47.4	2.44	207.63
50	5.86	.44	7.29	.41	48.4	.46	174.25
100	5.85	.43	7.28	.43	48.6	.07	167.64

Table 5. Identified Modal Parameters Obtained by Partial Decomposition Method With Modal Truncation

$p$	Mode 1		Mode 2		Mode 3		IDM error, max SV	OEM error, max SV
	Frequency, Hz	Damping, percent	Frequency, Hz	Damping, percent	Frequency, Hz	Damping, percent		
5	5.89	3.50	7.28	2.30	49.0	1.13	817.99	735.99
10	5.87	.65	7.29	.47	48.6	.74	262.81	202.84
15	5.85	.40	7.28	.41	48.6	.46	197.32	171.33
20	5.85	.37	7.28	.41	48.7	.44	216.79	174.46
25	5.85	.38	7.28	.42	48.7	.64	203.56	174.06
50	5.85	.38	7.28	.44	48.6	.47	198.69	175.11
100	5.85	.40	7.28	.45	48.5	.30	194.58	174.02

Table 5 lists the modal frequencies and damping ratios identified with the partial decomposition method for determining matrices  $A$  and  $C$  combined with the indirect method for computing matrices  $B$  and  $D$  without singular-values truncation. The full-size model is then reduced to the order of 6 (including only those modes of interest) and used to compute the output error. The output error decreases quickly when  $p$  increases from 5 to 10, and reaches a minimum at  $p = 15$ . The output error increases slightly again and then reduces to another minimum at  $p = 100$ . However, the minimum output-error value of 194.58 at  $p = 100$  improves little from the minimum output-error value of 197.32 at  $p = 15$ . Similar to the results shown in table 4, the frequencies identified for all different  $p$  values are very close and the damping ratios range from 3.5 percent to 0.4 percent for the first mode, from 2.3 percent to 0.45 percent for the second mode, and from 1.13 percent to 0.3 percent for the third mode.

The similarity of modal parameters in tables 4 and 5 is not surprising because both tables share the same observability matrix before singular-values truncation. Table 4 shows the modal parameters computed after singular-values truncation, i.e., some columns of observability matrix corresponding to small singular values are truncated. On the other hand, table 5 shows the modal parameters computed from the full-size observability matrix. The modal parameters shown in table 5 are chosen to represent the system.

The question may arise whether the output errors in table 5 may be reduced if matrices  $B$  and  $D$  are recalculated with the output-error minimization method with the same matrices  $A$  and  $C$ . The last column of table 5 provides an answer to this question. Indeed, the output errors are somewhat improved for all cases. The output-error value of 171.33 for  $p = 15$  in table 5 is better than the output-error value of 174.25 in table 4 at  $p = 50$ . This result indicates that the combination of singular-values truncation combined with output-error minimization may not produce the global minimum for any given  $p$  value. As a result, modal truncation combined with the output-error minimization method works well for model reduction.

### 3.5. Section 3 Summary

A new system realization algorithm is developed with a data correlation matrix to compute an observability matrix with singular-value decomposition. The data correlation matrix is formed by the autocorrelation matrix of the shifted output data subtracted from cross correlation between shifted input and output data weighted by the inverse of the autocorrelation matrix of the shifted input data. The observability matrix is then used to compute the state matrix  $A$  and the output matrix  $C$ .

Two computational methods are presented, including a full decomposition method and a partial decomposition method, to determine matrices  $A$  and  $C$ . The partial decomposition method seems easier to use than the full decomposition matrix because the partial decomposition method eliminates the need for singular-values truncation. In practice, there are no zero singular values regardless of how clean the data sequence is. Determining the number of singular values to truncate requires engineering judgment or use of special techniques such as sensitivity analysis.

Based on the computed matrices  $A$  and  $C$ , three methods are described for computing the input matrix  $B$ , the direct transmission matrix  $D$ , and the initial state vector  $x(0)$ . The indirect method uses the matrix with column vectors orthogonal to the observability matrix in conjunction with a data-correlation equation to extract matrices  $B$  and  $D$ . The direct method uses the observability matrix and the output equation to formulate an equation to solve for matrices  $B$  and  $D$ . The output-error minimization method determines matrices  $B$  and  $D$  and initial state vector  $x(0)$  by minimizing the error between the test output and the computed (i.e., reconstructed) output. When the input and output noises are white, Gaussian, and zero-mean, any combination of the methods mentioned in this summary for computing matrices  $A$ ,  $B$ ,  $C$ ,  $D$ , and initial state vector  $x(0)$  performs well. For other noises, any combination works well if no singular-values truncation is conducted. With singular-values truncation for model reduction, the combination of the partial decomposition algorithm and output-error minimization works better than the other methods and is comparable to the modal truncation technique.

#### 4. Unification of SRIM and Other Methods

Many system realization algorithms start with a state-space, discrete-time linear model and then formulate fundamental equations based on data correlation to compute system matrices. On the other hand, other algorithms use the finite-difference model and data correlation to solve for the system matrices. The two approaches appear fundamentally different because they use different types of models. The different models may yield identification results with noticeable discrepancy because of the different equation errors that are minimized. As a result, it is very difficult to interpret identification results and choose the technique best suited for a problem. The need exists to provide a comprehensive, yet coherent, unification of the different techniques.

Section 4 of this paper establishes the relationship among the SRIM presented in section 3 and other realization methods. The SRIM is derived with the state-space, discrete-time linear equation to form a special type of data correlation for system realization. Other realization algorithms, such as the observer/Kalman filter identification (OKID) technique (refs. 11 and 12), start by computing the coefficient matrices of the finite-difference equation, which also requires information from input- and output-data correlation. The similarity of using data correlation leads to establishing the relationship between the SRIM and the OKID method. The approach presented in this section provides a better way to understand and interpret other techniques, such as OKID and the subspace identification approach developed by other researchers (refs. 6 and 7).

##### 4.1. Time-Domain ARX Model

The finite difference model is commonly called the autoregressive exogeneous (ARX) model by the controls community. The ARX coefficient matrices can be computed from input and output data by minimizing the output-equation error (ref. 11), that is, the error between the actual output and the estimated output.

The discrete-time ARX model is typically written as

$$\begin{aligned} & \alpha_{p-1}y(k+p-1) + \alpha_{p-2}y(k+p-2) + \cdots + \alpha_0y(k) \\ & = \beta_{p-1}u(k+p-1) + \beta_{p-2}u(k+p-2) + \cdots + \beta_0u(k) \end{aligned} \tag{66}$$

where  $\alpha_i$  for  $i = 1, 2, \dots, p-1$  is an  $m \times m$  matrix,  $\beta_i$  for  $i = 1, 2, \dots, p-1$  is an  $m \times r$  matrix,  $y(k)$  is an  $m \times 1$  output vector at the time step  $k$ , and  $u(k)$  is an  $r \times 1$  input vector at the time step  $k$ . This equation relates the output sequence  $y(k)$  to the input sequence  $u(k)$  up to  $p$  time steps.

Equation (66) produces the following matrix equality:

$$\alpha Y_p(k) = \beta U_p(k) \quad (67)$$

where

$$\left. \begin{aligned} \alpha &= [\alpha_0 \quad \alpha_1 \quad \cdots \quad \alpha_{p-1}] \\ \beta &= [\beta_0 \quad \beta_1 \quad \cdots \quad \beta_{p-1}] \\ Y_p(k) &= \begin{bmatrix} y(k) & y(k+1) & \cdots & y(k+N-1) \\ y(k+1) & y(k+2) & \cdots & y(k+N) \\ \vdots & \vdots & \ddots & \vdots \\ y(k+p-1) & y(k+p) & \cdots & y(k+p+N-2) \end{bmatrix} \\ U_p(k) &= \begin{bmatrix} u(k) & u(k+1) & \cdots & u(k+N-1) \\ u(k+1) & u(k+2) & \cdots & u(k+N) \\ \vdots & \vdots & \ddots & \vdots \\ u(k+p-1) & u(k+p) & \cdots & u(k+p+N-2) \end{bmatrix} \end{aligned} \right\} \quad (68)$$

Postmultiplying equation (67) by  $U_p^T(k)$ , and noting equations (14) for definitions of  $\mathcal{R}_{yu}$  and  $\mathcal{R}_{uu}$ , yield

$$\alpha \mathcal{R}_{yu} = \beta \mathcal{R}_{uu} \quad \implies \quad \beta = \alpha \mathcal{R}_{yu} \mathcal{R}_{uu}^{-1} \quad (69)$$

Here, the existence of  $\mathcal{R}_{uu}^{-1}$  is assumed. Similarly, postmultiplying equation (67) by  $Y_p^T(k)$ , and noting equations (14) for definitions of  $\mathcal{R}_{yy}$  and  $\mathcal{R}_{yu}$ , yield

$$\alpha \mathcal{R}_{yy} = \beta \mathcal{R}_{yu}^T \quad (70)$$

Substituting equation (69) for  $\beta$  into equation (70) thus gives

$$\alpha \left[ \mathcal{R}_{yy} - \mathcal{R}_{yu} \mathcal{R}_{uu}^{-1} \mathcal{R}_{yu}^T \right] = 0$$

or

$$\alpha \mathcal{R}_{hh} = 0 \quad (71)$$

where

$$\mathcal{R}_{hh} = \mathcal{R}_{yy} - \mathcal{R}_{yu} \mathcal{R}_{uu}^{-1} \mathcal{R}_{yu}^T$$

which is identical to equation (20). Equation (71) implies that the  $m \times pm$  parameter matrix  $\alpha$  is in the null column space of the  $pm \times pm$  matrix  $\mathcal{R}_{hh}$ . In other words, any  $m$  columns of  $\mathcal{U}_o$  from equation (26) or  $\mathcal{U}'_o$  from equation (31) may be used to construct the matrix  $\alpha$  as follows:

$$\alpha = \text{any } m \text{ rows of } \mathcal{U}_o^T \text{ or } \mathcal{U}'_o{}^T \quad (72)$$

As a result, the  $m \times pr$  matrix  $\beta$  can be solved by

$$\beta = \text{any } m \text{ rows of } \mathbf{U}_o^T \mathcal{R}_{yu} \mathcal{R}_{uu}^{-1} \text{ or } \mathbf{U}_o'^T \mathcal{R}_{yu} \mathcal{R}_{uu}^{-1} \quad (73)$$

In practice, no zero singular value may exist to form the matrix  $\mathbf{U}_o$  because of system uncertainties, including input and output noises. The  $m$  columns corresponding to the  $m$  smallest singular values may then be selected to form  $\mathbf{U}_o$ .

Another way of computing  $\alpha$  and  $\beta$  is to combine equations (69) and (70) to yield

$$[-\alpha \quad \beta] \begin{bmatrix} \mathcal{R}_{yy} & \mathcal{R}_{yu} \\ \mathcal{R}_{yu}^T & \mathcal{R}_{uu} \end{bmatrix} = 0_{m \times p(m+r)} \quad (74)$$

or

$$\Theta \mathcal{R} = 0_{m \times p(m+r)} \quad (75)$$

where  $0_{m \times p(m+r)}$  is an  $m \times p(m+r)$  zero matrix and

$$\Theta = [-\alpha \quad \beta] \quad \text{and} \quad \mathcal{R} = \begin{bmatrix} \mathcal{R}_{yy} & \mathcal{R}_{yu} \\ \mathcal{R}_{yu}^T & \mathcal{R}_{uu} \end{bmatrix} \quad (76)$$

The size of matrix  $\Theta$  is  $m \times p(m+r)$  and the size of  $\mathcal{R}$  is  $p(m+r) \times p(m+r)$ . Equation (75) implies that the unknown matrix  $\Theta$  exists in the null column space of the matrix  $\mathcal{R}$ . Any  $m$  columns generated from the basis vectors of the null column space of  $\mathcal{R}$  may be considered as the solution for  $\Theta^T$ . In theory, the integer  $p$  must be chosen large enough so that the  $p(m+r) \times p(m+r)$  matrix  $\mathcal{R}$  is rank deficient. Because the  $pr \times pr$  input correlation matrix  $\mathcal{R}_{uu}$  is required to have the full rank of  $pr$ , the rank deficiency of  $\mathcal{R}$  implies that the rank of  $\mathcal{R}_{yy}$  must be less than  $pm$ . Specifically, the rank of  $\mathcal{R}_{yy}$  cannot be more than  $pm - m$  to have at least  $m$  independent columns to generate the null column space of  $\mathcal{R}$  of dimension  $m$ .

#### 4.1.1. Observer/Kalman Filter Identification (OKID) Algorithm

One solution for equation (75) can be derived by premultiplying equation (75) by the  $m \times m$  matrix  $\alpha_{p-1}^{-1}$  to obtain

$$\alpha_{p-1}^{-1} \Theta \mathcal{R} = 0_{m \times p(m+r)} \quad (77)$$

or, from equations (68) and (76)

$$[-\alpha_{p-1}^{-1} \alpha_0 \quad -\alpha_{p-1}^{-1} \alpha_1 \quad \cdots \quad -I_m \quad \alpha_{p-1}^{-1} \beta_0 \quad \alpha_{p-1}^{-1} \beta_1 \quad \cdots \quad \alpha_{p-1}^{-1} \beta_{p-1}] \mathcal{R} = 0_{m \times p(m+r)}$$

Now, take out the  $m$  rows of  $\mathcal{R}$  from  $(p-1)m+1$  to  $pm$  and move them to the right side of equation (77) as follows:

$$\tilde{\Theta} \begin{bmatrix} \mathcal{R}(1 : (p-1)m, :) \\ \mathcal{R}((p-1)m+1 : pm, :) \end{bmatrix} = \mathcal{R}((p-1)m+1 : pm, :) \quad (78)$$

where

$$\tilde{\Theta} = [-\alpha_{p-1}^{-1} \alpha_0 \quad -\alpha_{p-1}^{-1} \alpha_1 \quad \cdots \quad -\alpha_{p-1}^{-1} \alpha_{p-2} \quad \alpha_{p-1}^{-1} \beta_0 \quad \alpha_{p-1}^{-1} \beta_1 \quad \cdots \quad \alpha_{p-1}^{-1} \beta_{p-1}] \quad (79)$$

Note that the dimension of  $\tilde{\Theta}$  is  $m \times ((p-1)m + pr)$ , which is  $m$  columns shorter than  $\Theta$  because of removal of the identity matrix  $I_m$ . Equation (78) should have the least-squares solution for  $\tilde{\Theta}$  as follows:

$$\tilde{\Theta} = \mathcal{R}((p-1)m+1 : pm, :) \begin{bmatrix} \mathcal{R}(1 : (p-1)m, :) \\ \mathcal{R}(pm+1 : (p+r)m, :) \end{bmatrix}^\dagger \quad (80)$$

where  $\dagger$  means pseudoinverse. To avoid the pseudoinverse, delete  $m$  columns of  $\mathcal{R}((p-1)m+1 : pm, :)$  from  $(p-1)m+1$  to  $pm$  and corresponding  $m$  columns of  $\begin{bmatrix} \mathcal{R}(1 : (p-1)m, :) \\ \mathcal{R}(pm+1 : (p+r)m, :) \end{bmatrix}$ .

The parameter matrix  $\tilde{\Theta}$  obtained from equation (80) is identical to the parameter matrix used in the OKID technique (refs. 1 and 7) to identify system matrices  $A$ ,  $B$ ,  $C$ ,  $D$ , and an observer gain  $G$ . The correlation matrix  $\mathcal{R}$  is referred to as the information matrix and includes the correlation information between input and output data.

In theory, all methods which start with the same correlation matrix  $\mathcal{R}$  should produce the same identification results. In practice, the identification results may be somewhat different because of the presence of system uncertainties. For example, the matrix  $\tilde{\Theta}$  solved by equation (80) minimizes the output residual between the measured output and the ARX computed output (refs. 1 and 7). On the other hand, other methods minimize the error between the measured output and the output computed from an identified state-space model. Because different error criteria are used for minimization, identification results are expected to be somewhat different. Nevertheless, the results should not be significantly different unless the identified models are considerably reduced by either singular-values truncation and/or modal truncation.

#### 4.1.2. Experimental Example

The data taken from the truss structure (fig. 2) are used in this section for illustration. The OKID method is applied to determine the system matrices  $A$ ,  $B$ ,  $C$ , and  $D$ . Two computational steps are required. The first step is to use equation (80) to compute ARX coefficient matrices to determine system Markov parameters (pulse response). The second step is to use the eigensystem realization algorithm (ERA) from the computed pulse response to realize matrices  $A$ ,  $B$ ,  $C$ , and  $D$  simultaneously. In this example, no singular-values truncation is performed in ERA. The ERA-identified full-size model is then reduced to the order of 6, including only the modes of interest. The reduced model is then used to compute the output error.

Table 6 shows the modal frequencies and damping ratios identified with the OKID method. The output error decreases quickly when  $p$  increases from 5 to 10, and reaches a minimum at  $p = 15$ . The output error increases slightly again, and then reduces to another minimum at  $p = 50$ . Tables 5 and 6 have identical modal parameters. The output errors in both tables are very close except at  $p = 100$ , where the output-error value of 882.26 in table 6 is 4.5 times larger than the output-error value of 194.58 in table 5. Since the modal parameters are identical in both tables, the error from matrices  $C$ ,  $B$ , and  $D$  may be the cause of the discrepancy. Indeed, when matrices  $B$  and  $D$  are recomputed by the output-error minimization method, the output error returns to the same level in both tables.

Table 6. OKID-Identified Modal Parameters  
With Modal Truncation

$p$	Mode 1		Mode 2		Mode 3		OKID error, max SV	OEM error, max SV
	Frequency, Hz	Damping, percent	Frequency, Hz	Damping, percent	Frequency, Hz	Damping, percent		
5	5.89	3.50	7.28	2.30	49.0	1.13	818.05	735.99
10	5.87	.65	7.29	.47	48.6	.74	262.81	202.84
15	5.85	.40	7.28	.41	48.6	.46	197.22	171.33
20	5.85	.37	7.28	.41	48.7	.44	215.97	174.46
25	5.85	.38	7.28	.42	48.7	.64	203.21	174.06
50	5.85	.38	7.28	.44	48.6	.47	196.70	175.10
100	5.85	.40	7.28	.45	48.5	.30	882.26	174.02

#### 4.2. Frequency-Domain ARX Model

The frequency-domain ARX model is produced by taking the  $z$ -transform of the time-domain ARX model. The coefficient matrices of the frequency-domain ARX model can be obtained from the frequency-response data by minimizing the error between the real transfer function and the estimated transfer function at frequency points of interest. In theory, the ARX-coefficient matrices obtained from the frequency-domain approach should be identical to the matrices obtained from the time-domain approach.

Let  $G(z_k)$  be the transfer-function matrix of the system described by equations (1). Consider the left-matrix fraction (ref. 1)

$$G(z_k) = \alpha^{-1}(z_k)\beta(z_k) \quad (81)$$

where

$$\left. \begin{aligned} \alpha(z_k) &= \alpha_0 + \alpha_1 z_k^{-1} + \cdots + \alpha_{p-1} z_k^{-(p-1)} \\ \beta(z_k) &= \beta_0 + \beta_1 z_k^{-1} + \cdots + \beta_{p-1} z_k^{-(p-1)} \end{aligned} \right\} \quad (82)$$

are matrix polynomials. Every  $\alpha_i$  is an  $m \times m$  real square matrix, and each  $\beta_i$  is an  $m \times r$  real rectangular matrix. If  $G(z_k)$  represents the frequency-response function (FRF) obtained from experiments, the variable  $z_k = e^{j(2\pi k/\ell\Delta t)}$  ( $k = 0, 1, \dots, \ell - 1$ ) corresponds to the frequency points at  $2\pi k/\ell\Delta t$ , with  $\Delta t$  being the sampling time interval and  $\ell$  the length of data. The factorization in equation (81) is not unique. For convenience and simplicity, the orders of both polynomials can be chosen as  $p - 1$ .

Premultiplying equation (81) by  $\alpha(z_k)$  produces

$$\alpha(z_k)G(z_k) = \beta(z_k) \quad (83)$$

which can be rearranged into

$$\alpha_0 G(z_k) + \alpha_1 G(z_k) z_k^{-1} + \cdots + \alpha_{p-1} G(z_k) z_k^{-(p-1)} = \beta_0 + \beta_1 z_k^{-1} + \cdots + \beta_{p-1} z_k^{-(p-1)} \quad (84)$$

Equation (84) is the  $z$ -transform of equation (66). With  $G(z_k)$  and  $z_k^{-1}$  known, equation (84) is a linear equation. Because  $G(z_k)$  is known at  $z_k = e^{j(2\pi k/\ell\Delta t)}$  ( $k = 0, \dots, \ell - 1$ ), there are  $\ell$  equations available. Stacking up the  $\ell$  equations yields

$$\Theta\Phi = 0_{m \times \ell} \quad (85)$$

where

$$\Phi = \left[ \begin{array}{cccc} G(z_0) & G(z_1) & \dots & G(z_{\ell-1}) \\ G(z_0)z_0^{-1} & G(z_1)z_1^{-1} & \dots & G(z_{\ell-1})z_{\ell-1}^{-1} \\ \vdots & \vdots & \ddots & \vdots \\ G(z_0)z_0^{-(p-1)} & G(z_1)z_1^{-(p-1)} & \dots & G(z_{\ell-1})z_{\ell-1}^{-(p-1)} \\ I_r & I_r & \dots & I_r \\ z_0^{-1}I_r & z_1^{-1}I_r & \dots & z_{\ell-1}^{-1}I_r \\ \vdots & \vdots & \ddots & \vdots \\ z_0^{-(p-1)}I_r & z_1^{-(p-1)}I_r & \dots & z_{\ell-1}^{-(p-1)}I_r \end{array} \right] \quad (86)$$

$$\Theta = [-\alpha_0 \quad -\alpha_1 \quad \dots \quad -\alpha_{p-1} \quad \beta_0 \quad \beta_1 \quad \dots \quad \beta_{p-1}]$$

Note that  $\Phi$  is an  $(m+r)p \times (r\ell)$  matrix and  $\Theta$  is an  $m \times (m+r)p$  matrix. Equation (85) is a linear algebraic equation, implying that the parameter matrix  $\Theta$  is in the null column space of  $\Phi$ . The null column space of  $\Phi$  is identical to the null column space of  $\Phi\Phi^*$ , where  $*$  means complex conjugate and transpose. Postmultiplying equation (85) yields

$$\Theta\bar{\mathcal{R}} = 0_{m \times p(m+r)} \quad (87)$$

where

$$\bar{\mathcal{R}} = \Phi\Phi^* \quad (88)$$

The zero matrix  $0_{m \times p(m+r)}$  of dimension  $m \times p(m+r)$  (eq. (87)) is generally much smaller than  $0_{m \times \ell}$  (eq. (85)). Computing  $\Phi\Phi^*$  without fully forming the matrix  $\Phi$  is easy because of  $\Phi\Phi^*$  special configuration. Computing  $\Theta$  from equation (87) rather than from equation (85) is also much easier.

Equation (87) is identical in form to equation (75) except that  $\bar{\mathcal{R}}$  in equation (87) is a complex matrix and  $\mathcal{R}$  in equation (75) is a real matrix. Both equations (75) and (87) are derived to solve for ARX coefficient matrices  $\alpha$  and  $\beta$ . Both  $\alpha$  and  $\beta$  are real matrices. Therefore,  $\mathcal{R}$  and  $\bar{\mathcal{R}}$  should have common properties. Since  $\bar{\mathcal{R}}$  is a Hermitian matrix, that is,  $\bar{\mathcal{R}} = \bar{\mathcal{R}}^*$ , the real part of  $\bar{\mathcal{R}}$  is a symmetric matrix and the imaginary part of  $\bar{\mathcal{R}}$  is a skew symmetric matrix. It becomes intuitive to suggest that the real part of  $\bar{\mathcal{R}}$  be considered as  $\mathcal{R}$  as follows:

$$\mathcal{R} = \text{real part of } [\bar{\mathcal{R}}] = \text{real part of } [\Phi\Phi^*] \quad (89)$$

Although the matrix  $\mathcal{R}$  (eq. (89)) is constructed for computing the ARX coefficient matrices,  $\mathcal{R}$  may also be used for calculating the state matrix  $A$  and the output matrix  $C$ . Similar to the partition shown in equation (76), let  $\mathcal{R}$  be partitioned into four parts as follows:

$$\mathcal{R} = \begin{bmatrix} \mathcal{R}_{11} & \mathcal{R}_{12} \\ \mathcal{R}_{12}^T & \mathcal{R}_{22} \end{bmatrix} \quad (90)$$

where  $\mathcal{R}_{11}$  is a  $pm \times pm$  square matrix,  $\mathcal{R}_{12}$  a  $pm \times pr$  rectangular matrix, and  $\mathcal{R}_{22}$  a  $pr \times pr$  square matrix. Now consider the following conceptual equalities:

$$\mathcal{R}_{11} \equiv \mathcal{R}_{yy} \quad \mathcal{R}_{12} \equiv \mathcal{R}_{yu} \quad \mathcal{R}_{22} \equiv \mathcal{R}_{uu} \quad (91)$$

The  $pm \times pm$  matrix  $\mathcal{R}_{hh}$  defined in equation (20) can be formed as

$$\mathcal{R}_{hh} = \mathcal{R}_{11} - \mathcal{R}_{12}\mathcal{R}_{22}^{-1}\mathcal{R}_{12}^T \quad (92)$$

The matrix  $\mathcal{R}_{hh}$  can be used with either the full decomposition method or partial decomposition method to determine matrices  $A$  and  $C$ . On the other hand, the matrix product  $\mathcal{R}_{12}\mathcal{R}_{22}^{-1}$  can be used with either the direct method or indirect method to determine the input matrix  $B$  and the transmission matrix  $D$ .

#### 4.2.1. Transfer-Function Error Minimization Method

The indirect and direct methods for computing matrices  $B$  and  $D$  do not necessarily minimize the error between the real transfer function and estimated transfer function when small singular-values truncation is involved for computing matrices  $A$  and  $C$ . Similar to the output-error minimization method in the time domain, the transfer-function error minimization method forms an equation that explicitly relates the transfer function to matrices  $B$  and  $D$  with given matrices  $A$  and  $C$ .

The  $m \times r$  transfer function  $G(z_k)$  (eq. (81)) has another form of expression in terms of system matrices:

$$G(z_k) = D + C(z_k I_n - A)^{-1}B \quad (93)$$

for all  $z_k$  ( $k = 0, 1, \dots, \ell - 1$ ). Equation (93) produces

$$\begin{bmatrix} G(z_0) \\ G(z_1) \\ \vdots \\ G(z_{\ell-1}) \end{bmatrix} = \begin{bmatrix} I_m & C(z_0 I_n - A)^{-1} \\ I_m & C(z_1 I_n - A)^{-1} \\ \vdots & \vdots \\ I_m & C(z_{\ell-1} I_n - A)^{-1} \end{bmatrix} \begin{bmatrix} D \\ B \end{bmatrix} \quad (94)$$

where  $I_m$  and  $I_n$  are identity matrices of orders  $m$  and  $n$ , respectively. Given matrices  $A$ ,  $C$ , and  $G(z_k)$ , matrices  $B$  and  $D$  can then be determined as follows:

$$\begin{bmatrix} D \\ B \end{bmatrix} = \begin{bmatrix} I_m & C(z_0 I_n - A)^{-1} \\ I_m & C(z_1 I_n - A)^{-1} \\ \vdots & \vdots \\ I_m & C(z_{\ell-1} I_n - A)^{-1} \end{bmatrix}^\dagger \begin{bmatrix} G(z_0) \\ G(z_1) \\ \vdots \\ G(z_{\ell-1}) \end{bmatrix} \quad (95)$$

For noisy systems, the solutions for matrices  $B$  and  $D$  do not satisfy equation (94), but rather minimize the error in the least-square sense between the left side and the right side of equation (94). In practice, only the real part of equation (94) is needed to determine matrices  $B$  and  $D$ .

#### 4.2.2. Experimental Example

This example uses experimental data taken from a large truss-type structure. Figure 3 shows a Langley testbed (ref. 13) that is used to study controls-structures interaction problems (ref. 10). The system has eight pairs of colocated inputs and outputs for control. The inputs are air

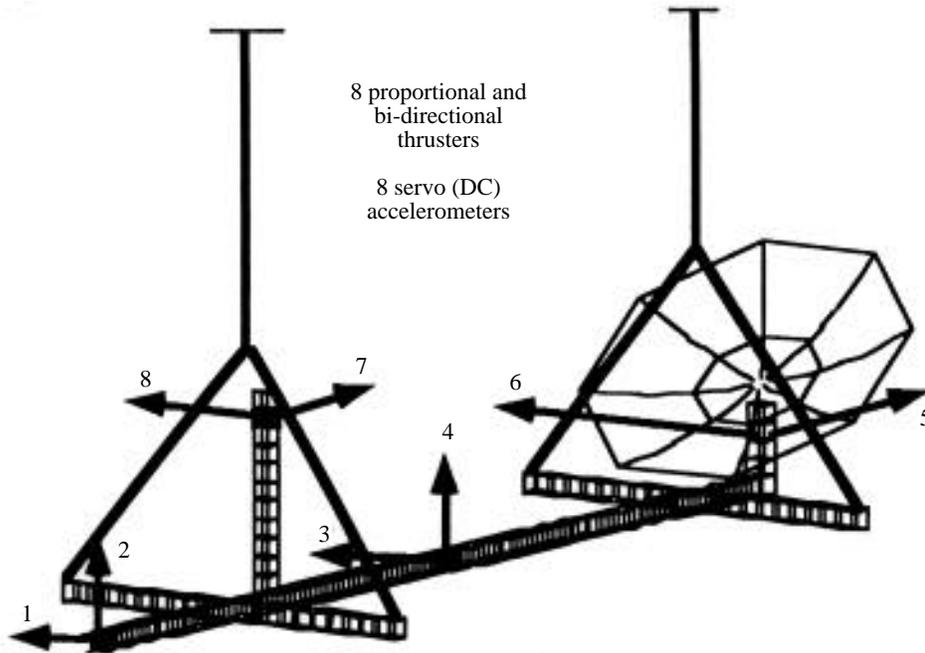


Figure 3. NASA large space-structure testbed.

thrusters and the outputs are accelerometers. Figure 3 depicts the locations of the input-output pairs. The structure was excited with random input signals to four thrusters located at positions 1, 2, 6, and 7. The input and output signals were filtered with low-pass digital filters, with the range set to 78 percent of the Nyquist frequency (12.8 Hz) to concentrate the energy in the low-frequency range below 10 Hz. A total of 2048 data points at a sampling rate of 25.6 Hz from each sensor are used for identification. In this example, four FRFs from two input and output pairs located at positions 1 and 2 are simultaneously used to identify a state-space model of the system.

The integer  $p = 50$  is sufficient to identify as many as 50 modes (for a system of dimension 100). A state-space model is obtained with the frequency-domain inversion with the system order truncated to 80 by singular-values truncation, and then further reduced to 78 by eliminating an unstable mode. Figure 4 shows the reconstructed frequency-response data (dashed lines) compared with the experimental data (solid lines).

Figure 4 shows the frequency response of output 1 with respect to input 1, which represents a strong signal. The reconstructed FRF is obtained with the identified system matrices  $A$ ,  $B$ ,  $C$ , and  $D$ , where matrices  $C$  and  $D$  are computed by the indirect method. There are 33 modes (corresponding to 66 states) with damping less than 1 percent. The largest singular value of the transfer-function error between the test and reconstructed FRFs is 12889. Careful examination of figure 4 reveals that there are noticeable discrepancies near the right end points of both the magnitude and phase plots. One way to fix the discrepancy is to recompute matrices  $B$  and  $D$  with the transfer-function error minimization method.

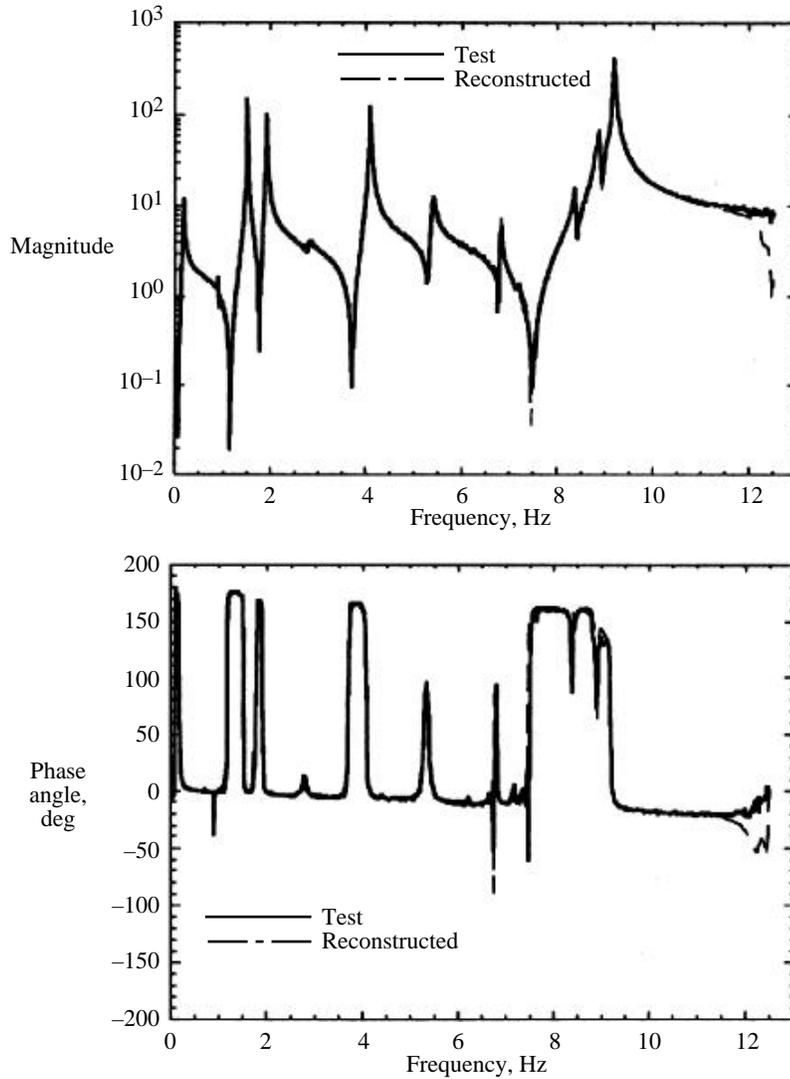


Figure 4. Comparison of test and reconstructed input-1/output-1 FRF's. Test data and input-1/output-1 FRF data reconstructed from the identified system matrices.

Figure 5 shows the frequency response of output 1 with respect to input 1 with the newly computed matrices  $B$  and  $D$ . The transfer-function error is reduced from 12889 to 65.184. There is clearly a trade-off, as the discrepancy between the test and reconstructed FRFs (fig. 4) is moved from about 12 Hz to about 8 Hz (fig. 5).

Similar results not shown in this paper were obtained for other input/output pairs. Note that the frequency response of output 2 with respect to input 1 represents a weak signal. The signal is weak because sensor 2 is orthogonal to input 1. The results show that matching is better for strong signals.

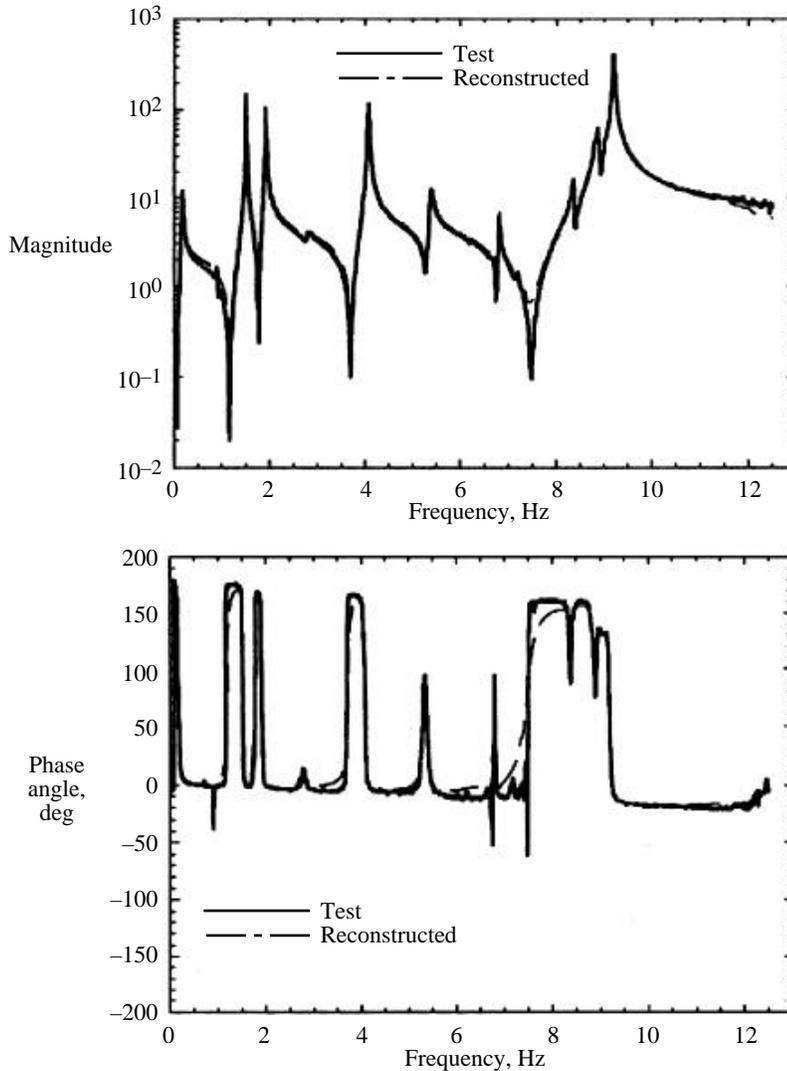


Figure 5. Comparison of test and reconstructed input-1/output-1 FRF's. Test data and input-1/output-1 FRF data reconstructed from matrices B and D.

### 4.3. Section 4 Summary

The idea of data correlation leads to relating the system realization method developed in this paper to other methods that are based on the discrete-time finite-difference model in the time domain and the frequency domain. The new state-space approach provides a way to better understand and interpret these methods through use of the information matrix. Indeed, the information matrix is the common ground for computing the system matrices. The method developed in this paper uses the basis vectors of the column space of the information matrix to compute the system matrices. On the other hand, other existing methods use the basis vectors of the null column space of the information matrix to compute the system matrices. In theory, both approaches should produce identical results. However, results may be noticeably different in practice because of system uncertainties and measurement noises, as evidenced by the experimental examples given in this paper. The mathematical unification presented in this paper should help users interpret the identification results obtained from different methods.

## 5. Concluding Remarks

This paper contains three main technical contributions. First, a generalized information matrix is introduced consisting of shifted input- and output-data correlation matrices. Second, a new system realization algorithm is derived with the information matrix as the basis for computing system matrices. Third, several system realization algorithms are unified via the system information matrix. For a pulse or free-decay response, the information matrix reduces to the shifted output-data correlation matrix. Therefore, the classical system realization algorithms based on a pulse response may be considered as a special case of the algorithm introduced in this paper, which implies that system realization algorithm unification includes classical realization methods.

NASA Langley Research Center  
Hampton, VA 23681-0001  
December 2, 1996

## Appendix A

### Efficient Computation of Output-Error Minimization Method

Computing  $\sum_{k=0}^{N-2} CA^{N-k-2}\underline{u}_n(k)$  of dimension  $m \times nr$  for any integer  $N$  may be achieved by making an imaginary linear system with zero initial conditions as follows:

$$\left. \begin{aligned} x_{ij}(N+1) &= Ax_{ij}(N) + e_i u_j(N) & x_{ij}(0) &= 0_{n \times 1} \\ z_{ij}(N) &= Cx_{ij}(N) & i &= 1, 2, \dots, n & j &= 1, 2, \dots, r \end{aligned} \right\} \quad (\text{A1})$$

where  $e_i$  is the  $i$ th column of the identity matrix of order  $n$ , and  $u_j(k)$  is the  $j$ th element of the vector  $u(k)$  defined in equation (60). At any time step  $N$ , equation (96) produces a matrix (such as  $Z(N)$ ) of  $m \times nr$  as

$$Z(N) = [z_{11}(N) \quad z_{21}(N) \quad \cdots \quad z_{n1}(N) \quad \cdots \quad z_{1r}(N) \quad z_{2r}(N) \quad \cdots \quad z_{nr}(N)] \quad (\text{A2})$$

which gives

$$\sum_{k=0}^{N-2} CA^{N-k-2}\underline{u}_n(k) = Z(N-1) \quad N > 1 \quad (\text{A3})$$

## Appendix B

### Efficient Computation of Information Matrix

Because of the nature of data shifting used to form  $Y_p(k)$  and  $U_p(k)$ , an efficient way of computing correlation matrices  $\mathcal{R}_{yy}$ ,  $\mathcal{R}_{yu}$ , and  $\mathcal{R}_{uu}$  exists. For simplicity without losing generality, consider the computation of  $\mathcal{R}_{yu}$ . The product of  $Y_p(k)U_p^T(k)$  can be written as

$$Y_p(k)U_p^T(k) = \begin{bmatrix} \sum_{\tau=k}^{k+N-1} y(\tau)u^T(\tau) & \sum_{\tau=k}^{k+N-1} y(\tau)u^T(\tau+1) & \sum_{\tau=k}^{k+N-1} y(\tau)u^T(\tau+2) & \cdots \\ \sum_{\tau=k+1}^{k+N} y(\tau)u^T(\tau-1) & \sum_{\tau=k+1}^{k+N} y(\tau)u^T(\tau) & \sum_{\tau=k+1}^{k+N} y(\tau)u^T(\tau+1) & \cdots \\ \sum_{\tau=k+2}^{k+N+1} y(\tau)u^T(\tau-2) & \sum_{\tau=k+2}^{k+N+1} y(\tau)u^T(\tau-1) & \sum_{\tau=k+2}^{k+N+1} y(\tau)u^T(\tau) & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (\text{B1})$$

From the pattern appearing in equation (B1), the user should have no difficulty filling out all elements not shown. For example, all diagonal elements are identical except for their upper and lower limits, and the same is true for all subdiagonal elements. As a result, the second diagonal element may be computed from the first diagonal element by

$$\sum_{\tau=k+1}^{k+N} y(\tau)u^T(\tau) = \sum_{\tau=k}^{k+N-1} y(\tau)u^T(\tau) + y(k+N)u^T(k+N) - y(k)u^T(k) \quad (\text{B2})$$

By induction, the diagonal  $m \times r$  submatrices can be computed recursively by

$$\sum_{\tau=k+i+1}^{k+N+i} y(\tau)u^T(\tau) = \sum_{\tau=k+i}^{k+N+i-1} y(\tau)u^T(\tau) + y(k+N+i)u^T(k+N+i) - y(k+i)u^T(k+i) \quad (\text{B3})$$

for  $i = 0, 1, \dots, p-1$ . This recursive formula indicates that each quantity is computed from its previous quantity. Similarly, the first upper off-diagonal submatrix on the second  $m$  rows may be calculated from the first upper off-diagonal submatrix on the first  $m$  rows by

$$\sum_{\tau=k+1}^{k+N} y(\tau)u^T(\tau+1) = \sum_{\tau=k}^{k+N-1} y(\tau)u^T(\tau+1) + y(k+N)u^T(k+N+1) - y(k)u^T(k+1) \quad (\text{B4})$$

Therefore, the first upper off-diagonal  $m \times r$  submatrices can be computed recursively by

$$\begin{aligned} \sum_{\tau=k+i+1}^{k+N+i} y(\tau)u^T(\tau+1) &= \sum_{\tau=k+i}^{k+N+i-1} y(\tau)u^T(\tau+1) + y(k+N+i)u^T(k+N+i+1) \\ &\quad - y(k+i)u^T(k+i+1) \end{aligned} \quad (\text{B5})$$

for  $i = 0, 1, \dots, p - 2$ . Furthermore, the second upper off-diagonal  $m \times r$  submatrices are calculated recursively by

$$\sum_{\tau=k+i+1}^{k+N+i} y(\tau)u^T(\tau+2) = \sum_{\tau=k+i}^{k+N+i-1} y(\tau)u^T(\tau+2) + y(k+N+i)u^T(k+N+i+2) - y(k+i)u^T(k+i+2) \quad (\text{B6})$$

for  $i = 0, 1, \dots, p - 3$ . Similarly, the third, the fourth, up to the  $(p - 1)$ th upper off-diagonal quantities can be calculated recursively as soon as their first upper off-diagonal quantities are known. The first upper off-diagonal quantities are the first  $m$  rows of the product  $Y_p(k)U_p^T(k)$ .

Similarly, the lower columns may be computed as soon as the first  $m$  columns are calculated. For example, the first lower off-diagonal  $m \times r$  submatrices may be computed recursively by

$$\sum_{\tau=k+i+1}^{k+N+i} y(\tau)u^T(\tau-1) = \sum_{\tau=k+i}^{k+N+i-1} y(\tau)u^T(\tau-1) + y(k+N+i)u^T(k+N+i-1) - y(k+i)u^T(k+i-1) \quad (\text{B7})$$

for  $i = 1, \dots, p - 1$ . The second lower off-diagonal  $m \times r$  submatrices may be calculated recursively by

$$\sum_{\tau=k+i+1}^{k+N+i} y(\tau)u^T(\tau-2) = \sum_{\tau=k+i}^{k+N+i-1} y(\tau)u^T(\tau-2) + y(k+N+i)u^T(k+N+i-2) - y(k+i)u^T(k+i-2) \quad (\text{B8})$$

for  $i = 2, \dots, p - 1$ . By induction, the third, the fourth, up to the  $(p - 1)$ th lower off-diagonal quantities can be calculated recursively as long as their first lower off-diagonal quantities are determined. The first lower off-diagonal quantities in equation (B8) are the first  $m$  columns of the product  $Y_p(k)U_p^T(k)$ .

Computing the product  $Y_p(k)U_p^T(k)$  becomes recursive as soon as the first  $m$  rows and the first  $m$  columns are computed. For a long data record, that is,  $N \gg 1$ , this recursive procedure has a very efficient computing time. The computations of  $Y_p(k)Y_p^T(k)$  and  $U_p(k)U_p^T(k)$  are similar to the computation of  $Y_p(k)U_p^T(k)$ . Because both  $Y_p(k)Y_p^T(k)$  and  $U_p(k)U_p^T(k)$  are symmetric, only the upper half or the lower half must be computed.

## 6. References

1. Juang, J.-N.: *Applied System Identification*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632, 1994.
2. Juang, J.-N.; and Pappa, R. S.: A Comparative Overview of Modal Testing and System Identification for Control of Structures, *Shock and Vibration Digest*, vol. 20, no. 5, May 1988, pp. 4–15.
3. Denman, E.; Hasselman, T.; Sun, C. T.; Juang, J.-N.; Junkins, J. L.; Udawadia, F.; Venkayya, V.; and Kamat M.: *Identification of Large Space Structures on Orbit*, AFRPL TR-86-054, Air Force Rocket Propulsion Laboratory, Edwards Air Force Base, California 93523-5000.
4. Kung, S.: A New Identification and Model Reduction Algorithm Via Singular Value Decomposition, *12th Asilomar Conference on Circuits, Systems and Computers*, November 1978, pp. 705–714.
5. Hollkamp, J. J.; and Batill, S. M.: Automated Parameter Identification and Order Reduction for Discrete Series Models, *AIAA Journal*, vol. 29, no. 1, 1991.
6. Moonen, M.; Demoor, B.; Vandenberghe, L.; and Vandewalle, J.: On- and Off-Line Identification of Linear State-Space Models, *International Journal of Control*, vol. 49, 1989, pp. 219–232.
7. Moonen, M.; and Vandenberghe, L.: QSVD Approach to On- and Off-line State Space Identification, *International Journal of Control*, vol. 50, 1990, pp. 1133–1146.
8. Verhaegen, M.: Subspace Model Identification: Part 1—The Output-Error State-Space Model Identification Class of Algorithms, *International Journal of Control*, vol. 56, no. 5, pp. 1187–1210, 1992.
9. Ljung, L.: *System Identification: Theory for the User*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1987.
10. Juang, J.-N.; Cooper, J. E.; and Wright, J. R.: An Eigensystem Realization Algorithm Using Data Correlations (ERA/DC) for Modal Parameter Identification, *Control-Theory and Advanced Technology*, vol. 4, no. 1, 1988, pp. 5–14.
11. Juang, J.-N.; Phan, M.; Horta, L. G.; and Longman, R. W.: Identification of Observer/Kalman Filter Markov Parameters: Theory and Experiments, *Journal of Guidance, Control and Dynamics*, vol. 16, no. 2, March–April 1993, pp. 320–329.
12. Chen, C. W.; Juang, J.-N.; and Lee, G.: “Frequency Domain State-Space System Identification,” Transactions of the ASME, *Journal of Vibration and Acoustics*, vol. 116, no. 4, October 1994, pp. 523–528.
13. Belvin, W. K.; Elliott, K. B.; Horta, L. G.; Bailey, J. P.; Bruner, A. M.; Sulla, J. L.; Won, J.; and Ugoletti, R. M.: *Langley’s CSI Evolutionary Model: Phase 0*, NASA Technical Memorandum 104165, November 1991.

## REPORT DOCUMENTATION PAGE

*Form Approved*  
*OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

<b>1. AGENCY USE ONLY</b> <i>(Leave blank)</i>	<b>2. REPORT DATE</b> April 1997	<b>3. REPORT TYPE AND DATES COVERED</b> Technical Paper	
<b>4. TITLE AND SUBTITLE</b>  State-Space System Realization With Input- and Output-Data Correlation		<b>5. FUNDING NUMBERS</b>  WU 233-10-14-03	
<b>6. AUTHOR(S)</b>  Jer-Nan Juang			
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b>  NASA Langley Research Center Hampton, VA 23681-0001		<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>  L-17548	
<b>9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b>  National Aeronautics and Space Administration Washington, DC 20546-0001		<b>10. SPONSORING/MONITORING AGENCY REPORT NUMBER</b>  NASA TP-3622	
<b>11. SUPPLEMENTARY NOTES</b>			
<b>12a. DISTRIBUTION/AVAILABILITY STATEMENT</b>  Unclassified-Unlimited Subject Category 39 Availability: NASA CASI (301) 621-0390		<b>12b. DISTRIBUTION CODE</b>	
<b>13. ABSTRACT</b> <i>(Maximum 200 words)</i>  This paper introduces a general version of the information matrix consisting of the autocorrelation and cross-correlation matrices of the shifted input and output data. Based on the concept of data correlation, a new system realization algorithm is developed to create a model directly from input and output data. The algorithm starts by computing a special type of correlation matrix derived from the information matrix. The special correlation matrix provides information on the system-observability matrix and the state-vector correlation. A system model is then developed from the observability matrix in conjunction with other algebraic manipulations. This approach leads to several different algorithms for computing system matrices for use in representing the system model. The relationship of the new algorithms with other realization algorithms in the time and frequency domains is established with matrix factorization of the information matrix. Several examples are given to illustrate the validity and usefulness of these new algorithms.			
<b>14. SUBJECT TERMS</b> System identification; System realization; Input/output map; State-space model identification.		<b>15. NUMBER OF PAGES</b> 41	
		<b>16. PRICE CODE</b> A03	
<b>17. SECURITY CLASSIFICATION OF REPORT</b> Unclassified	<b>18. SECURITY CLASSIFICATION OF THIS PAGE</b> Unclassified	<b>19. SECURITY CLASSIFICATION OF ABSTRACT</b> Unclassified	<b>20. LIMITATION OF ABSTRACT</b>