JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 20, No. 3, May–June 1997

System Realization Using Information Matrix

Jer-Nan Juang*
NASA Langley Research Center, Hampton, Virginia 23681-0001

A generalized version of the information matrix is introduced 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 identify a model directly from input and output data. The algorithm starts with computing the information matrix to derive a special correlation matrix that in turn produces the system observability matrix and the state-vector correlation. A system model can then be identified from the observability matrix in conjunction with other algebraic manipulations. The algorithm leads to several different methods for computing system matrices to represent the system model. An experimental example is given to illustrate the validity and usefulness of these methods with some comparison.

Nomenclature

A	= state matrix, $n \times n$
B	= input influence matrix, $n \times r$
$\boldsymbol{b}_i$	= ith column of B, $n \times 1$
C	= output influence matrix, $m \times n$
D	= direct transmission matrix, $m \times r$
$\boldsymbol{d}_i$	= ith column of D, $m \times 1$
I_i	= identity matrix of order i
k	= time index
ℓ	= data length
m	= number of outputs
n	= order of the system
n_0	= 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 the maximum order of the 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, $pr \times pr$
$\mathcal{R}_{uy}$, $\tilde{\mathcal{R}}_{uy}$	= cross correlation of input and output vectors, $pr \times pm$
$\mathcal{R}_{xu}$	= 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, $n \times n$
$\mathcal{R}_{xy}$	= cross correlation of state and output vectors, $n \times pm$
$\mathcal{R}_{yy}$	= autocorrelation of output vector, $pm \times pm$
r	= number of inputs
$\mathcal{T}_p$	= Toeplitz matrix, $pm \times pr$
$\mathcal{U}$	= left singular matrix
$\mathfrak{U}_n(k)$	= special force matrix at time index k, $m \times r$
$\mathcal{U}_n$	= columns of left singular matrix corresponding to nonzero singular values
$U_p(k)$	= input matrix containing $\boldsymbol{u}_p(k)$ up to $\boldsymbol{u}_p(k+N-1)$, $pr \times N$
$\mathcal{U}_0$	= columns of left singular matrix corresponding to zero singular values
$\mathcal{U}_{0n}$, $\mathcal{U}_{0\mathcal{T}}$	= working matrices
$\boldsymbol{u}(k)$	= input force vector at time index k, $r \times 1$
$u_i(k)$	= ith input force at time index k
$\boldsymbol{u}_p(k)$	= vector containing $\boldsymbol{u}(k)$ up to $\boldsymbol{u}(k+p-1)$, $rp \times 1$
$\boldsymbol{u}'_N(k)$	= vector containing $\boldsymbol{u}(k)$ up to $\boldsymbol{u}(k+N-1)$, $pN \times 1$
$\mathcal{V}$	= right singular matrix
$X(k)$	= matrix containing $\boldsymbol{x}(k)$ up to $\boldsymbol{x}(k+N-1)$, $n \times N$
$\boldsymbol{x}(k)$	= state vector at time index k, $n \times 1$
$Y_p(k)$	= output matrix containing $\boldsymbol{y}_p(k)$ to $\boldsymbol{y}_p(k+N-1)$, $pm \times N$
$\boldsymbol{y}(k)$	= output measurement vector at time index k, $m \times 1$
$\boldsymbol{y}_p(k)$	= vector containing $\boldsymbol{y}(k)$ up to $\boldsymbol{y}(k+p-1)$, $pm \times 1$
Γ	= working matrix
Σ	= singular value matrix
Σ_n	= diagonal matrix containing nonzero singular values
σ_i	= ith singular value
0_i	= zero square matrix of order i
$0_{i \times j}$	= zero rectangular matrix of dimension i by j

I. Introduction

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

This paper is motivated by the need to improve system identification techniques such as those in Refs. 1–3. These techniques require a QR factorization of a large Hankel matrix followed by a singular value decomposition and the solution of an overdetermined set of equations. Computational time and numerical accuracy become an issue when the length of measurement data is considerably long. The initial attempt is to find an alternate procedure for these techniques to perform system identification more efficiently in computation. The approach is to use the concept of data correlations as presented in Refs. 4–6. As a result, a new algorithm called system realization using information matrix (SRIM) is developed. The information matrix is similar to the one defined in Ref. 7 for the frequency-domain analysis, but it 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. The SRIM algorithm includes several methods with different merits for computing system matrices including the state matrix, the input and output matrices, and the direct transmission matrix to form a discrete-time model.

The eigensystem realization algorithm with data correlation (ERA/DC)[4] uses the shifted data correlation of the pulse response and factors the correlation matrix via the singular value

Received Feb. 23, 1996; revision received Dec. 23, 1996; accepted for publication Dec. 23, 1996.

*Principal Scientist, Structural Dynamics Branch. Fellow AIAA.

decomposition to realize a set of system matrices. The pulse response may be obtained by a pulse input or computed from input and output data via the observer/Kalman filter identification (OKID) technique.[5] Thus, the SRIM algorithm presented in this paper may be considered as an extension of the ERA/DC for system identification directly from input and output data.

To derive the SRIM algorithm, we start with the description of a discrete-time state-space model and give some key definitions, such as the observability matrix and the Toeplitz matrix. It follows by the development of the state-space model realization to compute the system matrices. Then the computational steps are provided for the algorithm. Finally, an experimental example is given for illustration.

II. State-Space Model

A deterministic linear time-invariant system is commonly represented by the following discrete-time state-space model[6]:

$$\begin{aligned} \boldsymbol{x}(k+1) &= A\boldsymbol{x}(k) + B\boldsymbol{u}(k) \\ \boldsymbol{y}(k) &= C\boldsymbol{x}(k) + D\boldsymbol{u}(k) \end{aligned} \tag{1}$$

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

With some algebraic manipulations, Eq. (1) produces

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

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

$$\boldsymbol{y}_p(k) = \begin{bmatrix} \boldsymbol{y}(k) \\ \boldsymbol{y}(k+1) \\ \boldsymbol{y}(k+2) \\ \vdots \\ \boldsymbol{y}(k+p-1) \end{bmatrix}$$

$$\mathcal{O}_p = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{p-1} \end{bmatrix} \qquad \boldsymbol{u}_p(k) = \begin{bmatrix} \boldsymbol{u}(k) \\ \boldsymbol{u}(k+1) \\ \boldsymbol{u}(k+2) \\ \vdots \\ \boldsymbol{u}(k+p-1) \end{bmatrix} \tag{3}$$

$$\mathcal{T}_p = \begin{bmatrix} D & & & & \\ CB & D & & & \\ CAB & CB & D & & \\ \vdots & \vdots & \vdots & \ddots & \\ CA^{p-2}B & CA^{p-3}B & CA^{p-4}B & \cdots & D \end{bmatrix}$$

Equation (2) thus becomes

$$\boldsymbol{y}_p(k) = \mathcal{O}_p\boldsymbol{x}(k) + \mathcal{T}_p\boldsymbol{u}_p(k) \tag{4}$$

The matrix $\mathcal{O}_p$ of dimension $pm \times n$ is commonly called the observability matrix and is formed from the state matrix A and the output matrix C. The matrix $\mathcal{T}_p$ of dimension $pm \times pr$ is a generalized Toeplitz matrix formed from the system Markov parameters D and CA^kB for $k = 0, 1, \ldots, p-2$. Note that $\mathcal{T}_p$ is unique even though A, B, C, and D are not unique, because the system Markov parameters D and CA^kB are unique. The use of Eq. (2) to realize a system model is presented in the following section.

III. State-Space Model Realization

In state-space system identification, the goal 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, i.e., 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 using the first $(p-1)m$ rows and all columns of $\mathcal{O}_p$, i.e.,

$$\begin{aligned} \mathcal{O}_p(m+1:pm,:) &= \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{p-1} \end{bmatrix} \\ \mathcal{O}_p[1:(p-1)m,:] &= \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{p-2} \end{bmatrix} \end{aligned} \tag{5}$$

which yields the following equality:

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

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

$$A = \mathcal{O}_p^{\dagger}[1:(p-1)m,:]\mathcal{O}_p(m+1:pm,:) \tag{7}$$

where † means the pseudoinverse. Note that the integer p should be chosen such that the matrix $\mathcal{O}_p(m+1:pm,:)$ of dimension $(p-1)m \times n$ has rank larger than or equal to n, i.e.,

$$(p-1)m \geq n \quad \Rightarrow \quad p \geq (n/m) + 1 \tag{8}$$

where n is the order of the system.

Similarly, the first m rows and the first r columns of the matrix $\mathcal{T}_p$ shown in Eq. (3) 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 $\mathcal{T}_p$, i.e.,

$$\begin{aligned} \mathcal{T}[m+1:(p-1)m, 1:r] &= \begin{bmatrix} CB \\ CAB \\ \vdots \\ CA^{p-2}B \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{p-2} \end{bmatrix} B \\ &= \mathcal{O}_p[1:(p-1)m,:]B \end{aligned} \tag{9}$$

Equation (9) clearly produces

$$B = \mathcal{O}_p^{\dagger}[1:(p-1)m, :]\mathcal{T}[m+1:(p-1)m, 1:r] \tag{10}$$

To determine $\mathcal{O}_p$ and $\mathcal{T}_p$, first expand the vector equation, Eq. (4), to a matrix equation, i.e.,

$$Y_p(k) = \mathcal{O}_p X(k) + \mathcal{T}_p U_p(k) \tag{11}$$

where

$$X(k) = [\boldsymbol{x}(k) \quad \boldsymbol{x}(k+1) \quad \cdots \quad \boldsymbol{x}(k+N-1)]$$

$$Y_p(k) = [\boldsymbol{y}_p(k) \quad \boldsymbol{y}_p(k+1) \quad \cdots \quad \boldsymbol{y}_p(k+N-1)]$$

$$= \begin{bmatrix} \boldsymbol{y}(k) & \boldsymbol{y}(k+1) & \cdots & \boldsymbol{y}(k+N-1) \\ \boldsymbol{y}(k+1) & \boldsymbol{y}(k+2) & \cdots & \boldsymbol{y}(k+N) \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{y}(k+p-1) & \boldsymbol{y}(k+p) & \cdots & \boldsymbol{y}(k+p+N-2) \end{bmatrix}$$

$$U_p(k) = [\boldsymbol{u}_p(k) \quad \boldsymbol{u}_p(k+1) \quad \cdots \quad \boldsymbol{u}_p(k+N-1)] \tag{12}$$

$$= \begin{bmatrix} \boldsymbol{u}(k) & \boldsymbol{u}(k+1) & \cdots & \boldsymbol{u}(k+N-1) \\ \boldsymbol{u}(k+1) & \boldsymbol{u}(k+2) & \cdots & \boldsymbol{u}(k+N) \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{u}(k+p-1) & \boldsymbol{u}(k+p) & \cdots & \boldsymbol{u}(k+p+N-2) \end{bmatrix}$$

The integer N must be sufficiently large such that the rank of $Y_p(k)$ and $U_p(k)$ is at least equal to the rank of $\mathcal{O}_p$. Equation (11) is the key equation to be used to solve for $\mathcal{O}_p$ and $\mathcal{T}_p$, which 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, we must focus on these two matrices to extract other information necessary to determine the system matrices A, B, C, and D.

Let us define the following quantities:

$$\mathcal{R}_{yy} = (1/N)Y_p(k)Y_p^T(k) \qquad \mathcal{R}_{yu} = (1/N)Y_p(k)U_p^T(k)$$

$$\mathcal{R}_{uu} = (1/N)U_p(k)U_p^T(k) \qquad \mathcal{R}_{xx} = (1/N)X(k)X^T(k) \tag{13}$$

$$\mathcal{R}_{yx} = (1/N)Y_p(k)X^T(k) \qquad \mathcal{R}_{xu} = (1/N)X(k)U_p^T(k)$$

where $N = \ell - p$, with ℓ 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 $\boldsymbol{y}$ with time shifts, the input data u with time shifts, and the state vector $\boldsymbol{x}$, respectively. The rectangular matrices $\mathcal{R}_{yu}$ $(mp \times rp)$, $\mathcal{R}_{yx}$ $(mp \times n)$, and $\mathcal{R}_{xu}$ $(n \times rp)$ represent, respectively, the cross correlations of the output data $\boldsymbol{y}$ and input data u, the output data $\boldsymbol{y}$ and the state vector $\boldsymbol{x}$, and the state vector $\boldsymbol{x}$ and the input data u. When the integer N is sufficiently large, the quantities defined in Eq. (13) approximate expected values in the statistical sense if the input and output data are stationary processes satisfying the ergodic property.

In view of Eq. (13), postmultiplying Eq. (11) by $U_p^T(k)$ and then dividing it by N yields

$$\mathcal{R}_{yu} = \mathcal{O}_p\mathcal{R}_{xu} + \mathcal{T}_p\mathcal{R}_{uu} \tag{14}$$

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

$$\mathcal{T}_p = [\mathcal{R}_{yu} - \mathcal{O}_p\mathcal{R}_{xu}]\mathcal{R}_{uu}^{-1} \tag{15}$$

The inverse matrix $\mathcal{R}_{uu}^{-1}$ exists only when integers p and N are properly chosen such that $\mathcal{R}_{uu}$ has at least rank rp. Similarly, postmultiplying Eq. (11) by $Y_p^T(k)$ yields

$$\mathcal{R}_{yy} = \mathcal{O}_p\mathcal{R}_{yx}^T + \mathcal{T}_p\mathcal{R}_{yu}^T \tag{16}$$

and postmultiplying Eq. (11) by $X^T(k)$ gives

$$\mathcal{R}_{yx} = \mathcal{O}_p\mathcal{R}_{xx} + \mathcal{T}_p\mathcal{R}_{xu}^T \tag{17}$$

Substituting Eq. (15) for $\mathcal{T}_p$ into Eqs. (16) and (17) and the resulting equation for $\mathcal{R}_{yx}$ into Eq. (16) produces

$$\mathcal{R}_{yy} - \mathcal{R}_{yu}\mathcal{R}_{uu}^{-1}\mathcal{R}_{yu}^T = \mathcal{O}_p\mathcal{R}_{xx}\mathcal{O}_p^T - \mathcal{O}_p\mathcal{R}_{xu}\mathcal{R}_{uu}^{-1}\mathcal{R}_{xu}^T\mathcal{O}_p^T \tag{18}$$

Now define

$$\mathcal{R}_{hh} = \mathcal{R}_{yy} - \mathcal{R}_{yu}\mathcal{R}_{uu}^{-1}\mathcal{R}_{yu}^T \tag{19}$$

and

$$\tilde{\mathcal{R}}_{xx} = \mathcal{R}_{xx} - \mathcal{R}_{xu}\mathcal{R}_{uu}^{-1}\mathcal{R}_{xu}^T \tag{20}$$

Equation (18) becomes

$$\mathcal{R}_{hh} = \mathcal{O}_p\tilde{\mathcal{R}}_{xx}\mathcal{O}_p^T \tag{21}$$

Equation (21) is the key equation to be used for determination of the 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 $\boldsymbol{u}(k)$ for $k = 1, 2, \ldots, \ell$ is rich and persistent, which results in a matrix $U_p(k)$ of full rank, i.e., rp. Assume that the input signal $u(i)$ for $i \geq k$ is uncorrelated with the state vector $\boldsymbol{x}(k)$ at time step k. In other words, 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 $\tilde{\mathcal{R}}_{xx}$ defined in Eq. (20) reduces to $\mathcal{R}_{xx}$. For example, if the input u is a zero-mean, white, and Gaussian random signal, then $\tilde{\mathcal{R}}_{xx} = \mathcal{R}_{xx}$ when the data length is sufficiently long, i.e, $N \to \infty$ in theory.

Let $\mathcal{R}$ be defined as

$$\mathcal{R} = \begin{bmatrix} \mathcal{R}_{yy} & \mathcal{R}_{yu} \\ \mathcal{R}_{yu}^T & \mathcal{R}_{uu} \end{bmatrix} = \begin{bmatrix} Y_p(k) \\ U_p(k) \end{bmatrix} \begin{bmatrix} Y_p^T(k) & U_p^T(k) \end{bmatrix} \tag{22}$$

The matrix $\mathcal{R}$ is defined here as the information matrix, which 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 of the information necessary to compute the system matrices A, B, C, and D. Factoring $\mathcal{R}$ yields

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

$$\times \begin{bmatrix} \mathcal{R}_{hh} & 0_{pm\times pr} \\ 0_{pr\times pm} & \mathcal{R}_{uu} \end{bmatrix} \begin{bmatrix} I_{pm} & 0_{pm\times pr} \\ \mathcal{R}_{uu}^{-1}\mathcal{R}_{yu} & I_{pr} \end{bmatrix} \tag{23}$$

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-hand side of Eq. (23) is a positive semidefinite or a positive definite matrix. In the matrix triple product on the right-hand side of Eq. (23), 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, i.e.,

$$\mathcal{R}_{hh} \geq 0 \tag{24}$$

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

The left-hand side of Eq. (21), i.e., the symmetric matrix $\mathcal{R}_{hh}$, is known from input and output data, whereas the right-hand side is formed from the product of the rectangular matrix $\mathcal{O}_p$ of dimension $mp \times n$, the symmetric matrix $\tilde{\mathcal{R}}_{xx}$ of dimension $n \times n$, and the transpose of $\mathcal{O}_p$. It is very clear that the matrix $\mathcal{R}_{hh}$ must be factored into three matrices in order to solve for the observability matrix $\mathcal{O}_p$. That is, indeed, the approach to be taken in the following section.

A. Computation of A and C

Two methods for computing A and C are shown in this section. One method decomposes the full matrix $\mathcal{R}_{hh}$ and, thus, is called the full decomposition method. The other method decomposes a portion of $\mathcal{R}_{hh}$ and is referred to as the partial decomposition method.

1. *Full Decomposition Method*

Given the 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. The 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

$$\mathcal{R}_{hh} = \mathcal{U}\Sigma^2\mathcal{U}^T = [\mathcal{U}_n \quad \mathcal{U}_0]\begin{bmatrix} \Sigma_n^2 & 0_{n\times n_0} \\ 0_{n_0\times n} & 0_{n_0} \end{bmatrix}\begin{bmatrix} \mathcal{U}_n^T \\ \mathcal{U}_0^T \end{bmatrix} = \mathcal{U}_n\Sigma_n^2\mathcal{U}_n^T \tag{25}$$

The integer $n_0 = pm - n$ is the number of dependent columns in $\mathcal{R}_{hh}$, $0_{n\times n_0}$ is an $n \times n_0$ zero matrix, and 0_{n_0} is a square zero matrix of order n_0. The $pm \times n$ matrix $\mathcal{U}_n$ corresponds to the n nonzero singular values in the diagonal matrix Σ_n, whereas the $pm \times n_0$ matrix $\mathcal{U}_0$ is associated with the n_0 zero singular values. Combining Eq. (21) with Eq. (25) thus produces

$$\mathcal{R}_{hh} = \mathcal{O}_p\tilde{\mathcal{R}}_{xx}\mathcal{O}_p^T = \mathcal{U}_n\Sigma_n^2\mathcal{U}_n^T \tag{26}$$

The last equality produces one solution for $\mathcal{O}_p$ and $\tilde{\mathcal{R}}_{xx}$, i.e.,

$$\mathcal{O}_p = \mathcal{U}_n \tag{27}$$

and

$$\tilde{\mathcal{R}}_{xx} = \Sigma_n^2 \tag{28}$$

Equation (27) 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 using Eq. (7). The first m rows of $\mathcal{U}_n$ constitute the output matrix C.

Equation (28) gives the correlation $\tilde{\mathcal{R}}_{xx}$, defined in Eq. (20), as the singular value matrix Σ_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 $\mathcal{R}_{xx}$ defined in Eq. (13), which is nothing but the correlation of the state vector $\boldsymbol{x}$. The diagonal nature of Σ_n^2 implies that all individual elements of the state vector $\boldsymbol{x}$ are linearly independent and orthogonal (uncoupled). Each individual element of the state vector $\boldsymbol{x}(k)$ represents one coordinate. The importance of each coordinate can then be measured by the magnitude of its corresponding singular value.

Let the diagonal matrix Σ_n be denoted by

$$\Sigma_n = \mathrm{diag}[\sigma_1, \ \sigma_2, \ \ldots, \ \sigma_n] = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n \end{bmatrix} \tag{29}$$

with monotonically nonincreasing σ_i $(i = 1, \ 2, \ \ldots, \ n)$:

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$$

Accordingly, the strength of the elements (coordinates) in the state vector $\boldsymbol{x}(k)$ can be quantified by the singular values. Assume that the singular values $\sigma_{i+1}, \ldots, \sigma_n$ 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 $\sigma_{i+1}, \ldots, \sigma_n$ are negligible in comparison with the other coordinates. The order of the system may then be reduced from n to i by deleting singular values $\sigma_{i+1}, \ldots, \sigma_n$.

In practice, none of the singular values will be identically zero because of system uncertainties and measurement noise, implying that $n_0 = 0$, $n = pm$, and $\mathcal{U}_0 = [\,]$ (empty) without singular values truncation. The observability matrix $\mathcal{O}_p$ shown in Eq. (27) becomes a square matrix because $\mathcal{U}_n$ obtained from Eq. (26) is a $pm \times pm$ matrix. The equality, Eq. (8), is violated indicating that Eq. (7) 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 in order to use the full decomposition method. In other words, the last m columns of $\mathcal{U}_n$ must be truncated and treated as $\mathcal{U}_0$. To overcome this problem, another method is presented in the following section.

2. *Partial Decomposition Method*

Regardless of what integer p is chosen, the minimum value of n_0 must be m (the number of outputs) to make $n < pm$ and then satisfy the equality constraint, Eq. (26). There is one way of avoiding any singular values truncation. Instead of taking the singular value decomposition of the $pm \times pm$ square matrix $\mathcal{R}_{hh}$, let us factor only part of the matrix, i.e.,

$$\begin{aligned} \mathcal{R}_{hh}[:, 1 : (p-1)m] &= \mathcal{U}\Sigma^2\mathcal{V}^T \\ &= [\mathcal{U}_n' \quad \mathcal{U}_0']\begin{bmatrix} \Sigma_n^2 & 0_{n\times n_0} \\ 0_{n_0'\times n} & 0_{n_0'\times n_0} \end{bmatrix}\begin{bmatrix} \mathcal{V}_n^T \\ \mathcal{V}_0^T \end{bmatrix} = \mathcal{V}_n\Sigma_n^2\mathcal{V}_n^T \end{aligned} \tag{30}$$

The dimension of $\mathcal{R}_{hh}[:, 1 : (p-1)m]$ is $pm \times (p-1)m$, i.e., there are more rows than columns. The integer n_0 indicates the number of zero singular values and also the number of columns of $\mathcal{V}_0$. The integer n_0' is the number of columns of $\mathcal{U}_0'$ that are orthogonal to the columns of $\mathcal{U}_n'$. For noisy data, there will not be any zero singular values, i.e., $n_0 = 0$. If no singular values are truncated, one obtains $n_0' = m$. If some small singular values are truncated, n_0' becomes the sum of m and the number of truncated singular values. In other words, there are at least m columns of $\mathcal{U}_0'$ that are orthogonal to the columns of $\mathcal{U}_n'$ in Eq. (30). From Eq. (26), it is easy to show that

$$\mathcal{R}_{hh}[:, 1 : (p-1)m] = \mathcal{O}_p\tilde{\mathcal{R}}_{xx}\mathcal{O}_p^T[:, 1 : (p-1)m] = \mathcal{U}_n'\Sigma_n^2\mathcal{V}_n^T \tag{31}$$

which yields one solution,

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

The second equation in Eq. (32) 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 the first equation must be satisfied. One important feature of this approach is that there are always enough columns of $\mathcal{U}_0'$ for use in computing A and C with or without singular values truncation. The disadvantage is that the singular values do not represent the correlation of the state vector anymore.

B. Computation of *B* and *D*

Similar to computing A and C, there are three methods available for computing B and D. The first method, called the indirect method, uses the column vectors, which 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 the output error between the measured output and the reconstructed output. The reconstructed output is the output time history obtained using the input time history to drive the identified system model, which is represented by the computed matrices A, B, C, and D.

1. *Indirect Method*

With A and C known, the input matrix B and the direct transmission matrix D can be computed from the Toeplitz matrix $\mathcal{T}_p$ defined in Eq. (3). To formulate an equation to solve for $\mathcal{T}_p$, one must find a way to eliminate the term associated with the observability matrix $\mathcal{O}_p$ from Eq. (14).

In view of Eqs. (25) and (27), premultiplying Eq. (14) by $\mathcal{U}_0^T$ and using the orthogonality property of $\mathcal{U}_n$ and $\mathcal{U}_0$ yields

$$\mathcal{U}_0^T\mathcal{R}_{yu} = \mathcal{U}_0^T\mathcal{T}_p\mathcal{R}_{uu}$$

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

$$\mathcal{U}_0^T\mathcal{T}_p = \mathcal{U}_0^T\mathcal{R}_{yu}\mathcal{R}_{uu}^{-1} \tag{33}$$

Equation (33) is the fundamental equation to solve for the input matrix B and the direct transmission matrix D. Note that Eq. (33) by no means implies that $\mathcal{T}_p = \mathcal{R}_{yu}\mathcal{R}_{uu}^{-1}$, because $\mathcal{U}_0^T$ is a rectangular matrix of dimension $n_0 \times mp$ with $n_0 < mp$. The right-hand side of Eq. (33) is a known quantity, whereas the left-hand side contains the matrix $\mathcal{T}_p$, which is partially known including A and C and partially unknown including B and D. Therefore, one must partition the matrix $\mathcal{T}_p$ into two parts to extract matrices B and D.

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

$$\mathcal{T}_p = \{\mathcal{T}_p(:, 1:r) \quad \mathcal{T}_p(:, r{+}1:2r) \quad \cdots \quad \mathcal{T}_p[:, (p{-}1)r{+}1:pr]\} \tag{34}$$

From Eq. (3) for the definitions of $\mathcal{T}_p$ and $\mathcal{O}_p$ and Eq. (27), one obtains

$$\begin{aligned}
\mathcal{T}_p(:, 1:r) &= \begin{bmatrix} D \\ \mathcal{U}_n[1:(p-1)m, :]B \end{bmatrix} \\
\mathcal{T}_p(:, r+1:2r) &= \begin{bmatrix} 0_{m\times r} \\ D \\ \mathcal{U}_n[1:(p-2)m, :]B \end{bmatrix} \\
\mathcal{T}_p(:, 2r+1:3r) &= \begin{bmatrix} 0_{2m\times r} \\ D \\ \mathcal{U}_n[1:(p-3)m, :]B \end{bmatrix} \\
&\vdots \\
\mathcal{T}_p[:, (p-1)r+1:pr] &= \begin{bmatrix} 0_{(p-1)m\times r} \\ D \end{bmatrix}
\end{aligned} \tag{35}$$

with $0_{i\times j}$ being a zero matrix of dimension $i \times j$. The product of $\mathcal{U}_0^T\mathcal{T}_p$ becomes

$$\begin{aligned}
\mathcal{U}_0^T\mathcal{T}_p(:, 1:r) &= \mathcal{U}_0^T(:, 1:m)D \\
&\quad + \mathcal{U}_0^T(:, m+1:pm)\mathcal{U}_n[1:(p-1)m, :]B \\
\mathcal{U}_0^T\mathcal{T}_p(:, r+1:2r) &= \mathcal{U}_0^T(:, m+1:2m)D \\
&\quad + \mathcal{U}_0^T(:, 2m+1:pm)\mathcal{U}_n[1:(p-2)m, :]B \\
\mathcal{U}_0^T\mathcal{T}_p(:, 2r+1:3r) &= \mathcal{U}_0^T(:, 2m+1:3m)D \\
&\quad + \mathcal{U}_0^T(:, 3m+1:pm)\mathcal{U}_n[1:(p-3)m, :]B \\
&\vdots \\
\mathcal{U}_0^T\mathcal{T}_p[:, (p-1)r+1:pr] &= \mathcal{U}_0^T[:, (p-1)m+1:pm]D
\end{aligned} \tag{36}$$

Equation (36) can be rewritten in the following matrix form:

$$\mathcal{U}_{0\mathcal{T}} = \mathcal{U}_{0n}\begin{bmatrix} D \\ B \end{bmatrix} \tag{37}$$

where

$$\mathcal{U}_{0\mathcal{T}} = \begin{bmatrix} \mathcal{U}_0^T\mathcal{T}_p(:, 1:r) \\ \mathcal{U}_0^T\mathcal{T}_p(:, r+1:2r) \\ \mathcal{U}_0^T\mathcal{T}_p(:, 2r+1:3r) \\ \vdots \\ \mathcal{U}_0^T\mathcal{T}_p[:, (p-1)r+1:pr] \end{bmatrix} \qquad \mathcal{U}_{0n} = \begin{bmatrix} \mathcal{U}_0^T(:, 1:m) & \mathcal{U}_0^T(:, m+1:pm)\mathcal{U}_n[1:(p-1)m, :] \\ \mathcal{U}_0^T(:, m+1:2m) & \mathcal{U}_0^T(:, 2m+1:pm)\mathcal{U}_n[1:(p-2)m, :] \\ \mathcal{U}_0^T(:, 2m+1:3m) & \mathcal{U}_0^T(:, 3m+1:pm)\mathcal{U}_n[1:(p-3)m, :] \\ \vdots & \vdots \\ \mathcal{U}_0^T[:, (p-1)m+1:pm] & 0_{n_0\times n} \end{bmatrix}$$

The dimension of $\mathcal{U}_{0\mathcal{T}}$ is $pn_0 \times pr$, whereas the dimension of $\mathcal{U}_{0n}$ is $pn_0 \times (m+n)$. Let the right-hand side of Eq. (33) be denoted by

$$\mathcal{U}_{0\mathcal{R}} = \mathcal{U}_0^T\mathcal{R}_{yu}\mathcal{R}_{uu}^{-1} \tag{38}$$

where $\mathcal{U}_{0\mathcal{R}}$ is an $n_0 \times pr$ matrix. Equation (37) shows that $\mathcal{U}_{0\mathcal{T}}$ is thus given by

$$\mathcal{U}_{0\mathcal{T}} = \begin{bmatrix} \mathcal{U}_{0\mathcal{R}}(:, 1:r) \\ \mathcal{U}_{0\mathcal{R}}(:, r+1:2r) \\ \mathcal{U}_{0\mathcal{R}}(:, 2r+1:3r) \\ \vdots \\ \mathcal{U}_{0\mathcal{R}}[:, (p-1)r+1:pr] \end{bmatrix} \tag{39}$$

and B and D can be computed by

$$\begin{bmatrix} D \\ B \end{bmatrix} = \mathcal{U}_{0n}^{\dagger}\mathcal{U}_{0\mathcal{T}} \tag{40}$$

The first m rows of $\mathcal{U}_{0n}^{\dagger}\mathcal{U}_{0\mathcal{T}}$ form the matrix D, and the last n rows produce the matrix B.

Equation (40) has a unique least-squares solution for B and D only if the matrix $\mathcal{U}_{0n}$ has more rows than columns. Because the dimension of $\mathcal{U}_{0n}$ is $pn_0 \times (m+n)$, the integer p must be chosen such that $pn_0 \geq (m+n)$, where $n_0 = pm - n$, 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_0 is $n_0 = m$. This indicates that the order of the system must be determined such that $pn_0 \geq (m+n)$ is satisfied, in particular, for the case where all of the singular values beyond σ_n [see Eq. (29)], i.e., $\sigma_{n+1}, \ldots, \sigma_{pm}$, are not exactly zeros but small quantities.

For small n_0, computing B and D from Eq. (40) is quite efficient in time. In practice, the integer n_0 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 using the matrix $\mathcal{U}_0$ associated with the zero singular values is presented in the following section.

2. *Direct Method*

Instead of using $\mathcal{U}_0$ to derive Eq. (33), the direct method depends on the observability matrix $\mathcal{O}_p$ to formulate an equation to solve for B and D. The approach used to derive the direct method is similar to that for the indirect method.

First, use the notation $X(k)$ defined in Eq. (12) and the state equation, Eq. (1), to form

$$\begin{aligned} X(k+1) &= [\boldsymbol{x}(k+1) \quad \boldsymbol{x}(k+2) \quad \cdots \quad \boldsymbol{x}(k+N)] \\ &= AX(k) + B\boldsymbol{u}'_N(k) \end{aligned} \tag{41}$$

where $\boldsymbol{u}'_N(k)$ is defined as

$$\boldsymbol{u}'_N(k) = [\boldsymbol{u}(k) \quad \boldsymbol{u}(k+1) \quad \cdots \quad \boldsymbol{u}(k+N-1)] \tag{42}$$

Substituting Eq. (41) into Eq. (11) yields

$$\begin{aligned} Y_p(k+1) &= \mathcal{O}_pX(k+1) + \mathcal{T}_pU_p(k+1) \\ &= \mathcal{O}_pAX(k) + \mathcal{O}_pBu'_N(k) + \mathcal{T}_pU_p(k+1) \\ &= \mathcal{O}_pAX(k) + (\mathcal{O}_pB \quad \mathcal{T}_p)\begin{bmatrix} u'_N(k) \\ U_p(k+1) \end{bmatrix} \\ &= \mathcal{O}_pAX(k) + (\mathcal{O}_pB \quad \mathcal{T}_p)U_{p+1}(k+1) \end{aligned} \tag{43}$$

From Eq. (11), the least-squares solution for $X(k)$ is

$$X(k) = \mathcal{O}_p^{\dagger}[Y_p(k) - \mathcal{T}_pU_p(k)] \tag{44}$$

Using Eq. (44) for $X(k)$, Eq. (43) produces

$$\begin{aligned} &Y_p(k+1) - \mathcal{O}_pA\mathcal{O}_p^{\dagger}Y_p(k) \\ &\quad = -\mathcal{O}_pA\mathcal{O}_p^{\dagger}\mathcal{T}_pU_p(k) + [\mathcal{O}_pB \quad \mathcal{T}_p]U_{p+1}(k+1) \\ &\quad = \big\{[\mathcal{O}_pB \quad \mathcal{T}_p] - \big[\mathcal{O}_pA\mathcal{O}_p^{\dagger}\mathcal{T}_p \quad 0_{pm\times r}\big]\big\}U_{p+1}(k+1) \\ &\quad = \Gamma U_{p+1}(k+1) \end{aligned} \tag{45}$$

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

$$\Gamma = [\mathcal{O}_pB \quad \mathcal{T}_p] - \big[\mathcal{O}_pA\mathcal{O}_p^{\dagger}\mathcal{T}_p \quad 0_{pm\times r}\big] \tag{46}$$

Note that $\mathcal{O}_pB$ is a $pm \times r$ matrix, whereas $\mathcal{O}_pA\mathcal{O}_p^{\dagger}\mathcal{T}_p$ is a $pm \times pr$ matrix because $\mathcal{T}_p$ is a $pm \times pr$ matrix. Because the dimensions of $\mathcal{O}_p B$ and $\mathcal{O}_p A \mathcal{O}_p^\dagger \mathcal{T}_p$ are not the same, 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 Eq. (45) by $U_{p+1}^T(k+1)$ thus results in

$$\tilde{\mathcal{R}}_{yu}[m+1:(p+1)m, :] - \mathcal{O}_p A \mathcal{O}_p^\dagger \tilde{\mathcal{R}}_{yu}(1:pm, :) = \Gamma \tilde{\mathcal{R}}_{uu} \tag{47}$$

where

$$\begin{aligned} \tilde{\mathcal{R}}_{uu} &= (1/N) U_{p+1}(k+1) U_{p+1}^T(k+1) \\ \tilde{\mathcal{R}}_{yu} &= (1/N) Y_{p+1}(k+1) U_{p+1}^T(k+1) \end{aligned} \tag{48}$$

Now premultiplying Eq. (48) by $\mathcal{O}_p^\dagger$ and postmultiplying by $\tilde{\mathcal{R}}_{uu}^{-1}$ yields

$$\begin{aligned} &\mathcal{O}_p^\dagger \tilde{\mathcal{R}}_{yu}[m+1:(p+1)m, :] \tilde{\mathcal{R}}_{uu}^{-1} \\ &\quad - A \mathcal{O}_p^\dagger \tilde{\mathcal{R}}_{yu}(1:pm, :) \tilde{\mathcal{R}}_{uu}^{-1} = \mathcal{O}_p^\dagger \Gamma \end{aligned} \tag{49}$$

where $\mathcal{O}_p^\dagger \mathcal{O}_p = I_n$ has been applied. With input and output data given and A and C determined, all quantities on the left-hand side of Eq. (49) are computable. The unknown matrices B and D are embedded in the matrix Γ. Similar to Eq. (36) for computing the product of $\mathcal{U}_0^T \mathcal{T}_p$, matrix $\mathcal{O}_p^\dagger \Gamma$ may be computed as follows:

$$\mathcal{O}_p^\dagger \Gamma = \begin{bmatrix} B & \mathcal{O}_p^\dagger \mathcal{T}_p \end{bmatrix} - \begin{bmatrix} A \mathcal{O}_p^\dagger \mathcal{T}_p & 0_{pm \times r} \end{bmatrix} \tag{50}$$

Similar to Eq. (36) with the use of the alternate expression for the $pm \times pr$ matrix $\mathcal{T}_p$ shown in Eq. (35), one obtains

$$\begin{aligned} &\mathcal{O}_p^\dagger \Gamma(:, 1:r) = -A\mathcal{O}_p^\dagger(:, 1:m) D \\ &\quad + B - A\mathcal{O}_p^\dagger(:, m+1:pm) \mathcal{O}_p[1:(p-1)m, :] B \\ &\mathcal{O}_p^\dagger \Gamma(:, r+1:2r) = \big[\mathcal{O}_p^\dagger(:, 1:m) - A\mathcal{O}_p^\dagger(:, m+1:2m)\big] D \\ &\quad + \big\{\mathcal{O}_p^\dagger(:, m+1:pm) \mathcal{O}_p[:, 1:(p-1)m] \\ &\quad - A\mathcal{O}_p^\dagger(:, 2m+1:pm) \mathcal{O}_p[1:(p-2)m, :]\big\} B \\ &\mathcal{O}_p^\dagger \Gamma(:, 2r+1:3r) = \big[\mathcal{O}_p^\dagger(:, m+1:2m) \\ &\quad - A\mathcal{O}_p^\dagger(:, 2m+1:3m)\big] D \\ &\quad + \big\{\mathcal{O}_p^\dagger(:, 2m+1:pm) \mathcal{O}_p[:, 1:(p-2)m] \\ &\quad - A\mathcal{O}_p^\dagger(:, 3m+1:pm) \mathcal{O}_p[1:(p-3)m, :]\big\} B \\ &\qquad\qquad \vdots \\ &\mathcal{O}_p^\dagger \Gamma[:, pr+1:(p+1)r] = \mathcal{O}_p^\dagger[:, (p-1)m+1:pm] D \end{aligned} \tag{51}$$

Similar to Eq. (37), Eq. (52) can be rewritten in the following matrix form:

$$\mathcal{O}_{p\Gamma} = \mathcal{O}_{pA} \begin{bmatrix} D \\ B \end{bmatrix} \tag{52}$$

where

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

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

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, whereas $\mathcal{O}_{pA}$ is a $(p+1)n \times (m+n)$ matrix. Let the left-hand side of Eq. (49) be denoted by

$$\begin{aligned} \mathcal{O}_{p\mathcal{R}} &= \mathcal{O}_p^\dagger \tilde{\mathcal{R}}_{yu}[m+1:(p+1)m, :] \tilde{\mathcal{R}}_{uu}^{-1} \\ &\quad - A\mathcal{O}_p^\dagger \tilde{\mathcal{R}}_{yu}(1:pm, :) \tilde{\mathcal{R}}_{uu}^{-1} \end{aligned} \tag{53}$$

Equation (49) implies that

$$\mathcal{O}_{p\Gamma} = \begin{bmatrix} \mathcal{O}_{p\mathcal{R}}(:, 1:r) \\ \mathcal{O}_{p\mathcal{R}}(:, r+1:2r) \\ \mathcal{O}_{p\mathcal{R}}(:, 2r+1:3r) \\ \vdots \\ \mathcal{O}_{p\mathcal{R}}[:, pr+1:(p+1)r] \end{bmatrix} \tag{54}$$

The matrices B and D can then be determined from Eq. (52) by

$$\begin{bmatrix} D \\ B \end{bmatrix} = \mathcal{O}_{pA}^\dagger \mathcal{O}_{p\Gamma} \tag{55}$$

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 (55) has a unique least-squares solution for B and D only if the matrix $\mathcal{O}_{pA}$ has more rows than columns. Because the size of $\mathcal{O}_{pA}$ is $pn \times (m+n)$, the integer p must be chosen such that $pn \geq (m+n)$, where n is the order of the system. The direct method for computing B and D is, in general, 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, there is no need to use all of 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 $m + n$. Note 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 Eqs. (37) and (52), respectively. This by no means implies that the output error between the real output and the reconstructed output is minimized. The reconstructed output here means the output time history obtained using the input time history to drive the identified system model, which is represented by the computed matrices A, B, C, and D. An alternate method is given in the following, which minimizes the output error.

3. *Output-Error Minimization Method*

The output-error minimization method starts with rearranging the output equation, Eq. (4). The rearrangement is performed to formulate a linear equation that explicitly relates the output vector to the elements of B and D. The least-squares solution for B and D

will then minimize the output error between the real output and the reconstructed output.

Use Eq. (4) with $p = N$ and $k = 0$ to obtain

$$\boldsymbol{y}_N(0) = \mathcal{O}_N \boldsymbol{x}(0) + \mathcal{T}_N \boldsymbol{u}_p(0) \tag{56}$$

The matrices B and D are embedded in the last term, $\mathcal{T}_N \boldsymbol{u}_p(0)$, on the right-hand side of Eq. (56). How to extract B and D from Eq. (56) will be shown as follows.

Let the column vectors in B and D be expressed as

$$B = [\boldsymbol{b}_1 \quad \boldsymbol{b}_2 \quad \cdots \quad \boldsymbol{b}_r] \qquad \text{and} \qquad D = [\boldsymbol{d}_1 \quad \boldsymbol{d}_2 \quad \cdots \quad \boldsymbol{d}_r] \tag{57}$$

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

$$\boldsymbol{b} = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \vdots \\ \boldsymbol{b}_r \end{bmatrix} \qquad \text{and} \qquad \boldsymbol{d} = \begin{bmatrix} \boldsymbol{d}_1 \\ \boldsymbol{d}_2 \\ \vdots \\ \boldsymbol{d}_r \end{bmatrix} \tag{58}$$

The column vector $\boldsymbol{b}$ is nothing but the result of stacking together all of the column vectors of the input matrix B whereas $\boldsymbol{d}$ includes all of the column vectors of the transmission matrix D. Similarly, let the input vector $\boldsymbol{u}(k)$ be explicitly written as

$$\boldsymbol{u}(k) = \begin{bmatrix} \boldsymbol{u}_1(k) \\ \boldsymbol{u}_2(k) \\ \vdots \\ \boldsymbol{u}_r(k) \end{bmatrix} \tag{59}$$

where the quantities $\boldsymbol{u}_i(k)$ for $i = 1, 2, \ldots, r$ are scalar, with r being the number of inputs.

Using $\boldsymbol{b}$ and $\boldsymbol{d}$ defined in Eq. (58) and $\mathcal{T}_N$ in Eq. (2), $\mathcal{T}_N u_p(0)$ may now be rewritten as

$$\mathcal{T}_N \boldsymbol{u}_p(0) =$$

$$\begin{bmatrix} D & & & & \\ CB & D & & & \\ CAB & CB & D & & \\ \vdots & \vdots & \vdots & \ddots & \\ CA^{N-2}B & CA^{N-3}B & CA^{N-4}B & \cdots & D \end{bmatrix} \begin{bmatrix} u(0) \\ u(1) \\ u(2) \\ \vdots \\ u(N-1) \end{bmatrix}$$

$$= \begin{bmatrix} \mathcal{U}_m(0) \\ \mathcal{U}_m(1) \\ \mathcal{U}_m(2) \\ \vdots \\ \mathcal{U}_m(N-1) \end{bmatrix} d + \begin{bmatrix} 0_{m \times n} \\ C\mathcal{U}_n(0) \\ CA\mathcal{U}_n(0) + C\mathcal{U}_n(1) \\ \vdots \\ \sum_{k=0}^{N-2} CA^{N-k-2}\mathcal{U}_n(k) \end{bmatrix} \boldsymbol{b} \tag{60}$$

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

$$\begin{aligned} \mathcal{U}_m(k) &= [I_m \boldsymbol{u}_1(k) \quad I_m \boldsymbol{u}_2(k) \quad \cdots \quad I_m \boldsymbol{u}_r(k)] \\ \mathcal{U}_n(k) &= [I_n \boldsymbol{u}_1(k) \quad I_n \boldsymbol{u}_2(k) \quad \cdots \quad I_n \boldsymbol{u}_r(k)] \end{aligned} \tag{61}$$

and I_m and I_n are identity matrices of order m and n, respectively. The matrix size of $\mathcal{U}_m(k)$ is $m \times mr$ and $\mathcal{U}_n(k)$ 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 as shown in Eq. (60).

Substituting Eq. (61) into Eq. (56) yields

$$\boldsymbol{y}_N(0) = \Phi\boldsymbol{\Theta} \tag{62}$$

where

$$\boldsymbol{\Theta} = \begin{bmatrix} \boldsymbol{x}(0) \\ \boldsymbol{d} \\ \boldsymbol{b} \end{bmatrix}$$

$$\Phi = \begin{bmatrix} C & \mathcal{U}_m(0) & 0_{m\times n} \\ CA & \mathcal{U}_m(1) & C\mathcal{U}_n(0) \\ CA^2 & \mathcal{U}_m(2) & CA\mathcal{U}_n(0) + C\mathcal{U}_n(1) \\ \vdots & & \\ CA^{N-1} & \mathcal{U}_m(N-1) & \sum_{k=0}^{N-2} CA^{N-k-2}\mathcal{U}_n(k) \end{bmatrix} \tag{63}$$

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

$$\boldsymbol{\Theta} = \Phi^{\dagger}\boldsymbol{y}_N(0) \tag{64}$$

where † denotes the pseudoinverse. The least-squares solution $\boldsymbol{\Theta}$ does not actually satisfy Eq. (62) when the system has input and output noises. However, it minimizes the error between the actual output vector $\boldsymbol{y}_N(0)$ and the computed output vector $\hat{\boldsymbol{y}}_N(0) = \Phi\boldsymbol{\Theta}$. Solving the least-squares solution $\boldsymbol{\Theta}$ can be very time consuming because the number of rows in Φ 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.

To this end, the SRIM algorithm has been developed. Two methods are presented for computing A and C, and three methods are shown for calculating B and D. For the reader who is interested in programming the algorithm, the computational steps are given in the following section.

IV. Computational Steps

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

1) Choose an integer p such that $p \geq (n/m) + 1$, 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 $pr \times pr$ as defined in Eq. (13) using the matrices $Y_p(k)$ of dimension $pm \times N$ and $U_p(k)$ of dimension $pr \times N$ defined in Eq. (12). 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 such that $\ell - k - p + 2 \geq N \gg \min(pm, pr)$, where ℓ is the length of the data.

3) Calculate the correlation matrix $\mathcal{R}_{hh}$ of dimension $pm \times pm$ defined in Eq. (13), i.e., $\mathcal{R}_{hh} = \mathcal{R}_{yy} - \mathcal{R}_{yu}\mathcal{R}_{uu}^{-1}\mathcal{R}_{yu}^T$.

4) Use singular value decomposition to factor $\mathcal{R}_{hh}$ for the full decomposition method [Eq. (25)] or a portion of $\mathcal{R}_{hh}$ for the partial decomposition method [Eq. (30)].

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. (25)] and $\mathcal{U}_0$ of dimension $pm \times n_0$, where $n_0 = pm - n$ is the number of truncated small singular values. The integer n_0 must satisfy the condition $pn_0 \geq (m+n)$ for the full decomposition method. For the partial decomposition method, $\mathcal{U}_n$ is replaced by $\mathcal{U}_n'$ [Eq. (30)], and the integer n_0 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 Eq. (7) to determine the state matrix A. The output matrix C is the first m rows of $\mathcal{U}_n$.

7) Compute $\mathcal{U}_{0\mathcal{R}} = \mathcal{U}_0^T \mathcal{R}_{yu}\mathcal{R}_{uu}^{-1}$ shown in Eq. (38) for the indirect method and construct $\mathcal{U}_{0n}$ and $\mathcal{U}_{0\mathcal{T}}$ shown in Eqs. (37) and (39). Determine the input matrix B and the direct transmission matrix D from Eq. (40), i.e., the first m rows of $\mathcal{U}_{0n}^{\dagger}\mathcal{U}_{0\mathcal{T}}$ form D and the last n rows produce B. For the direct method, construct $\mathcal{O}_{p\Gamma}$ and $\mathcal{O}_{pA}$ from Eq. (52) and solve for 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 D, and the last n rows produce B. For the output-error minimization method, construct $\boldsymbol{y}_N(0)$ and Φ from Eq. (63) and solve for B and D by computing $\Phi^{\dagger}\boldsymbol{y}_N(0)$. The first n elements of $\Phi^{\dagger}\boldsymbol{y}_N(0)$ form the initial state vector $\boldsymbol{x}(0)$, 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 parameter identification is desired.

9) Calculate mode singular values (see Ref. 6) to quantify and distinguish the system and noise modes. This step provides a way for model reduction using modal truncation.

The computational steps reduce to those for the ERA/DC method shown in Ref. 6 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 the 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.

The information matrix may be computed using the recursive formula presented in Appendix B of Ref. 8. The SRIM algorithm is more efficient computationally than subspace model identification (SMI) techniques.[2,3] The SMI techniques require a QR factorization of a large matrix $[U_p^T(k) \quad Y_p^T(k)]^T$ followed by a singular value decomposition and the solution of an overdetermined set of equations. Furthermore, the proposed method using the concept of data correlation permits more physical insight than the SMI techniques.

V. Illustrative Example

To illustrate the SRIM algorithm, an experimental example is given using a truss structure tested at NASA Langley Research Center. Figure 1 shows the truss structure used. The L-shaped structure consists of nine bays on its vertical section and one bay on its horizontal section, extending 90 and 20 in., respectively. The shorter section is clamped to a steel plate, which is rigidly attached to the wall. The square cross section is 10 × 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 servo-accelerometers located at a corner of the square cross section provide the in-plane tip acceleration measurements. In addition, an offset weight of 30 lb is added to enhance the dynamic coupling between the two principal axes and to lower the structure fundamental frequency. For identification, the truss is excited using 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.

Table 1 lists the modal frequencies and damping ratios identified using the partial decomposition method for determining A and C in conjunction with the output-error minimization method for computing B and D. The initial index p is arbitrarily set as shown in Table 1 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 system model to 6. The last column in Table 1 (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 using the same input signal. The error matrix has the size of $m \times \ell$, where m is the number of outputs and ℓ the length of the data. As shown in Table 1, the output error decreases continuously as p increases from 5 to 100. The speed of decreasing the output error is quite slow from $p = 50$ to 100. The frequencies identified for all different p are quite close, whereas the damping ratios range from 2.5 to 0.4% for the first mode, from 1.5 to 0.4% for the second mode, and from 1.2 to 0.07% for the third mode.

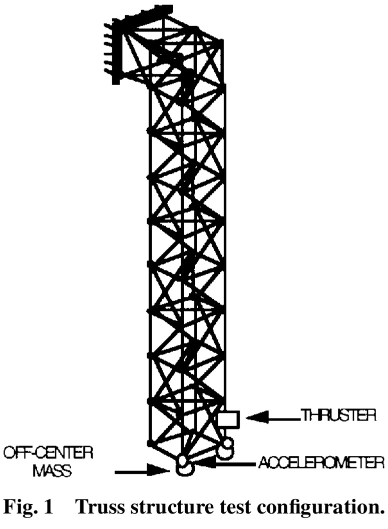

Fig. 1 Truss structure test configuration.

Table 1 Identified modal parameters obtained by the partial decomposition method with singular values truncation

	Mode 1		Mode 2		Mode 3		OEM[a]
p	Freq, Hz	Damp, %	Freq, Hz	Damp, %	Freq, Hz	Damp, %	error, max SV
5	5.89	2.48	7.29	1.54	48.5	1.19	621.77
10	5.88	2.08	7.30	0.99	48.2	0.71	545.58
15	5.88	1.10	7.30	0.51	48.0	0.99	365.04
20	5.88	0.62	7.29	0.38	47.5	2.06	264.57
25	5.87	0.46	7.29	0.42	47.4	2.44	207.63
50	5.86	0.44	7.29	0.41	48.4	0.46	174.25
100	5.85	0.43	7.28	0.43	48.6	0.07	167.64

[a] Output-error minimization method.

Table 2 Identified modal parameters obtained by the partial decomposition method with modal truncation

	Mode 1		Mode 2		Mode 3		IDM[a]	OEM[b]
p	Freq, Hz	Damp, %	Freq, Hz	Damp, %	Freq, Hz	Damp, %	error, max SV	error, max SV
5	5.89	3.50	7.28	2.30	49.0	1.13	817.99	735.99
10	5.87	0.65	7.29	0.47	48.6	0.74	262.81	202.84
15	5.85	0.40	7.28	0.41	48.6	0.46	197.32	171.33
20	5.85	0.37	7.28	0.41	48.7	0.44	216.79	174.46
25	5.85	0.38	7.28	0.42	48.7	0.64	203.56	174.06
50	5.85	0.38	7.28	0.44	48.6	0.47	198.69	175.11
100	5.85	0.40	7.28	0.45	48.5	0.30	194.58	174.02

[a] Indirect method. [b] Output-error minimization method.

Results from the indirect and direct methods for computing B and D are not shown because they produce the output errors several orders of magnitude higher than the ones shown in Table 1. As shown in Ref. 8, both methods work fine for the simulation data with input and output noises assumed to be white random, Gaussian, and zero mean. Therefore, it is believed that the noise nonlinearities are the major causes for the problem of using indirect and direct methods. From this example, one may conclude that the indirect and direct methods should not be used in practice for computing B and D if A and C are obtained by the reduced model via singular values truncation.

Table 2 lists the modal frequencies and damping ratios identified using the partial decomposition method for determining A and C with the indirect method for computing 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. The reduced model is used to compute the output error. The output error decreases quickly when p increases from 5 to 10 and reaches a minimum at $p = 15$. It increases slightly again and then reduces to another minimum at $p = 100$. However, the minimum 194.58 at $p = 100$ does not improve much from the minimum 197.32 at $p = 15$. Similar to Table 1, the frequencies identified for all different p are very close, whereas the damping ratios range from 3.5 to 0.4% for the first mode, from 2.3 to 0.45% for the second mode, and from 1.13 to 0.3% for the third mode.

The reader should not be surprised with the similarity in modal parameters in Tables 1 and 2 because both share the same observability matrix before singular values truncation. Table 1 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 2 shows the modal parameters computed from the full-size observability matrix. The modal parameters shown in Table 2 are the ones chosen to represent the system.

The question may arise as to whether the output errors in Table 2 may be reduced if B and D are recalculated via the output-error minimization method using the same A and C. The question is answered by the last column of Table 2. Indeed, the output errors are somewhat improved for all cases. The output error 171.33 for $p = 15$ in Table 2 is better than 174.25 in Table 1 at $p = 50$. This indicates that the combination of singular values truncation with the output-error minimization may not produce the global minimum for any given p. As a result, it seems clear that the modal truncation combined with the output-error minimization method is a good method to do model reduction.

VI. Concluding Remarks

The main contribution of this paper is the development of a new system realization algorithm using the information matrix as the basis for computing system matrices. The algorithm uses a data correlation matrix to compute an observability matrix via the singular value decomposition. The data correlation matrix is formed from the autocorrelation matrix of the shifted output data subtracted by the 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 and the output matrix. Two computational methods are presented including a full decomposition method and a partial decomposition method to determine the state matrix and the output matrix. The partial decomposition method seems easier to use than the full decomposition matrix because it eliminates the need for singular value truncation. In practice, there are no zero singular values regardless of how clean the data sequence is. To determine how many singular values should be truncated requires engineering judgment or special techniques such as sensitivity analysis. Based on the computed state and output matrices, three methods are described, including the indirect method, the direct method, and the output-error minimization method to compute the input matrix, the direct transmission matrix, and the initial state vector. When the input and output noises are white, Gaussian, and zero mean, any combination of these methods performs well. For other noises, any combination also works well if no singular value truncation is conducted. With singular value truncation for model reduction, the combination of partial decomposition algorithm with the output-error minimization works better than the other methods but is comparable to the modal truncation technique.

References

[1] Moonen, M., De Moor, B., Vandenberghe, L., and Vandewalle, J., “On- and Off-Line Identification of Linear State-Space Models,” *International Journal of Control,* Vol. 49, No.1, 1989, pp. 219–232.

[2] 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, 1992, pp. 1187–1210.

[3] Van Overschee P., and De Moor B., “A Unifying Theorem for Three Subspace System Identification Algorithms,” *Automatica* (Special Issue on Trends in System Identification), Vol. 31, No. 12, 1995, pp. 1853–1864.

[4] 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.

[5] 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, 1993, pp. 320–329.

[6] Juang, J.-N., *Applied System Identification*, Prentice–Hall, Englewood Cliffs, NJ, 1994, Chaps. 5 and 6.

[7] Ljung, L., *System Identification: Theory for the User*, Prentice–Hall, Englewood Cliffs, NJ, 1987, pp. 184–186.

[8] Juang, J.-N., “State-Space System Realization Using Input/Output Data Correlation,” NASA TR 17548, March 1997.