SUPERSPACE AND SUBSPACE IDENTIFICATION OF BILINEAR MODELS BY DISCRETE-LEVEL INPUTS

Minh Q. Phan*, Francesco Vicario†, Richard W. Longman‡, and Raimondo Betti§

When excited by an input consisting of a number of discrete levels, a bilinear system becomes a linear time-varying system whose dynamics switches from one linear subsystem to another depending on the input level. This paper describes an identification method that uses the concept of a superstate of a linear switching system as a superstate of the bilinear system. In a superspace method, these superstates are used directly to identify a bilinear system model. In a subspace method, two or more superstate representations are intersected to find a reduced dimension subspace prior to identification of a bilinear model.

INTRODUCTION

The research areas of linear time-varying systems and bilinear systems have seen a recent spark of interests in the aerospace engineering community, Refs. [1-9]. Although one might view bilinear systems as a bridge between linear and nonlinear systems, its importance is far greater. Whereas some dynamical systems are inherently bilinear, other nonlinear systems can be converted into bilinear form by a process known as Carleman linearization, Ref. [10]. The latter application of bilinear system is important because it offers an opportunity to present a very large class of nonlinear systems in the standard bilinear form. Finding a common (or universal) representation of nonlinear systems is indeed a worthwhile goal of nonlinear system identification. For example, Euler’s equations which describe the rotation of a rigid body, is not bilinear in the standard form because the equations are bilinear in the states, but linear in the input. However, they can be approximated by a standard bilinear state-space model, which is linear in the states, but bilinear in the states and inputs, Ref. [4].

One bilinear system identification approach involves the creation of input-output-to-state representations (IOSR), Refs. [10,11]. These IOSR’s relate the state of the bilinear system to a superstate consisting of input and output measurements and their nonlinear polynomial products. The transformation matrix that relates such a superstate to the bilinear system state is time-invariant. The main problem with this approach is that the dimension of the superstate increases very quickly as a function of the dimension of the bilinear system because of the explosion of the number of nonlinear terms in the superstate. In fact, overcoming this curse of dimensionality is the most challenging problem in discrete-time bilinear system identification today.

*Thayer School of Engineering, Dartmouth College, Hanover, NH.
†Department of Mechanical Engineering, Columbia University, New York, NY, now with Philips Research North America, Briarcliff Manor, NY.
‡Department of Mechanical Engineering, Columbia University, New York, NY.
§Department of Civil Engineering and Engineering Mechanics, Columbia University, New York, NY.
An approach to alleviate the curse of dimensionality is making the bilinear system behave as a linear time-varying system by using an input that can only take on one among a set of discrete levels at any time step, Ref [12]. The linear time-varying system is actually a linear switching system whose dynamics switches from one linear subsystem to another depending on the input level that is currently active. It is emphasized here that the discrete-level input is only used in the identification step. Once identified, the bilinear model can predict the system response correctly to any other input, including inputs that do not follow the same discrete levels.

Because each subsystem is linear, its superstate is that of linear system and involves only time-shifted input and output measurements without any nonlinear products that cause the explosion of dimensions. Although the switching subsystems are decoupled from each other in the space of the original minimum-order states, they are in fact coupled to each other in the space of the superstates, rendering the problem more challenging. Nevertheless, for each linear subsystem, the dimension of the linear superstate can be kept to the minimum. In exchange for this reduced dimension superstate, the transformation matrix that relates each subsystem superstate to the original bilinear system state is time-varying. If we further constraint the input to a small number of levels (without sacrificing richness), the number of these transformations matrices can also be kept relatively small.

It is important to realize that it is not necessary to actually find the aforementioned time-varying transformation matrices. The superstate of the switching system can be treated as a non-minimum order state of the original bilinear system to identify a bilinear state-space model directly. Model reduction is then applied to reduce the identified non-minimum order bilinear model to the minimum dimension. This is a superspace identification technique. Another strategy is to develop multiple versions of the superstates of the linear switching system (such as causal, non-causal, or mixed representations) and intersecting them to recover the state of the bilinear system before identifying its bilinear state-space model. This is a subspace technique. This present paper describes (1) a method to generate non-causal superstate that results in a minimum number of linear switching subsystems for an input with a given number of discrete levels, (2) a superspace method to identify the bilinear system directly from the causal superstate of the linear switching system, and (3) a subspace method to intersect two or more superstate representations of the linear switching system to recover the bilinear system state, and (4) numerical examples to illustrate the developed algorithms.

**PROBLEM STATEMENT**

Consider an \(n\)-state single-input \(q\)-output discrete bilinear system of the form:

\[
\begin{align*}
    x(k+1) &= Ax(k) + Bu(k) + Nx(k)u(k) \\
    y(k) &= Cx(k) + Du(k)
\end{align*}
\]  

(1)

We consider the single-input case for simplicity. The multiple-input case can be generalized later. For the current method, the bilinear system is re-written in the form of a linear time-varying system with an input-dependent system matrix,

\[
\begin{align*}
    x(k+1) &= [A + Nu(k)]x(k) + Bu(k) \\
    y(k) &= Cx(k) + Du(k)
\end{align*}
\]  

(2)

By design, the input \(u(k)\) can take on one of a fixed set of discrete levels, \(u(k) \in \{\alpha_1, \alpha_2, ..., \alpha_L\}\), and output \(y(k)\) is recorded. This restriction on the input excitation signal turns the discrete-time bilinear system into a linear switching system, \(A(k) \in \{A_1, A_2, ..., A_L\}\), where \(A_i = A + \alpha_i N, i = 1, 2, ..., L\).
Given a such an input-output data record that is sufficiently long and rich, the goal of the identification problem is to find a realization of the bilinear model denoted by $A_r, B_r, C_r, D_r, N_r$ that has the same input-output relationship as the original bilinear system. The discrete-level input is only used in the identification step. The identified bilinear model should be able to predict the system response correctly to any other input, including inputs that do not follow the same discrete levels.

**MATHEMATICAL FORMULATION**

In this section, we describe the notion of a time-varying input-output-to-state representation (IOSR) that relates the current state of the bilinear system to a data vector comprised solely of time-shifted input and output measurements. Such an IOSR can be causal or non-causal or mixed, depending on whether past, future, or a combination of measurements are involved relative to the current time of the bilinear system state.

**Single and Multiple-Step Ahead State and Output Equations**

For simplicity of presentation, we will derive the relevant expressions for a 3-state, single-input, single-output system, and then generalize the results to an $n$-state, single-input, $q$-output system. The generalization to an $r$-input system will be considered later. Starting with the input-dependent linear time-varying representation of a bilinear system, $A(k) \in \{A_1, A_2, ..., A_L\}$, where $A_i = A + \alpha_i N, i = 1, 2, ..., L$, we propagate the state equation forward three time steps,

$$x(k + 1) = A(k)x(k) + Bu(k)$$

$$x(k + 2) = A(k + 1)A(k)x(k) + [A(k + 1)B \quad B]\ \begin{bmatrix} u(k) \\ u(k + 1) \end{bmatrix}$$

$$= A(k + 1)A(k)x(k) + C(k + 1)\begin{bmatrix} u(k) \\ u(k + 1) \end{bmatrix}$$

$$x(k + 3) = A(k + 2)A(k + 1)A(k)x(k) + C(k + 2)\begin{bmatrix} u(k) \\ u(k + 1) \\ u(k + 2) \end{bmatrix}$$

where the time-varying controllability matrix $C(k + 2)$ is

$$C(k + 2) = [A(k + 2)A(k + 1)B \quad A(k + 1)B \quad B]$$

Similarly, the output equations are also propagated forward two time steps,

$$y(k) = Cx(k) + Du(k)$$

$$y(k + 1) = CA(k)x(k) + CBu(k) + Du(k + 1)$$

$$y(k + 2) = C \left( A(k + 1)A(k)x(k) + [A(k + 1)B \quad B] \begin{bmatrix} u(k) \\ u(k + 1) \end{bmatrix} \right) + Du(k + 2)$$

and then package them together in the form,

$$\begin{bmatrix} y(k) \\ y(k + 1) \\ y(k + 2) \end{bmatrix} = O(k + 1)x(k) + T(k + 1)\begin{bmatrix} u(k) \\ u(k + 1) \\ u(k + 2) \end{bmatrix}$$
\begin{align}
O(k+1) &= \begin{bmatrix}
C \\
CA(k) \\
CA(k+1)A(k)
\end{bmatrix} \\
T(k+1) &= \begin{bmatrix}
D & 0 & 0 \\
CB & D & 0 \\
CA(k+1)B & CB & D
\end{bmatrix}
\end{align} \tag{11}

We only need to propagate the output equations two time steps to make the time-varying observability matrix \(O(k+1)\) square in this case.

**A Causal IOSR By Interaction Matrix**

The first interaction matrix \(M_1\) is introduced to the \(x(k+3)\) equation,

\[
x(k+3) = A(k+2)A(k+1)A(k)x(k) + C(k+2) \begin{bmatrix} u(k) \\ u(k+1) \\ u(k+2) \end{bmatrix} \\
+ M_1 \begin{bmatrix} y(k) \\ y(k+1) \\ y(k+2) \end{bmatrix} - M_1 \begin{bmatrix} y(k) \\ y(k+1) \\ y(k+2) \end{bmatrix}
\]

Substituting Eq. (10) into the first term that involves \(M_1\) in Eq. (12) but leaving the second term that involves \(M_1\) alone produces

\[
x(k+3) = \begin{bmatrix} C(k+2) & M_1T(k+1) \end{bmatrix} \begin{bmatrix} u(k) \\ u(k+1) \\ u(k+2) \end{bmatrix} - M_1 \begin{bmatrix} y(k) \\ y(k+1) \\ y(k+2) \end{bmatrix}
\]

where interaction matrix \(M_1\) is used to remove the state-dependent term on the right hand side of Eq. () by setting

\[
A(k+2)A(k+1)A(k) + M_1O(k+1) = 0 \tag{14}
\]

It can be seen from Eq. () that \(M_1\) is time-dependent. Furthermore, for \(M_1\) to exist, \(O(k+1)\) must be invertible. This is the condition to ensure the existence of the interaction matrix \(M_1\). In the present identification method, there is no need to find \(M_1\), but only to assume its existence.

Shifting time back by two time steps and re-ordering the input elements to make the expression easier to read, we have

\[
x(k+1) = T_1 (A(k), A(k-1), A(k-2)) \begin{bmatrix} u(k) \\ u(k-1) \\ u(k-2) \\ y(k) \\ y(k-1) \\ y(k-2) \end{bmatrix}
\]

Let \(p\) denote the number of time steps used to advance the state equation to produce Eq. (). Thus the above expression for \(x(k+1)\) is derived for \(p = 3\). It can be shown that the general expression
for $x(k + 1)$ for any value of $p$ would be

$$x(k + 1) = T_f (A(k), A(k - 1), \ldots, A(k - p + 1))$$

where $k \geq p - 1$. The current state $x(k)$ expressed in terms of input-output measurements is obtained by shifting the time indices back by one time step, where $k \geq p$. In the above derivation for a 3-state 1-output system, we need $O(k + 1)$ to be at least a square 3-by-3 matrix by using $p = 3$, and furthermore require it to be full rank to ensure the existence of the interaction matrix $M_1$. It is certainly possible to use $p > 3$, in which case the corresponding observability matrix would be a tall matrix (more rows than columns). As long as this matrix is full rank, a corresponding $M_1$ is guaranteed to exist. In general, the parameter $p$ is required to be such that $pq \geq n$ where $n$ is the minimum dimension of the bilinear state, and $q$ is the number of independent outputs. The above ISOR is causal or forward-time because the current state of the bilinear system is expressed in terms of $p$ past values of input and output measurements.

In Eq. (16), the transformation matrix $T_f$ of the causal IOSR is time-varying and dependent on a number of past system matrices from $A(k - 1)$ to $A(k - p)$. Because the number of time-varying system matrices $A(k)$’s is fixed by the number of discrete input levels, the number of linear transformations $T_f$’s is fixed. For an $N$-level input, the number of linear time-varying transformation matrices $T_f$’s is at most $L^p$ for any choice of $p$ such that $pq \geq n$.

A Non-Causal IOSR by Interaction Matrix

The first interaction matrix $M_1$ is introduced to the $x(k + 3)$ equation and that leads to a causal IOSR. Now a second interaction matrix $M_2$ is introduced to the $x(k + 1)$ equation and this would lead to a non-causal IOSR. We proceed with the derivation as follows,

$$x(k + 1) = A(k)x(k) + Bu(k) + M_2 \begin{bmatrix} y(k) \\ y(k + 1) \\ y(k + 2) \end{bmatrix} - M_2 \begin{bmatrix} y(k) \\ y(k + 1) \\ y(k + 2) \end{bmatrix}$$

Substituting Eq. (16) into the first expression involving the interaction matrix $M_2$ in Eq. (16), but leaving the second expression involving $M_2$ alone and simplifying the resulting expression produces,

$$x(k + 1) = (M_2 T(k + 1) + \begin{bmatrix} B & 0 & 0 \end{bmatrix}) \begin{bmatrix} u(k) \\ u(k + 1) \\ u(k + 2) \end{bmatrix} - M_2 \begin{bmatrix} y(k) \\ y(k + 1) \\ y(k + 2) \end{bmatrix}$$

where $M_2$ is used to remove the state-dependent term on the right hand side of Eq. (16) by setting

$$A(k) + M_2 O(k + 1) = 0$$
The condition to ensure the existence of $M_2$ is the same as in the previous case. We can re-package Eq. (1) in a form that is a counterpart of Eq. (1). 

\[
x(k + 1) = T_2 (A(k), A(k + 1)) \begin{bmatrix} u(k) \\ u(k + 1) \\ u(k + 2) \\ y(k) \\ y(k + 1) \\ y(k + 2) \end{bmatrix}
\]

(19)

It is interesting to contrast the two IOSR’s given in Eq. (1) and Eq. (2). Both relate the state $x(k + 1)$ to input-output measurements. The data vector in Eq. (1) consists of the present and past input-output measurements, whereas the data vector in Eq. (2) consists of present and future input-output measurements. Therefore, we refer Eq. (1) as a non-causal or backward-time IOSR. In general, for a choice of $p$, the general expression for $x(k + 1)$ would be

\[
x(k + 1) = T_b (A(k), A(k + 1), ..., A(k + p - 2)) \begin{bmatrix} u(k) \\ u(k + 1) \\ \vdots \\ u(k + p - 1) \\ y(k) \\ y(k + 1) \\ \vdots \\ y(k + p - 1) \end{bmatrix}
\]

(20)

where $k \geq 0$ and $p \geq 2$. By shifting the time indices back by one time step, we have an expression for the current state $x(k)$. For later reference, we use the generic notation $z(k)$ to denote an input-output data vector and $T(k)$ the corresponding time-varying transformation matrix that relates the data vector back to the bilinear system state $x(k)$. Thus the shorthand notation is:

\[
x(k) = T(k)z(k)
\]

(21)

Depending on which IOSR is being considered, we have different expressions for the data vector $z(k)$ and the corresponding transformation matrix $T(k)$.

**Minimum-Dimension IOSR’s**

In this section, we call out the differences between the forward and backward time IOSR’s when $p = 1$ or 2. Selecting $p = 1$ is possible when the number of independent outputs is equal to or greater than the number of states, i.e. $C$ is tall matrix (more rows than columns) and full rank. In the forward-time case, Eq. (1) reduces to

\[
x(k + 1) = T_f (A(k)) \begin{bmatrix} u(k) \\ y(k) \end{bmatrix}
\]

(22)

which says that $T_f$ depends on $A(k)$, hence $u(k)$, only. This is indeed the case because the forward-time case (and the backward-time case), the relationship between $x(k + 1)$ and input-output measurements reduces to

\[
x(k + 1) = \begin{bmatrix} B - A(k)C^{-1}D & A(k)C^{-1} \end{bmatrix} \begin{bmatrix} u(k) \\ y(k) \end{bmatrix}
\]

(23)
In other words, when \( p = 1 \), the forward-time and backward-time ISOR’s are identical,
\[
T_f (A(k)) = T_b (A(k)) = \begin{bmatrix} B - A(k)C^{-1}D & A(k)C^{-1} \end{bmatrix}, \quad p = 1
\]
and the time-varying transformation matrix depends on \( A(k) \) or \( u(k) \) only. When \( p = 2 \), the forward-time \( T_f \) depends on \( A(k) \) and \( A(k - 1) \), but the backward-time \( T_b \) depends on \( A(k) \) only,
\[
x(k + 1) = T_f (A(k), A(k - 1)) \begin{bmatrix} u(k) \\ y(k) \\ y(k - 1) \end{bmatrix} = T_b (A(k)) \begin{bmatrix} u(k) \\ u(k + 1) \\ y(k) \\ y(k + 1) \end{bmatrix}
\]

To confirm that this is the case, we derive the backward-time IOSR for \( p = 2 \) as follows.
\[
x(k + 1) = A(k)x(k) + Bu(k) + M_2 \begin{bmatrix} y(k) \\ y(k + 1) \end{bmatrix} - M_2 \begin{bmatrix} y(k) \\ y(k + 1) \end{bmatrix}
\]

Using
\[
\begin{bmatrix} y(k) \\ y(k + 1) \end{bmatrix} = O(k)x(k) + T \begin{bmatrix} u(k) \\ u(k + 1) \end{bmatrix}
\]

where
\[
O(k) = \begin{bmatrix} C \\ CA(k) \end{bmatrix}, \quad T = \begin{bmatrix} D & 0 \\ CB & D \end{bmatrix}
\]

and imposing the condition on the interaction matrix \( M_2 \) such that \( A(k) + M_2O(k) = 0 \) leads to
\[
x(k + 1) = (M_2T + \begin{bmatrix} B & 0 \end{bmatrix}) \begin{bmatrix} u(k) \\ u(k + 1) \end{bmatrix} - M_2 \begin{bmatrix} y(k) \\ y(k + 1) \end{bmatrix}
\]

The relationship between \( x(k + 1) \) to the input-output measurements given in Eq. \( 1 \) is now verified
where \( T_b (A(k)) = \begin{bmatrix} M_2T + \begin{bmatrix} B & 0 \end{bmatrix} \\ -M_2 \end{bmatrix} \) and \( M_2 = -A(k)O(k)^{-1} \). Thus for both cases \( p = 1 \) and \( p = 2 \), the backward-time IOSR depends only on \( A(k) \). On the other hand, the forward-time IOSR depends on \( A(k) \) for \( p = 1 \), but on \( A(k) \) and \( A(k - 1) \) for \( p = 2 \).

**A Mixed Causal and Non-Causal IOSR**

Another IOSR can be derived by introducing an interaction matrix \( M_3 \) to the \( x(k + 2) \) equation and imposing the condition \( A(k + 1)A(k) + M_3O(k + 1) = 0 \) produces an expression for \( x(k + 2) \), then shifting time back by one time step,
\[
x(k + 1) = T_m (A(k - 1), A(k), ..., A(k + p - 3)) \begin{bmatrix} u(k - 1) \\ u(k) \\ \vdots \\ u(k + p - 2) \\ y(k - 1) \\ y(k) \\ \vdots \\ y(k + p - 2) \end{bmatrix}
\]

This is a mixed causal and non-causal IOSR. Other IOSR’s can be similarly derived.
SUPERSPACE IDENTIFICATION

If the bilinear system state \( x(k) \) is known, identification of the bilinear state-space matrices becomes a simple linear problem. In superspace identification, we look for a time-invariant transformation matrix \( T \) such that

\[
x(k) = Tz_s(k)
\]  

and then treat \( z_s(k) \), which is referred to as a superstate, as the state of the bilinear system for identification of a bilinear state-space model. Although the superstate \( z_s(k) \) typically has higher dimension than the true state \( x(k) \), it can still be used to find an non-minimum order model. The bilinear state-space model matrices \( \bar{A}, \bar{B}, \) and \( \bar{N} \) are solved from these linear equations,

\[
[ z_s(p) \quad z_s(p+1) \quad \cdots ] = [ \bar{A} \quad \bar{B} \quad \bar{N} ] \begin{bmatrix} z_s(p-1) & z_s(p) & \cdots \\ u(p-1) & u(p) & \cdots \\ z_s(p-1)u(p-1) & z_s(p)u(p) & \cdots \end{bmatrix}
\]  

(32)

and \( \tilde{C}, \tilde{D} \) are solved from

\[
[ y(p) \quad y(p+1) \quad \cdots ] = [ \tilde{C} \quad \tilde{D} ] \begin{bmatrix} z_s(p) & z_s(p+1) & \cdots \\ u(p) & u(p+1) & \cdots \end{bmatrix}
\]  

(33)

To obtain the final bilinear state-space model of the correct minimum order, we simply remove the unobservable portion of the identified model \( \bar{A}, \bar{B}, \bar{C}, \bar{D}, \bar{N} \) associated with the pair \( \bar{A}, \bar{C} \). Successful application of a superspace identification algorithm requires the matrices that post-multiply \( [ \bar{A} \quad \bar{B} \quad \bar{N} ] \) and \( [ \bar{C} \quad \bar{D} ] \) to be full rank (a richness condition) and at least square (a data length condition). Not every definition of the superstate \( z_s(k) \) satisfies the rank condition. Therefore, not every definition of the superstate \( z_s(k) \) can be used for superspace identification. In the following section, we describe a superstate definition that qualifies.

Superstate of a Linear Switching System

The input-output data vector \( z(k) \) that appear in different ISORs derived in the previous sections cannot be treated as a superstate \( z_s(k) \) of the bilinear system because the transformation that relates \( z(k) \) back to \( x(k) \) is time-varying, \( x(k) = T(k)z(k) \). In the current approach, however, the bilinear system becomes a linear switching system when the excitation input value is constrained to a set discrete values. For a linear switching system, there is a simple approach to create a superstate vector where the transformation matrix that relates such the superstate vector back to the state of the linear switching system is time-invariant. Let the linear switching system be denoted by

\[
x(k+1) = A(k)x(k) + Bu(k)
\]  

(34)

where \( A(k) \in \{ A_1, A_2, ..., A_N \} \). At each time step \( k \), the IOSR derivation shows that there exists a corresponding linear transformation matrix that relates a certain input-output data vector to the state of the switching system, \( x(k) = T(k)z(k) \). The specific expression of \( z(k) \) and \( T(k) \) depend on which IOSR is used. Because \( A(k) \) belongs to a set of possible system matrices, the corresponding \( T(k) \) also belongs a set of possible transformation matrices, i.e.,

\[
T(k) \in \{ T_1, T_2, ..., T_{N_s} \}
\]  

(35)
The number of transformation $N_s$ depends on the number of combinations of input levels and the parameter $p$. For example, if $p = 2$ is selected and two-discrete level input is used, $L = 2$, then

$$x(k) = T_f (A(k-1), A(k-2)) \begin{bmatrix} u(k-1) \\ u(k-2) \\ y(k-1) \\ y(k-2) \end{bmatrix}$$

Therefore, $N_s = 4$ for the forward-time IOSR corresponding to 4 possible combinations of input levels of $u(k)$ and $u(k-1)$ recognizing that $A(k)$ depends on $u(k)$ only, $A(k) = A + Nu(k)$. For $p = 2$, the “backward-time” IOSR is

$$x(k) = T_b (A(k-1)) \begin{bmatrix} u(k-1) \\ u(k) \\ y(k-1) \\ y(k) \end{bmatrix}$$

Therefore $N_s = 2$ for the “backward-time” IOSR corresponding to the 2 possible input levels of $u(k-1)$. Define a transformation matrix $T_s$ consisting of all possible transformation matrices $T_i$, where $i = 1, 2, ..., N_s$,

$$T_s = \begin{bmatrix} T_1 & T_2 & \cdots & T_{N_s} \end{bmatrix}$$

Because only one transformation matrix applies at any time step $k$, we can write,

$$x(k) = T_s z_s(k)$$

where the superstate $z_s(k)$ of the linear switching system is

$$z_s(k) = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ z(k) \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Each 0 in $z_s(k)$ is a zero-vector of the same dimension as the input-output data vector $z(k)$. The superstate $z_s(k)$ consists mostly of zeros except for a single $z(k)$ corresponding to a particular transformation matrix that applies at $k$. Note that which particular transformation applies is known because the input combination associated with it is known. Therefore the location where $z(k)$ should be located in $z_s(k)$ is known at each time step $k$. As mentioned, not every superstate definition can be used in a superset identification method. For the two definitions of $z(k)$ considered so far, only the definition associated with the forward-time or causal IOSR given in Eq. (36) can be used in superspace identification. The definition of $z(k)$ associated with the backward-time or non-causal IOSR cannot be used because it leads to rank deficiency in the matrices that need to be inverted to solve Eq. (36) and Eq. (37). This is due to the fact that $u(k)$ is involved in the definition of $z(k)$ in the backward-time or non-causal IOSR, but not in the forward-time or causal IOSR.
Reduced-Dimension Superstate

The superstate $z_s(k)$ defined in Eq. (1) consists solely of time-shifted input-output measurements and zeros. As long as the input is constrained to a fixed number of levels, the bilinear system becomes a linear switching system. The superstate $z_s(k)$ of the linear switching system is in fact a superstate of the original bilinear system. Therefore $z_s(k)$ can be used to identify a state-space model of the original bilinear system using Eq. (1) and Eq. (2). However, before doing this, it is good practice to determine if the dimension of the superstate can be reduced prior to using them in Eq. (1) and Eq. (2). To this end, form the following data matrix,

$$Z_s = \begin{bmatrix} z_s(p) & z_s(p+1) & z_s(p+2) & \cdots \end{bmatrix}$$

(41)

and check the rank of its row space. This is best performed by a singular value decomposition (SVD) of $Z_s$ where $Z_s = U_s S V_s^T$. The row space of $Z_s$ is spanned by the rows of $V_s^T$ associated with the non-zero singular values in $S$. Let $Z_{sr}$ be a matrix formed by these rows of $V_s^T$. The columns of $Z_{sr}$ can then be treated as the (possibly) reduced-dimension superstate $z_{sr}(k)$. These reduced-dimension superstates $z_{sr}(k)$ can be used in place of the original superstate $z_s(k)$ in the identification of the bilinear state-space model in Eq. (1) and Eq. (2).

Model Reduction

Regardless of whether the original superstate $z_s(k)$ or its reduced-dimension version $z_{sr}(k)$ is used, the identified bilinear state-space model derived from Eq. (1) and Eq. (2) is non-minimum order. The identified model contains an unobservable subspace which can be removed by model reduction as follows. Let $\tilde{\Theta}$ be the observability matrix associated with the pair $\bar{A}$ and $\bar{C}$,

$$\tilde{\Theta} = \begin{bmatrix} \bar{C} \\ \bar{C} \bar{A} \\ \vdots \\ \bar{C} \bar{A}^{\bar{n}-1} \end{bmatrix}$$

(42)

where $\bar{n}$ is used to denote the state dimension of $\bar{A}$. A singular value decomposition of $\tilde{\Theta}$ reveals the unobservable subspace of the identified model that can be removed. Let $V_r$ denote the matrix formed by the right singular vectors associated with $n$ non-zero singular values of $\tilde{\Theta}$. The minimum-order bilinear state-space model is simply

$$A_r = V_r^T \bar{A} V_r \quad N_r = V_r^T \bar{N} V_r \quad B_r = V_r^T \bar{B} \quad C_r = \bar{C} V_r \quad D_r = \bar{D}$$

(43)

The final bilinear state-space model has the correct minimum dimension as the original bilinear state-space model, and the same input-output relationship. It can be used to predict correctly the bilinear system response to any arbitrary input, including inputs that do not have the same discrete levels used in the identification.

SUBSPACE IDENTIFICATION

A single superstate vector such as the one given in Eq. (1) can be used in a superspace identification method. When used in conjunction with one or more other superstate definitions, a subspace identification method can be developed. The subspace method considered here involves the intersection of two or more superspaces to find the subspace where the linear switching system state
resides. The logic of a subspace intersection method goes as follows. Let \( X \) denote a matrix whose columns are the state of the bilinear (or linear switching) system,

\[
X = \begin{bmatrix}
x(p) & x(p + 1) & x(p + 2) & \cdots
\end{bmatrix}
\]  

(44)

and \( Z_{s1} \) and \( Z_{s2} \) are matrices whose columns are superstate definitions labeled 1 and 2,

\[
Z_{s1} = \begin{bmatrix}
z_{s1}(p) & z_{s1}(p + 1) & z_{s1}(p + 2) & \cdots
\end{bmatrix}
\]  

(45)

\[
Z_{s2} = \begin{bmatrix}
z_{s2}(p) & z_{s2}(p + 1) & z_{s2}(p + 2) & \cdots
\end{bmatrix}
\]  

(46)

For example, we can use the input-output data vector \( z_1(k) \) associated with the forward-time or causal IOSR to build the superstate \( z_{s1}(k) \), and the input-output data vector \( z_2(k) \) associated the backward-time or non-causal IOSR to build the superstate \( z_{s2}(k) \) in accordance with the strategy described in Eq. (47) where

\[
z_1(k) = \begin{bmatrix}
u(k - 1) \\
u(k - 2) \\
\vdots \\
u(k - p) \\
y(k - 1) \\
y(k - 2) \\
\vdots \\
y(k - p)
\end{bmatrix} \quad z_2(k) = \begin{bmatrix}
u(k + p - 2) \\
\vdots \\
u(k - 1) \\
y(k) \\
\vdots \\
y(k + p - 2)
\end{bmatrix}
\]  

(47)

Notice that the dimensions of \( z_1(k) \) and \( z_2(k) \) are the same, but the dimensions of their associated superstates \( z_{s1}(k) \) and \( z_{s2}(k) \) are different. This is due to the fact that the number of possible transformation matrices relating \( z_1(k) \) and \( z_2(k) \) back to the bilinear system state \( x(k) \) is different for the same value of \( p \). If the rows of \( X \) reside in the row space of \( Z_{s1} \) and also in the row space of \( Z_{s2} \), then the rows of \( X \) must reside in the intersection of the row spaces of \( Z_{s1} \) and \( Z_{s2} \), i.e.,

\[
X \subset Z_{s1} \quad X \subset Z_{s2} \quad \Rightarrow \quad X \subset (Z_{s1} \cap Z_{s2})
\]  

(48)

By finding the intersection subspace of \( Z_{s1} \) and \( Z_{s2} \), we have a reduced-dimension subspace for \( X \).

**Intersection Subspace**

Given \( Z_{s1} \) and \( Z_{s2} \), it is straightforward to find the row space of the intersection subspace. Let \( v_i \) denote a common row vector that lies in both the row space of \( Z_{s1} \) and the row space of \( Z_{s2} \), i.e.,

\[
v_i = a_i^T Z_{s1} = b_i^T Z_{s2}
\]  

(49)

where \( a_i \) and \( b_i \) are column vectors of the coefficients that express each \( v_i \) in terms of the rows of \( Z_{s1} \) and \( Z_{s2} \), respectively. These common row vectors \( v_i \)'s are not necessarily independent. From Eq. (47) we write \( R c_i = 0 \), where

\[
R = \begin{bmatrix}
(Z_{s1})^T \\
(Z_{s2})^T
\end{bmatrix} \quad c_i = \begin{bmatrix}
a_i \\
b_i
\end{bmatrix}
\]  

(50)

Thus the column vector \( c_i \) lies in the null space of \( R \). Because the SVD is an effective method to determine the null space of a matrix, the above observation leads to a simple algorithm to find the
basis vectors of the intersection (row) subspace of \( Z_{s1} \) and \( Z_{s2} \). First, we perform an singular value decomposition of \( R \) to find all the right singular vectors \( c_i \) associated with the zero singular values of \( R \). The transposes of these vectors \( c_i \)'s are the basis vectors that define the null space of \( R \). Let \( m \) denote the number of these vectors, hence \( i = 1, 2, ..., m \). Next, from each \( c_i \) we extract \( a_i \) and \( b_i \), and use them to determine the \( m \) common vectors \( v_i \)'s associated with the \( m \) null space basis vectors. As mentioned, these common vectors are not necessarily independent. To find the actual dimension of the intersection subspace and the orthogonal basis vectors that span it, define a matrix \( V \) formed by these common row vectors, \( V = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix} \) (51)

The row basis vectors of the intersection subspace of \( Z_{s1} \) and \( Z_{s2} \) can be found from the SVD of \( V \). They are the transposes of the right singular vectors of \( V \) associated with its non-zero singular values.

**Bilinear Model Subspace Identification**

Let \( Z_I \) denote the basis vectors that span the intersection row space of \( Z_{s1} \) and \( Z_{s2} \) obtained from the SVD of \( V \) described in the previous section. The columns of \( Z_I \) can be interpreted as the states of the bilinear (or linear switching) system, \( Z_I = [z_I(p) \ z_I(p+1) \ z_I(p+2) \ \cdots] \) (52)

A bilinear state-space model can be identified from

\[
\begin{bmatrix}
  z_I(p) & z_I(p+1) & \cdots \\
\end{bmatrix} = \begin{bmatrix}
  \tilde{A} & \tilde{B} & \tilde{N} \\
\end{bmatrix}
\begin{bmatrix}
  z_I(p-1) & z_I(p) & \cdots \\
  u(s-1) & u(p) & \cdots \\
  z_I(p-1)u(p-1) & z_I(p)u(p) & \cdots \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
  y(p) & y(p+1) & \cdots \\
\end{bmatrix} = \begin{bmatrix}
  \tilde{C} & \tilde{D} \\
\end{bmatrix}
\begin{bmatrix}
  z_I(p) & z_I(p+1) & \cdots \\
  u(p) & u(p+1) & \cdots \\
\end{bmatrix}
\]

The subscript \( I \) denotes the state obtained from the intersection subspace of \( Z_{s1} \) and \( Z_{s2} \).

The number of basis vectors which define the dimension of the row space of \( V \) is not necessarily equal to the minimum dimension of the bilinear state-space model,

\[
\text{rank}(Z_{s1} \cap Z_{s2}) = \text{rank}(Z_{s1}) + \text{rank}(Z_{s2}) - \text{rank} \left( \begin{bmatrix} Z_{s1} \\ Z_{s2} \end{bmatrix} \right) \geq n \quad (55)
\]

In other words, it is not guaranteed that the dimension of the intersection subspace is the same as the minimum dimension of the original bilinear state-space model. Therefore, the identified bilinear state-space model \( \tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}, \tilde{N} \) is not necessarily of minimum-order. When \( \text{rank}(Z_{s1} \cap Z_{s2}) > n \), model reduction needs to be performed on the identified bilinear model to find the minimum-order bilinear state-space representation. As in the superspace approach, the additional dynamics is unobservable, and can be removed to produce a final minimum-order bilinear state-space model using the same algorithm summarized in Eqs. 12-14.
IDENTIFICATION ALGORITHMS

We now summarize the superspace and subspace identification algorithms developed in this paper from the perspective of application. Although there are several bookkeeping steps involved, the overall algorithms are relatively simple.

Data Collection

The first step is to create an input excitation signal consisting of \( N \) discrete levels, one of which is 0, \( u(k) \in \{ \alpha_1, \alpha_2, ..., \alpha_L \} \). For example, a two-level input with \( \alpha_1 = 0, \alpha_2 = 1 \) corresponds to a series of “random” unit pulses with “random” durations. Such an excitation signal is sufficiently rich for bilinear system identification, Ref. [9]. A three-level input with \( \alpha_1 = 0, \alpha_2 = 1, \alpha_3 = -1 \) corresponds to a series of unit pulses that can switch signs. Excite the bilinear system with this input signal and record the resultant output measurements.

Selection of Parameter \( p \)

The parameter \( p \) controls the number of forward and/or backward time steps in the input-output data vector \( z(k) \). The parameter \( p \) must be selected such that \( pq \geq n \), where \( q \) is the number of independent outputs and \( n \) is the minimum dimension of the bilinear model to be identified. If \( p \) is larger than necessary, the final model reduction step will reveal the true minimum dimension of the bilinear system. Therefore, there is no need in theory to pick the minimum value for \( p \) although a smaller \( p \) results in less computational requirements.

If the goal of the identification problem is to approximate a nonlinear system by a bilinear model, then in addition to the actual outputs, we can create additional artificial outputs by using polynomial products of the existing outputs. For example, if there are two actual outputs, \( y_1(k) \) and \( y_2(k) \), the additional second-order outputs would be \( y_1^2(k), y_2^2(k), y_1(k)y_2(k) \). Similarly, artificial outputs of higher orders can be created. These outputs are permissible because they are in fact state elements in a bilinear model approximation of a general nonlinear system by Carleman linearization, Ref. [10]. The additional artificial outputs will improve the identification of the bilinear approximation due to improved observability. Having more outputs will allow a smaller \( p \) to be used, which in turn reduces the dimension of the superstate representation in the subsequent identification steps.

Data States and Superstates

The next step is to build state vectors \( z(k) \) associated with the derived IOSR’s. The data vector \( z(k) \) associated with the forward-time or causal IOSR is defined in Eq. ( ), and the data vector \( z(k) \) associated with the backward-time or non-causal IOSR is defined in Eq. ( ). For superspace identification, only the forward-time \( z(k) \) is needed. For subspace identification, the backward-time \( z(k) \) is also needed. Recall that the relationship between \( z(k) \) and the bilinear system state \( x(k) \) is \( x(k) = T(k)z(k) \) where depending on context, \( T(k) \) and \( z(k) \) could be the forward-time or backward-time representations.

After creating the data vector \( z(k) \) for all available time steps, we build the superstates of the linear switching system with the forward-time \( z(k) \) for the superspace method, and with the backward-time \( z(k) \) for the subspace method. We also need to know the number of possible switching transformation matrices \( T(k) \)’s and when each transformation matrix applies at each time step. Refer to Eq. ( ) for this determination in the forward-time case, and Eq. ( ) in the backward-time case.
For example, if \( p = 2 \) is used, then according to Eq. (1) and Eq. (3) the forward-time transformation matrix \( T(k) \) is dependent on \( A(k) \) and \( A(k - 1) \) which in turn is dependent on \( u(k) \) and \( u(k - 1) \).

Because the input can only take a certain number of discrete levels, the number of combinations of \( u(k) \) and \( u(k - 1) \) is fixed. For example if two-level input is used, then there are at most 4 combinations of \( u(k) \) and \( u(k - 1) \), hence there are at most \( N_s = 4 \) possible forward-time transformation matrices \( T(k) \)’s in \( x(k) = T(k)z(k) \).

Note that for \( p = 2 \), the backward-time transformation matrix \( T(k) \) is dependent on \( A(k - 1) \), hence \( u(k - 1) \), only. Therefore, for \( p = 2 \), there are only two backward-time transformation matrices \( T(k) \)’s involved in \( x(k) = T(k)z(k) \).

Recall that from \( x(k) = T(k)z(k) \) where \( T(k) \in \{T_1, T_2, ..., T_{N_s}\} \), we build the superstate \( z_s(k) \) where

\[
T_s = \begin{bmatrix}
T_1 & T_2 & \cdots & T_{N_s}
\end{bmatrix}
\tag{56}
\]

The number of possible transformation matrices \( T(k) \)’s determine the dimension of the superstate vector \( z_s(k) \). As defined in Eq. (5), a superstate consists mostly of zeros except for one \( z(k) \) corresponding the current \( T(k) \) that applies. Back to the example with \( p = 2 \) and a two-level input, at any time step \( k \), the combination \( u(k) \) and \( u(k - 1) \) is known. Therefore, it can be easily determined when the corresponding \( T_i \) applies, and hence the location of the corresponding \( z(k) \) in the superstate \( z_s(k) \).

We form all the superstate vectors \( z_s(k) \) for the forward-time and backward-time IOSR’s with the available data, denoted by \( z_{s1}(k) \) and \( z_{s2}(k) \). We then arrange them in the two matrices \( Z_{s1} \) and \( Z_{s2} \) as defined in Eqs. (1) and (4), respectively.

**Superspace Identification**

Although the columns of \( Z_{s1} \) can be treated as the states of the bilinear system as specified in Eq. (6), it is useful to reduce its dimension before doing so. The row space of \( Z_{s1} \) is spanned by the transpose of the right singular vectors associated with the non-zero singular values of \( Z_{s1} \). Let \( Z_{sr} \) be a matrix formed by these row vectors. The columns of \( Z_{sr} \) are the reduced-dimension superstate \( z_{sr}(k) \) to be used in place of the original superstate \( z_s(k) \) in the identification of the bilinear state-space model.

The bilinear state-space model matrices \( \tilde{A}, \tilde{B}, \text{ and } \tilde{N} \) are solved from

\[
\begin{bmatrix}
\tilde{A} & \tilde{B} & \tilde{N}
\end{bmatrix} = \begin{bmatrix}
z_{sr}(p) & z_{sr}(p+1) & \cdots
\end{bmatrix} \begin{bmatrix}
z_{sr}(p-1) & z_{sr}(p) & \cdots
\end{bmatrix}^+
\]

and \( \tilde{C}, \tilde{D} \) are solved from

\[
\begin{bmatrix}
\tilde{C} & \tilde{D}
\end{bmatrix} = \begin{bmatrix}
y(p) & y(p+1) & \cdots
\end{bmatrix} \begin{bmatrix}
z_{sr}(p) & z_{sr}(p+1) & \cdots
\end{bmatrix}^+
\tag{58}
\]

where the \( + \) denotes the pseudo-inverse operation via the SVD. If the input is sufficiently rich, then the matrices whose pseudo-inverses need to be computed are full rank. This is a richness condition required for successful identification by this particular superspace method.

**Subspace Identification**

Subspace identification requires the intersection of two superspaces \( Z_{s1} \) and \( Z_{s2} \). The procedure to find their intersection subspace is summarized here.
First, we perform the singular value decomposition of $R$ from $Z_{s1}$ and $Z_{s2}$ as specified in Eq. () to find all the right singular vectors $c_i$ associated with the zero singular values of $R$. From each $c_i$, we extract $a_i$ and $b_i$, and use them to determine the $m$ common vectors $v_i$’s associated with the $m$ null space basis vectors. These common vectors are not necessarily independent. To find the actual dimension of the intersection subspace and the orthogonal basis vectors that span it, define a matrix $V$ formed by these common row vectors $v_i$’s. The row basis vectors of the intersection row space of $Z_{s1}$ and $Z_{s2}$ can be found from the SVD of $V$. They are the transposes of the right singular vectors of $V$ associated with its non-zero singular values.

Let $Z_I$ denote the basis vectors that span the intersection row space of $Z_{s1}$ and $Z_{s2}$ described in the previous paragraph. The columns of $Z_I$ can be interpreted as states of the bilinear (or linear switching) system. A bilinear state-space model can be identified from

\[
\begin{bmatrix}
\tilde{A} & \tilde{B} & \tilde{N}
\end{bmatrix} = \begin{bmatrix}
z_I(p) & z_I(p+1) & \cdots
\end{bmatrix} \begin{bmatrix}
z_I(p-1) & z_I(p) & \cdots
\end{bmatrix}^+ (59)
\]

\[
\begin{bmatrix}
\tilde{C} & \tilde{D}
\end{bmatrix} = \begin{bmatrix}
y(p) & y(p+1) & \cdots
\end{bmatrix} \begin{bmatrix}
z_I(p) & z_I(p+1) & \cdots
\end{bmatrix}^+ (60)
\]

where the + again denotes the pseudo-inverse operation via the SVD. If the input is sufficiently rich, then the matrices that the pseudo-inverse operates on are full rank. This is the richness condition required for successful identification by this particular subspace method.

Model Reduction

In both superspace and subspace methods presented here, the identified state-space model obtained from Eqs. ()-() or Eqs. ()-() are not necessarily of minimum-order. Model reduction is required to reveal the final minimum order of the system. In both cases, the over-parametrized models contain unobservable dynamics that can be removed by model reduction. Equations ()-() provide all the necessary model reduction equations for the model identified by the superspace method. The same equations can be applied to the model identified by the subspace method.

Generalization to Multiple-Input Bilinear Systems

An $n$-state, $r$-input, $q$-output discrete-time bilinear state-space model has the form:

\[
x(k+1) = Ax(k) + Bu(k) + \sum_{i=1}^{r} N_i x(k) u_i(k) \\
y(k) = Cx(k) + Du(k)
\]

(61)

The state equation can be re-written in the form of an input-dependent system matrix as

\[
x(k+1) = \left( A + \sum_{i=1}^{r} N_i u_i(k) \right) x(k) + Bu(k)
\]

(62)

To convert the bilinear system to a linear switching system, let each scalar input $u_i(k)$ assume one among a fixed set of discrete levels, $u_i(k) \in \{ \alpha_1, \alpha_2, \ldots, \alpha_L \}, i = 1, 2, \ldots, r$. In this arrangement,
has up to \( L^r \) possibilities. For a single-input system, \( A(k) \) depends on the single input \( u(k) \) alone and has only \( L \) possibilities. Mathematically, the formulation can be generalized to multiple-input systems, but significant computational requirements would be incurred. For this reason, it is desirable to keep \( p \) small by increasing the number of outputs.

**ILLUSTRATIVE EXAMPLES**

Let the truth bilinear system be

\[
A = \begin{bmatrix} 0 & 0.5 \\ 0.5 & -0.5 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad N = \begin{bmatrix} 0.3 & 1 \\ -1 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 1 \end{bmatrix}, \quad D = 1
\]

We use a two-level input where \( u(k) \in \{0, 0.1\} \) to create a 100-time step input-output data record from an arbitrary non-zero initial condition \( x(0) \) which is not known to the identification algorithms (Figure 1). Because this is a two-state system \((n = 2)\) with one output \((q = 1)\), it is required that \( p \geq 2 \). We use \( p = 2 \) in this example. The two \( 4 \times 1 \) data vector \( z(k) \)'s where \( x(k) = T(k)z(k) \) for the forward-time and backward-time IOSR's are labeled \( z_f(k) \) and \( z_b(k) \), respectively,

\[
z_f(k) = \begin{bmatrix} u(k-1) \\ u(k-2) \\ y(k-1) \\ y(k-2) \end{bmatrix}, \quad z_b(k) = \begin{bmatrix} u(k-1) \\ u(k) \\ y(k-1) \\ y(k) \end{bmatrix}
\]

Similarly, the corresponding transformation matrices \( T(k) \)'s are labeled \( T_f(k) \) and \( T_b(k) \). At each time step \( k \), the forward-time transformation matrix \( T_f(k) \) depends on \( u(k-1) \) and \( u(k-2) \). Because each input can only assume one of two levels, there are 4 possible combinations of \( u(k-1) \) and \( u(k-2) \), hence a total of 4 \( T_f(k) \)'s. The backward-time transformation matrix \( T_b(k) \) depends on \( u(k-1) \) alone, therefore there are only 2 \( T_b(k) \)'s. The forward-time superstate consists of four \( 4 \times 1 \) blocks, and the backward-time superstate two \( 4 \times 1 \) blocks. Their dimensions are \( 16 \times 1 \) and \( 8 \times 1 \), respectively.

**Superspace Identification**

To create the superstate vector \( z_s(k) \), it is useful to think in terms of a time-invariant transformation matrix consisting of 4 possible \( T_f(k) \)'s as \( T_1, T_2, T_3, T_4 \) although we don’t need to find what they are. The order of these 4 transformation matrices is not important as long as the superstate vectors match a particular order. We might choose \( T_1 \) to correspond to the pair \( [u(k-1), u(k-2)] = [0, 0], T_2 \) to the pair \( [u(k-1), u(k-2)] = [0, 0.1], \) etc. At each time step \( k \), a superstate vector associated with the forward-time IOSR is created. The superstate vector consists of zeros except for a single \( z_f(k) \) correctly placed within it. For example, if at time step \( k \), if \( [u(k-1), u(k-2)] = [0, 0.1] \), then \( z_f(k) \) would be placed in the second \( 4 \times 1 \) block of the \( 16 \times 1 \) superstate vector.

The next step is to determine the row rank of a matrix whose columns are the superstate \( z_s(k) \) at all time steps. It is found that the rank of this matrix is 11. This means that the dimension of
the superstate $z_{s}(k)$ can be reduced from 16 to 11. The reduced dimension superstate $z_{sr}(k)$ is then used to find a 11-state bilinear state-space model. The matrix that needs to be inverted to find $[\bar{A}, \bar{B}, \bar{N}]$ is a $23 \times 96$ full-rank matrix. At minimum, it has to be square and full rank. Thus this 100-sample data record is more than sufficient in length and richness (theoretically speaking). The final step is model reduction. The observability matrix formed from the identified $\bar{A}$ and $\bar{C}$ has rank 2. Thus the final identified bilinear model dimension is 2 which is the correct dimension of the truth bilinear state-space model. To test the validity of the identified model, a random test input is applied to the original system and also to the final 2-state identified bilinear model using zero initial conditions. Figure 2 shows a comparison between the actual output by the truth model and the predicted output by the identified model. Their difference is of the order $10^{-14}$ confirming that the identified bilinear model is exact.

**Subspace Identification**

In the same manner, the superstate vector associated with the backward-time IOSR is created. Because the transformation matrix $T(k) = T_{b}(k)$ depends only on $u(k-1)$, it is quite simple to build the superstate vector because there are only two possible $T_{b}(k)$’s. Again the order of these 2 transformations is not important as long as the $8 \times 1$ superstate vector is formed correctly to match it. For example, we let $T_{1}$ correspond to $u(k-1) = 0$ and $T_{2}$ to $u(k-1) = 0.1$. At any time step $k$, we simply check the value of $u(k-1)$ to determine which $4 \times 1$ block it should contain $z(k)$ among the two $4 \times 1$ blocks that define the backward-time IOSR superstate vector.
After forming the two matrices of the forward-time and backward-time superstates, denoted by \( Z_{s1} \) and \( Z_{s2} \), the next step is to find the intersection subspace. Recall that \( Z_{s1} \) has 16 rows and \( Z_{s2} \) has 8 rows. An SVD of the combined matrix \( R = \left[ Z_{s1}^T \quad Z_{s2}^T \right] \), which has a total of 16 + 8 = 24 columns, reveals that there are 11 zero singular values from which 11 common vectors \( v_i \)'s are produced. Another SVD of the matrix formed by these common row vectors reveal the actual dimension of the intersection subspace which is 5. The columns of a matrix formed by these basis vectors are then treated as the bilinear system states for identification to find a 5-state bilinear model. The final model reduction step reveals the true dimension of the bilinear system which is 2. Figure 3 shows a comparison between the actual output by the truth model and the predicted output by the identified bilinear model using a random test input. Their difference is again of the order \( 1e^{-14} \) confirming that the identified bilinear model is exact.

**CONCLUDING REMARKS**

We have presented an overall approach to identify a discrete bilinear system as a linear switching system by constraining the input such that at any time step it can only assume one out of a fixed number of discrete levels. The constraint on the input is only for the purpose of identification. The final identified model predicts the response of the bilinear system correctly to any valid arbitrary input. This paper complements a similar development reported in Ref. [12]. The linear switching model strategy overcomes the high dimensionality associated with input-output-to-state relationships (IOSR’s) that are developed earlier, Refs. [9], [11]. By converting the bilinear problem into a linear switching problem, it is possible to express the state of the bilinear system solely in
Figure 3. Actual and Predicted Output and Prediction Error (Subspace Method)

terms of time-shifted input-output measurements (and a lot of zeros) without involving any nonlinear products which cause the dimension of the superstate to explode in earlier IOSR’s. There are several such superstate representations, including causal, non-causal, or mixed representations. In this paper we focus on two specific representations which are causal and non-causal. The superstate created by the causal representation can be used directly to identify a bilinear state-space model of the system, leading to a superspace method. By intersection the spaces of the causal and non-causal superstates, we can also determine a subspace where the bilinear state resides, and use it to identify a bilinear state-space model. This is therefore an intersection subspace method. The paper puts the two methods under one common framework and significantly expands the way the overall bilinear system identification landscape is viewed.

Although the dynamics of certain physical systems is bilinear, a longer term goal of bilinear system identification is to model non-linear dynamical systems. This approach is entirely consistent with an important result in Carleman linearization which states that a general nonlinear system can be represented in the form of a bilinear system, Ref. [10]. In the field of nonlinear system identification, this feature has not been significantly exploited due to the fact that bilinear system identification has not fully matured. Indeed, expressing general nonlinear systems in a common and standard form is valuable in its own right. The lack of a generic form or structure is a major source of difficulties in nonlinear system identification. Generic forms that have been considered to date include Volterra series, various feedforward and recurrent artificial neural networks, etc., and they all have their respective strength and weakness. We believe that the bilinear representation will potentially emerge as a powerful representation for the stated identification objective.
When used in conjunction with bilinear state estimation and bilinear control theory, it will offer a viable unifying framework for nonlinear system identification, estimation, and control. Although the linear time-varying or linear switching approach has gone a long way to alleviate the computational requirements in discrete bilinear system identification, there are still much work to be done. Finding the simplest and the most numerically efficient discrete-time bilinear state-space system identification methods for high dimensional systems remains an active research topic.

REFERENCES


