A New Subspace Identification Method for Bilinear Systems

Huixin Chen and **Jan Maciejowski** [*]

Department of Engineering
University of Cambridge
Cambridge CB2 1PZ U.K.
Revised 23 April 2001

Abstract

In this paper, asymptotically unbiased subspace algorithms for the identification of bilinear systems are developed. Two three-block subspace algorithms are developed for the deterministic system case and two four-block ones for the combined deterministic-stochastic case. The input signal to the system does not have to be white, which is a major advantage over an existing subspace method for bilinear systems. All the identification algorithms give asymptotically unbiased estimates with general inputs, and the rate of reduction of bias with block size is estimated. Simulation results show that the new algorithms converge much more rapidly (with sample size) than existing methods, and hence are more effective with small sample sizes. The faster convergence is presumably due to the insensitivity of the algorithms to the sample-spectrum of the input signal. These advantages are achieved by a new arrangement of the input-output equations into 'blocks', and projections onto different spaces than the ones used in earlier methods. A further advantage of our algorithms is that the dimensions of the matrices involved are significantly smaller, so that the computational complexity is lower, though still large.

Keywords:

System identification, Bilinear system, Subspace method.

[*]Corresponding author. Tel. +44 1223 332732 Fax. +44 1223 332662 Email: jmm@eng.cam.ac.uk

1 Introduction

Bilinear systems are attractive models for many dynamical processes, because they allow a significantly larger class of behaviours than linear systems, yet retain a rich theory which is closely related to the familiar theory of linear systems [15, 8]. They exhibit phenomena encountered in many engineering systems, such as amplitude-dependent time constants. Many practical system models are bilinear, and more general nonlinear systems can often be approximated well by bilinear models [17].

Most studies of the identification problem of bilinear systems have assumed an input-output formulation. Standard methods such as recursive least squares, extended least squares, recursive instrumental variable and recursive prediction error algorithms, have been applied to identifying bilinear systems. Simulation studies have been undertaken [14], and some statistical results (strong consistency and parameter estimate convergence rates) are also available [7].

In this paper, we consider the identification of MIMO bilinear systems in state-space form. There are many advantages of using state-space models, particularly in the multivariable case [5]. In recent years 'subspace' methods have been developed which have proved to be extremely effective for the identification of linear systems [6, 18, 20, 22]. In Favoreel *et al* [10, 11, 13] an extension of such methods was given for bilinear systems, but the algorithm presented there is effective only if the measured input signal to the system being identified is white. To our knowledge this was the first extension of the subspace approach to bilinear systems. In [12] another subspace algorithm for bilinear systems was presented by the same authors, which apparently does not require a white input signal. However Verdult and Verhaegen [24] pointed out that this algorithm gives biased results, and proposed an alternative algorithm, which involved a nonlinear optimization step.

In this paper alternative subspace algorithms for identifying bilinear systems are proposed. They do not require the measured input to be white, and the matrices which need to be constructed and operated upon are much smaller than those which appear in [12, 13]. Simulations show that they work well when the input signal is not white; they also show that if the input signal is white, then good results are obtained with much smaller sample sizes than are required for the algorithm of [12, 13].

The paper is organised as follows. Section 2 defines the problems that we consider, including assumptions and admissible solution algorithms. Section 3 introduces a considerable, but apparently unavoidable, amount of notation. Section 4 contains four lemmas which together are the key to the algorithms developed later in the paper. These lemmas concern firstly a rewriting of the system equations in 'block-form', that is involving contiguous blocks (across time) of input, state, and output variables; this is a standard first step in all subspace algorithms, and the only difference here from [10] is that we include the proof for the stochastic case. Secondly, these lemmas present approximate linear relationships between state sequences and input-output data sequences, and quantify the approximation involved. Section 5 then develops new 'three-block' subspace identification algorithms for the deterministic case. These algorithms make use of 'past', 'current' and 'future' data, and in that sense are 'three-block'. Algorithm I makes more complete use of the bilinear structure of the system, and is hence expected to be more accurate than Algorithm II, but at the expense of greater computational complexity. Section 6 develops corresponding algorithms (III and IV) for the stochastic case. These are 'four-block' algorithms, the additional block of 'remote future' data being required to allow the stochastic noise effects to be averaged out — in this case the 'current' data block can be regarded as providing a kind of 'instrumental variable'. Section 7 contains several examples which demonstrate that the algorithms presented here do indeed work with non-white input signals, and that they require much smaller sample sizes to obtain comparable results, than the algorithms presented in [10, 12], when the input *is* white.

2 Problem set-up

In this paper we consider deterministic time-invariant bilinear systems of the form:

$$\begin{aligned} x_{t+1} &= Ax_t + N(u_t \otimes x_t) + Bu_t \\ y_t &= Cx_t + Du_t \end{aligned} \tag{1}$$

and combined deterministic-stochastic time-invariant bilinear systems of the form:

$$\begin{aligned} x_{t+1} &= Ax_t + N(u_t \otimes x_t) + Bu_t + w_t \\ y_t &= Cx_t + Du_t + v_t \end{aligned} \tag{2}$$

where $x_t \in \mathbf{R}^n$ is the state, $u_t \in \mathbf{R}^m$ is a measured input, and $y_t \in \mathbf{R}^l$ is a measured output, $N = [N_1\ N_2\ ...N_m] \in \mathbf{R}^{n \times nm}$, and $N_i \in \mathbf{R}^{n \times n}$ $(i = 1, ..., m)$.

We assume that the sample size is $\tilde{N}$, namely that input-output data $\{u(t), y(t) : t = 0, 1, \dots, \tilde{N}-1\}$ are available.

The system input, output, state, and noise sequences, $\{u_t\}, \{y_t\}, \{x_t\}, \{w_t\}, \{v_t\}$ are assumed to be realisations of stationary stochastic processes $\mathbf{u_t}, \mathbf{y_t}, \mathbf{x_t}, \mathbf{w_t}, \mathbf{v_t}$. In fact we assume slightly more than this, that these processes are ergodic [19]. In particular we assume that for any two processes $\mathbf{a_s}$ and $\mathbf{b_t}$, with (finite segments of) realisations $(a_s, a_{s+1}, \dots, a_{N+s-1})$ and $(b_t, b_{t+1}, \dots, b_{N+t-1})$, respectively, and with $a_s \in R^m$, $b_t \in R^\ell$,

$$\lim_{N\to\infty} \frac{1}{N} \sum_{i=0}^{N-1} a_{s+i} b_{t+i}^T = E[\mathbf{a_s} \mathbf{b_t}^T] \qquad w.p.1 \tag{3}$$

The input process $\mathbf{u_t}$ is assumed to be independent of the processes $\mathbf{w_t}$ and $\mathbf{v_t}$, and the joint covariance matrix of $\mathbf{w_t}$ and $\mathbf{v_t}$ is assumed to be:

$$E\left[\begin{pmatrix} \mathbf{w_s} \\ \mathbf{v_s} \end{pmatrix} \begin{pmatrix} \mathbf{w_t} \\ \mathbf{v_t} \end{pmatrix}^T\right] = \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \delta_{st} \geq 0$$

For linear systems the ergodicity of the output would be equivalent to ergodicity of the input and noise processes, together with stability of the system. But for bilinear systems the equivalence is not so complete. We need the following assumption, which is a kind of stability condition:

$$\lambda = \max_t \overline{\sigma}(A + \Sigma_{i=1}^n u_{t,i} N_i) < 1, \qquad (0 \leq t \leq \tilde{N} - 1) \tag{4}$$

where $u_{t,i}$ denotes the i'th element of u_t and $\overline{\sigma}(.)$ denotes the greatest singular value of a matrix. Note that this assumption is additional to the assumption of ergodicity; it is a sufficient condition for our theorems to hold, but probably not necessary for the algorithms to work.

Our objective is to estimate the state dimension n, the system matrices A, B, C, D, N, and possibly the covariance matrices Q, R, S, from the input-output data. As with linear systems, the state coordinate transformation $z = Tx$ leaves the input-output relation unchanged, if T is invertible; this is seen most easily by rewriting the state transition equation in (1) in the form

$$x_{t+1} = Ax_t + \sum_{i=1}^{m} (N_i x_t) u_{t,i} + Bu_t$$

Since no particular state coordinates are specified, the system and covariance matrices can only be estimated up to such a transformation. In section 7 we shall judge the success of estimation by comparing the eigenvalues of the 'true' and estimated A and N_i matrices (which remain invariant under such transformations).

As usual in system identification, we shall need some assumption that the input excites the system sufficiently. In this paper the exact assumption needed depends on the particular algorithm being considered, and appears in slightly different form in conditions (25), (37), (47), (58), (71) and (83). Each of these is a condition on the rank of a matrix constructed from the input-output data.

We restrict ourselves to certain kinds of solution algorithms. The objective of our research is to find non-iterative methods, based on linear algebra, which are analogous to those used in subspace methods for the identification of linear systems.

3 Definitions and Notations

The use of much specialised notation seems to be unavoidable in the current context. Mostly we follow the notation used in [12, 3], but we introduce all the notation here for completeness.

For arbitrary t we define

$$X_t \triangleq [x_t \; x_{t+1} \; ... \; x_{t+j-1}] \in \mathbf{R}^{n \times j}$$

but for the special cases $t = 0, t = k, t = 2k$ and $t = 3k$ we define, with some abuse of notation,

$$\begin{aligned}
X_p &\triangleq X_0 = [x_0 \; x_1 \; ... \; x_{j-1}] \in \mathbf{R}^{n \times j} \\
X_c &\triangleq X_k = [x_k \; x_{k+1} \; ... \; x_{k+j-1}] \in \mathbf{R}^{n \times j} \\
X_f &\triangleq X_{2k} = [x_{2k} \; x_{2k+1} \; ... \; x_{2k+j-1}] \in \mathbf{R}^{n \times j} \\
X_r &\triangleq X_{3k} = [x_{3k} \; x_{3k+1} \; ... \; x_{3k+j-1}] \in \mathbf{R}^{n \times j}
\end{aligned}$$

where k is the *row block size*. The suffices p, c, f and r are supposed to be mnemonic, representing 'past', 'current', 'future' and 'remote future' respectively. We define U_t, U_p, U_c, U_f, U_r, Y_t, Y_p, Y_c, Y_f, Y_r, W_t, W_p, W_c, W_f, W_r, V_t, V_p, V_c, V_f, V_r, similarly. These matrices will later be used to construct larger matrices with a 'generalised block-Hankel' structure. In order to use all the available data in these, the number of columns j is such that $\tilde{N} = 2k + j - 1$ in the 'three-block' case (section 5) and $\tilde{N} = 3k + j - 1$ in the 'four-block' case (section 6).

We use $\otimes$ to denote the Kronecker product and $\odot$ the Khatri-Rao product of two matrices with $F \in \mathbf{R}^{q \times p}$ and $G \in \mathbf{R}^{r \times p}$, as defined in [16, 21]: $F \odot G \triangleq [f_1 \otimes g_1, f_2 \otimes g_2, \ldots, f_p \otimes g_p] \in \mathbf{R}^{qr \times p}$.

Lemma 1 *For F, G, H, J of compatible dimensions, $F \in \mathbf{R}^{k \times l}$, $G \in \mathbf{R}^{l \times m}$, $H \in \mathbf{R}^{p \times l}$, $J \in \mathbf{R}^{l \times m}$:*

$$(FG \otimes HJ) = (F \otimes H)(G \otimes J) \tag{5}$$

$$(FG \odot HJ) = (F \otimes H)(G \odot J) \tag{6}$$

Proof: see appendix.

The following integers will be useful for denoting the dimensions of various matrices. (Recall that m is the number of inputs and l the number of outputs.)

$$\begin{aligned}
d_i &= \Sigma_{p=1}^{i} (m+1)^{p-1} l \\
e_i &= \Sigma_{p=1}^{i} (m+1)^{p-1} m
\end{aligned}$$

In the following definitions, notation of the form $M_{i|j}$ $(i \geq j)$ denotes that the matrix $M_{i|j}$ contains input-output (and in a few cases, state) data blocks starting at times $j, j+1, \ldots, i$. Note that many of these definitions are recursive, and consequently that the row-dimensions of some of the defined matrices grow very quickly with the 'row block size' k.

For arbitrary $q \geq 0$ and $i \geq q+2$, we define

$$
\begin{aligned}
X_{q|q} &\triangleq \begin{pmatrix} X_q \\ U_q \odot X_q \end{pmatrix} \in \mathbf{R}^{(m+1)n \times j} \\
X_{i-1|q} &\triangleq \begin{pmatrix} X_{i-2|q} \\ U_{i-1} \odot X_{i-2|q} \end{pmatrix} \in \mathbf{R}^{(m+1)^{i-q} n \times j} \\
U_{q|q} &\triangleq U_q \in \mathbf{R}^{m \times j} \\
U_{i-1|q} &\triangleq \begin{pmatrix} U_{i-1} \\ U_{i-2|q} \\ U_{i-1} \odot U_{i-2|q} \end{pmatrix} \in \mathbf{R}^{e_{i-q} \times j} \\
Y_{q|q} &\triangleq Y_q \in \mathbf{R}^{l \times j} \\
Y_{i-1|q} &\triangleq \begin{pmatrix} Y_{i-1} \\ Y_{i-2|q} \\ U_{i-1} \odot Y_{i-2|q} \end{pmatrix} \in \mathbf{R}^{d_{i-q} \times j} \\
U^+_{q|q} &\triangleq U_q \in \mathbf{R}^{m \times j} \\
U^+_{i-1|q} &\triangleq \begin{pmatrix} U^+_{i-2|q} \\ U_{i-1} \\ U_{i-1} \odot U^+_{i-2|q} \end{pmatrix} \in \mathbf{R}^{e_{i-q} \times j}
\end{aligned}
$$

In the following definition the notation (derived from Matlab) $U_{q,(i:j)}$ is used to denote the sub-matrix of U_q formed by rows i to j (inclusive), and $U_{q,j}$ denotes the j'th row of the matrix U_q.

$$
\begin{aligned}
U^{++}_{q|q} &\triangleq \begin{pmatrix} U_{q,1} \odot U_q \\ U_{q,2} \odot U_{q,(2:m)} \\ U_{q,3} \odot U_{q,(3:m)} \\ \vdots \\ U_{q,m} \odot U_{q,m} \end{pmatrix} \in \mathbf{R}^{\frac{m(m+1)}{2} \times j} \\
U^{++}_{i-1|q} &\triangleq \begin{pmatrix} U^{++}_{i-2|q} \\ U_{i-1} \odot U^{++}_{i-2|q} \end{pmatrix} \in \mathbf{R}^{\frac{m}{2}(m+1)^{i-q} \times j}
\end{aligned}
$$

$$
\begin{aligned}
X^p &\triangleq X_{k-1|0},\ X^c \triangleq X_{2k-1|k},\ X^f \triangleq X_{3k-1|2k},\ X^r \triangleq X_{4k-1|3k}\\
U^p &\triangleq U_{k-1|0},\ U^c \triangleq U_{2k-1|k},\ U^f \triangleq U_{3k-1|2k},\ U^r \triangleq U_{4k-1|3k}\\
U^{+p} &\triangleq U^+_{k-1|0},\ U^{+c} \triangleq U^+_{2k-1|k},\ U^{+f} \triangleq U^+_{3k-1|2k},\ U^{+r} \triangleq U^+_{4k-1|3k}\\
U^{p,y} &\triangleq U^{+p} \odot Y_p,\ U^{c,y} \triangleq U^{+c} \odot Y_c,\ U^{f,y} \triangleq U^{+f} \odot Y_f,\ U^{r,y} \triangleq U^{+r} \odot Y_r,\\
U^{c,u} &\triangleq \begin{pmatrix} U^c \\ U^{+c} \odot U^p \end{pmatrix},\ U^{f,u} \triangleq \begin{pmatrix} U^f \\ U^{+f} \odot U^c \end{pmatrix},\ U^{r,u} \triangleq \begin{pmatrix} U^r \\ U^{+r} \odot U^f \end{pmatrix}\\
U^{++p} &\triangleq U^{++}_{k-1|0},\ U^{++c} \triangleq U^{++}_{2k-1|k},\ U^{++f} \triangleq U^{++}_{3k-1|2k},\ U^{++r} \triangleq U^{++}_{4k-1|3k}\\
U^{c,u,y} &\triangleq \begin{pmatrix} U^c \\ U^{++c} \\ U^{c,y} \\ U^{+c} \odot U^p \end{pmatrix},\ U^{f,u,y} \triangleq \begin{pmatrix} U^f \\ U^{++f} \\ U^{f,y} \\ U^{+f} \odot U^c \end{pmatrix},\ U^{r,u,y} \triangleq \begin{pmatrix} U^r \\ U^{++r} \\ U^{r,y} \\ U^{+r} \odot U^f \end{pmatrix},\\
\tilde{U}^{p,u,y} &\triangleq \begin{pmatrix} U^p \\ U^{++p} \\ U^{p,y} \end{pmatrix},\ \tilde{U}^{c,u,y} \triangleq \begin{pmatrix} U^c \\ U^{++c} \\ U^{c,y} \end{pmatrix},\ \tilde{U}^{f,u,y} \triangleq \begin{pmatrix} U^f \\ U^{++f} \\ U^{f,y} \end{pmatrix},\ \tilde{U}^{r,u,y} \triangleq \begin{pmatrix} U^r \\ U^{++r} \\ U^{r,y} \end{pmatrix}
\end{aligned}
$$

Y^p, Y^c, Y^f, Y^r, W^p, W^c, W^f, W^r, V^p, V^c, V^f, V^r, $U_{i-1|q}$, $W_{i-1|q}$ and $V_{i-1|q}$ are defined similarly.

Row space: Finally, we denote by $\mathcal{A}$ the space spanned by the rows of the matrix A. That is, if $A \in \mathbf{R}^{m\times n}$ then $\mathcal{A} = \text{span}\{\alpha^T A, \quad \alpha \in \mathbf{R}^m\}$. Similarly $\mathcal{U}_p$, $\mathcal{U}_c$, $\mathcal{U}_f$, $\mathcal{U}_r$, $\mathcal{Y}_p$, $\mathcal{Y}_c$, $\mathcal{Y}_f$, $\mathcal{Y}_r$, $\mathcal{U}^p$, $\mathcal{Y}^p$, $\mathcal{U}^f$, $\mathcal{Y}^f$, $\tilde{\mathcal{U}}^{p,u,y}$, $\tilde{\mathcal{U}}^{f,u,y}$ and $\mathcal{U}^{r,u}$ etc are defined as the spaces spanned by the rows of the corresponding matrices.

Remark 1. The meaning of $U^+_{i-1|q}$ is different from that in [10]. $U^{++}_{i-1|q}$, $U^{c,u}$ etc are newly introduced in this paper.

As usual, $+, \oplus$ and $\cap$ will denote the sum, the direct sum and the intersection of two vector spaces, $\cdot^\perp$ will denote the orthogonal complement of a subspace with respect to the predefined ambient space, $\setminus$ will denote set difference, the Moore-Penrose inverse will be written as $\cdot^\dagger$, and the Hermitian as $\cdot^*$.

Let $\mathcal{U}^+_{i+q|q} \odot \mathcal{U}_q$, $\mathcal{U}^{++}_{i+q|q}$, $\mathcal{U}^+_{i+q|q+1} \odot \mathcal{U}_q$ and $\mathcal{U}_{i+q|q}$ be the spaces spanned by the rows of the matrices $U^+_{i+q|q} \odot U_q$, $U^{++}_{i+q|q}$, $U^+_{i+q|q+1} \odot U_q$, and $U_{i+q|q}$, respectively. Then we have the following useful lemmas.

Lemma 2 *For any integer $i \geq 1$ and $q \geq 0$,*

$$
(\mathcal{U}^+_{i+q|q} \odot \mathcal{U}_q) \setminus \mathcal{U}^{++}_{i+q|q} = \mathcal{U}^+_{i+q|q+1} \odot \mathcal{U}_q \subset \mathcal{U}_{i+q|q}
$$

Proof: see appendix.

Lemma 3 *For $j \geq 0$, and row block size k, we have*

$$\mathcal{X}_{k-1+j|j} = \begin{pmatrix} \mathcal{X}_j \\ \mathcal{U}^{+}_{k-1+j|j} \odot \mathcal{X}_j \end{pmatrix}$$

Proof: see appendix.

4 Block-form equations

In this section we state four key lemmas which allow us to develop subspace-like identification algorithms in the following sections.

Lemma 4 *The system (2) can be rewritten in the following 'block form':*

$$\begin{aligned} X_{t+1} &= AX_t + N(U_t \odot X_t) + BU_t + W_t \\ Y_t &= CX_t + DU_t + V_t \end{aligned} \tag{7}$$

Proof: The Lemma follows immediately from the structure of X_t, Y_t, U_t, W_t, V_t, and the definition of the operators $\otimes$ and $\odot$. ∎

Lemma 5 (Input-Output Equation) *For system (2), we have, for all $k > 0, q \geq 0$:*

$$X_{k+q} = \overline{A}_k X_{k+q-1|q} + \triangle^U_k U_{k+q-1|q} + \triangle^W_k W_{k+q-1|q} \tag{8}$$

$$Y_{k+q-1|q} = \mathcal{L}^X_k X_{k+q-1|q} + \mathcal{L}^U_k U_{k+q-1|q} + \mathcal{L}^W_k W_{k+q-1|q} + \mathcal{L}^V_k V_{k+q-1|q} \tag{9}$$

where

$$\begin{aligned}
\overline{A}_k &\triangleq [A\overline{A}_{k-1}, N_1\overline{A}_{k-1}, \dots, N_m\overline{A}_{k-1}] \\
\overline{A}_1 &\triangleq [A, N_1, \dots, N_m] \\
\triangle^U_k &\triangleq \begin{bmatrix} B & A\triangle^U_{k-1} & N_1\triangle^U_{k-1} \dots, & N_m\triangle^U_{k-1} \end{bmatrix} \\
\triangle^W_k &\triangleq [I_{k\times k}, A\triangle^W_{k-1}, N_1\triangle^W_{k-1}, \dots, N_m\triangle^W_{k-1}] \\
\mathcal{L}^X_k &\triangleq \begin{bmatrix} C\triangle^X_{k-1} & 0 & \dots & 0 \\ \mathcal{L}^X_{k-1} & 0 & \dots & 0 \\ 0 & \mathcal{L}^X_{k-1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \vdots & \mathcal{L}^X_{k-1} \end{bmatrix} \\
\mathcal{L}^U_k &\triangleq \begin{bmatrix} D & C\triangle^U_{k-1} & 0 & \dots & 0 \\ 0 & \mathcal{L}^U_{k-1} & 0 & \dots & 0 \\ 0 & 0 & \mathcal{L}^U_{k-1} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \vdots & \mathcal{L}^U_{k-1} \end{bmatrix}
\end{aligned}$$

$$
\mathcal{L}_k^W \triangleq \begin{bmatrix} 0 & C\triangle_{k-1}^W & 0 & \dots & 0 \\ 0 & \mathcal{L}_{k-1}^W & 0 & \dots & 0 \\ 0 & 0 & \mathcal{L}_{k-1}^W & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \vdots & \mathcal{L}_{k-1}^W \end{bmatrix}
$$

$$
\mathcal{L}_k^V \triangleq \begin{bmatrix} I_l & 0 & \dots & 0 \\ 0 & \mathcal{L}_{k-1}^V & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathcal{L}_{k-1}^V \end{bmatrix}
$$

with

$$
\triangle_1^U \triangleq B, \quad \triangle_1^W \triangleq I_k, \quad \mathcal{L}_1^X \triangleq [C, 0_{l\times mn}], \quad \mathcal{L}_1^U \triangleq D, \quad \mathcal{L}_1^W \triangleq 0_{l\times n}, \quad \mathcal{L}_1^V \triangleq I_l
$$

Proof: For the special (deterministic) case $w_t = 0, v_t = 0$, the proof is already given in [10]. Here we give the proof for the general case.

The proof is by induction. For $k = 1$,

$$
\begin{aligned}
X_{1+q} &= AX_q + N(U_q \odot X_q) + BU_q + W_q \\
&= (A, N)\begin{pmatrix} X_q \\ U_q \odot X_q \end{pmatrix} + BU_q + W_q \\
&= \overline{A}_1 X_{q|q} + \triangle_1^U U_{q|q} + \triangle_1^W W_{q|q}
\end{aligned}
$$

$$
Y_{q|q} = Y_q = CX_q + DU_q + V_q = [C\ 0]\begin{pmatrix} X_q \\ U_q \odot X_q \end{pmatrix} + DU_{q|q} + V_{q|q}
$$

Hence both (8) and (9) hold for $k = 1$. We assume that (8), (9) hold for $k = M$. Suppose that $k = M + 1$. Then

$$
\begin{aligned}
X_{M+1+q} &= AX_{M+q} + N(U_{M+q} \odot X_{M+q}) + BU_{M+q} + W_{M+q} \\
&= A(\overline{A}_M X_{M-1+q|q} + \triangle_M^U U_{M-1+q|q} + \triangle_M^W W_{M-1+q|q}) \\
&\quad + N(U_{M+q} \odot (\overline{A}_M X_{M-1+q|q} + \triangle_M^U U_{M-1+q|q} + \triangle_M^W W_{M-1+q|q})) + BU_{M+q} + W_{M+q} \\
&= A(\overline{A}_M X_{M-1+q|q} + N(I \otimes \overline{A}_M)(U_{M+q} \odot X_{M-1+q|q})) \\
&\quad + A(\triangle_M^U U_{M-1+q|q} + N(I \otimes \triangle_M^U)(U_{M+q} \odot U_{M-1+q|q})) + BU_{M+q} \\
&\quad + A(\triangle_M^W W_{M-1+q|q} + N(I \otimes \triangle_M^W)(U_{M+q} \odot W_{M-1+q|q})) + W_{M+q} \qquad (10)
\end{aligned}
$$

Notice that $N(I \otimes \overline{A}_M) = [N_1\overline{A}_M, N_2\overline{A}_M, ..., N_m\overline{A}_M]$, $N(I \otimes \triangle_M^U) = [N_1\triangle_M^U, N_2\triangle_M^U, ..., N_m\triangle_M^U]$ and $N(I \otimes \triangle_M^W) = [N_1\triangle_M^W, N_2\triangle_M^W, ..., N_m\triangle_M^W]$. From (10), we have that (8) holds for $k = M + 1$.

$$
Y_{M+q|q} = \begin{pmatrix} Y_{M+q} \\ Y_{M-1+q|q} \\ U_{M+q} \odot Y_{M-1+q|q} \end{pmatrix}
$$

$$
\begin{aligned}
Y_{M+q} &= CX_{M+q} + DU_{M+q} + V_{M+q} \\
&= C(\overline{A}_M X_{M-1+q|q} + \triangle_M^U U_{M-1+q|q} + \triangle_M^W W_{M-1+q|q}) + DU_{M+q} + V_{M+q} \\
&= C\overline{A}_M X_{M-1+q|q} + C\triangle_M^U U_{M-1+q|q} + C\triangle_M^W W_{M-1+q|0} + DU_{M+q} + V_{M+q}
\end{aligned}
$$

$$
Y_{M-1+q|q} = \mathcal{L}_M^X X_{M-1+q|q} + \mathcal{L}_M^U U_{M-1+q|q} + \mathcal{L}_M^W W_{M-1+q|q} + \mathcal{L}_M^V V_{M-1+q|q}
$$

$$
\begin{aligned}
U_{M+q} \odot Y_{M-1+q|q} &= U_{M+q} \odot (\mathcal{L}_M^X X_{M-1+q|q} + \mathcal{L}_M^U U_{M-1+q|q} \\
&\quad + \mathcal{L}_M^W W_{M-1+q|q} + \mathcal{L}_M^V V_{M-1+q|q}) \\
&= (I \otimes \mathcal{L}_M^X)(U_{M+q} \odot X_{M-1+q|q}) + (I \otimes \mathcal{L}_M^U)(U_{M+q} \odot U_{M-1+q|q}) \\
&\quad + (I \otimes \mathcal{L}_M^W)(U_{M+q} \odot W_{M-1+q|q}) + (I \otimes \mathcal{L}_M^V)(U_{M+q} \odot V_{M-1+q|q})
\end{aligned}
$$

so equation (9) holds for $k = M + 1$.

Hence (8) and (9) hold. This proves Lemma 5. ∎

Lemma 6 *For the system (2), if condition (4) holds, then*

$$
X_c = \triangle_k^U U^p + \triangle_k^W W^p + \varepsilon(\lambda^{k-1}) \tag{11}
$$

$$
X_f = \triangle_k^U U^c + \triangle_k^W W^c + \varepsilon(\lambda^{k-1}) \tag{12}
$$

$$
X_r = \triangle_k^U U^f + \triangle_k^W W^f + \varepsilon(\lambda^{k-1}) \tag{13}
$$

where $\varepsilon(\lambda^{k-1})$ denotes a matrix such that $\|\varepsilon(\lambda^{k-1})\|_1 = o(\lambda^{k-1})$, and k is the block size.

Proof: For system (2), for $i \geq k-1$, we have:

$$
\begin{aligned}
x_{i+1} &= [\prod_{t=0}^{i}(A + \sum_{j=1}^{n} u_{t,j}N_j)]x_0 + \sum_{l=0}^{i-1}[\prod_{s=l+1}^{i}(A + \sum_{j=1}^{n} u_{s,j}N_j)](Bu_l + w_l) + Bu_i + w_i \\
&= [\prod_{t=0}^{i}(A + \sum_{j=1}^{n} u_{t,j}N_j)]x_0 + \sum_{l=i-k+1}^{i-1}[\prod_{s=l+1}^{i}(A + \sum_{j=1}^{n} u_{s,j}N_j)](Bu_l + w_l) + Bu_i + w_i \\
&\quad + \sum_{l=0}^{i-k}[\prod_{s=l+1}^{i}(A + \sum_{j=1}^{n} u_{s,j}N_j)](Bu_l + w_l)
\end{aligned}
$$

where $\prod_{i=1}^{n} A_i = A_n A_{n-1} \ldots A_1$.

For any $q \geq 0, i > q+1$, we define

$$
\begin{aligned}
u_{q|q} &\triangleq u_q \\
u_{i-1|q} &\triangleq \begin{pmatrix} u_{i-1} \\ u_{i-2|q} \\ u_{i-1} \otimes u_{i-2|q} \end{pmatrix}
\end{aligned}
$$

and prove the fact that for any $i \geq k$,

$$\triangle_k^U u_{i|i-k+1} = \sum_{l=i-k+1}^{i-1} [\prod_{s=l+1}^{i} (A + \sum_{j=1}^{n} u_{s,j} N_j)] B u_l + B u_i \tag{14}$$

We prove equation (14) by induction. First, for $k = 1$, $\triangle_1^U = B, u_{i|i} = u_i$ and the right hand side of equation (14) is Bu_i, so (14) holds for $k = 1$.

Suppose that (14) holds for $k = N$, then for $k = N + 1$,

$$\begin{aligned}
\triangle_{N+1}^U u_{i|i-N} &= [B, A\triangle_N^U, N_1\triangle_N^U, ..., N_m\triangle_N^U] \begin{pmatrix} u_i \\ u_{i-1|i-N} \\ u_i \otimes u_{i-1|i-N} \end{pmatrix} \\
&= Bu_i + A\triangle_N^U u_{i-1|i-N} + [N_1\triangle_N^U, ..., N_m\triangle_N^U](u_i \otimes u_{i-1|i-N}) \\
&= Bu_i + (A + \sum_{j=1}^{n} u_{i,j} N_j)(\triangle_N^U u_{i-1|i-N}) \\
&= Bu_i + (A + \sum_{j=1}^{n} u_{i,j})(\sum_{l=i-N}^{i-2} [\prod_{s=l+1}^{i-1} (A + \sum_{j=1}^{n} u_{s,j} N_j)] B u_l + B u_{i-1}) \\
&= \sum_{l=i-N}^{i-1} [\prod_{s=l+1}^{i} (A + \sum_{j=1}^{n} u_{s,j} N_j)] B u_l + B u_i
\end{aligned}$$

So equation (14) holds.

Similarly,

$$\triangle_k^U w_{i|i-k+1} = \sum_{l=i-k+1}^{i-1} [\prod_{s=l+1}^{i} (A + \sum_{j=1}^{n} u_{s,j} N_j)] w_l + w_i \tag{15}$$

Hence

$$\begin{aligned}
x_{i+1} - \triangle_k^U u_{i|i-k+1} - \triangle_k^W w_{i|i-k+1} &= x_{i+1} - \sum_{l=i-k+1}^{i-1} [\prod_{s=l+1}^{i} (A + \sum_{j=1}^{n} u_{s,j} N_j)](B u_l + w_l) - B u_i - w_i \\
&= [\prod_{t=0}^{i} (A + \sum_{j=1}^{n} u_{t,j} N_j)] x_0 \\
&\quad + \sum_{l=0}^{i-k} [\prod_{s=l+1}^{i} (A + \sum_{j=1}^{n} u_{s,j} N_j)](B u_l + w_l)
\end{aligned}$$

and

$$\begin{aligned}
\|x_{i+1} - \triangle_k^U u_{i|i-k+1} - \triangle_k^W w_{i|i-k+1}\|_2 &= \|[\prod_{t=0}^{i} (A + \sum_{j=1}^{n} u_{t,j} N_j)] x_0 \\
&\quad + \sum_{l=0}^{i-k} [\prod_{s=l+1}^{i} (A + \sum_{j=1}^{n} u_{s,j} N_j)](B u_l + w_l)\|_2 \\
&\leq L \sum_{t=k}^{i} \lambda^t = o(\lambda^{k-1})
\end{aligned}$$

where $L = \max_t(\|Bu_t + w_t\|_2, \|x_0\|_2)$, and (4) has been used in the last step.

Taking $i = k-1$, we have $X_c = X_k = [x_k, x_{k+1}, ..., x_{k+j-1}]$, $U^p = U_{k-1|0} = [u_{k-1|0}, u_{k|1}, ..., u_{k+j-2|j-1}]$, and $W^p = W_{k-1|0} = [w_{k-1|0}, w_{k|1}, ..., w_{k+j-2|j-1}]$. Hence (11) is proved. (12) and (13) are proved similarly, by taking $i = 2k-1$ and $i = 3k-1$, respectively. Hence Lemma 6 is proved. ∎

Lemma 7 *For the system (2), if condition (4) holds, then*

$$\begin{aligned} X_c &= E(Y_c - DU_c - V_c) + (I - EC)\triangle_k^U U^p + (I - EC)\triangle_k^W W^p + \varepsilon(\lambda^{k-1}) \\ X_f &= E(Y_f - DU_f - V_f) + (I - EC)\triangle_k^U U^c + (I - EC)\triangle_k^W W^c + \varepsilon(\lambda^{k-1}) \\ X_r &= E(Y_r - DU_r - V_r) + (I - EC)\triangle_k^U U^f + (I - EC)\triangle_k^W W^f + \varepsilon(\lambda^{k-1}) \end{aligned} \tag{16}$$

for any matrix E of compatible dimensions.

Proof: We start from the identities:

$$X_c = ECX_c + (I - EC)X_c$$
$$X_f = ECX_f + (I - EC)X_f$$
$$X_r = ECX_r + (I - EC)X_r$$

But from (2) we have:

$$CX_c = Y_c - DU_c - V_c$$
$$CX_f = Y_f - DU_f - V_f$$
$$CX_r = Y_r - DU_r - V_r$$

Now applying Lemma 6, Lemma 7 follows immediately. ∎

Remark 2 Note that Lemmas 4 – 7 specialise to the deterministic case (1) in the obvious way, by setting $W_t = 0$, $V_t = 0$, etc.

Remark 3 If $E = 0$, then Lemma 6 is a special case of Lemma 7.

Remark 4 If $l \geq n$, and $rank(C) = n$, then $I_n - C^\dagger C = 0$, where $C^\dagger$ denotes the pseudo-inverse of C. Consequently, setting $E = C^\dagger$ in Lemma 7 makes expressions (16) exact; this follows directly from Lemma 4:

$$\begin{aligned} X_c &= C^\dagger(Y_c - DU_c - V_c) \\ X_f &= C^\dagger(Y_f - DU_f - V_f) \\ X_r &= C^\dagger(Y_r - DU_r - V_r) \end{aligned} \tag{17}$$

Remark 5 Lemma 7 generalises the approach followed in [23], where a decomposition of the form $MX_t = M_1(Y_t - DU_t) + M_2\tilde{C}X_t$ is obtained.

5 The deterministic case

In this section, two 'three-block' subspace algorithms for the identification of deterministic bilinear systems are developed.

We now introduce the symbol Π to denote orthogonal projection. The orthogonal projection of the rows of a matrix A onto the row space of matrix B will be denoted by $\Pi_{\mathcal{B}} A$. It is defined by

$$\Pi_{\mathcal{B}} A \triangleq AB^T(BB^T)^{\dagger}B \tag{18}$$

Lemma 8 *Let X, Y, M, P be such that $X = PY + M$, where Y has full row rank, δ is a small positive number and $\|M\|_1 = o(\delta)$. Define $\mathcal{Y}$ as the space spanned by the rows of Y, then*

$$\|\Pi_{\mathcal{Y}^{\perp}} X\|_1 = o(\delta) \tag{19}$$

Proof: See Appendix.

5.1 Algorithm I

The key to algorithm I is the use of Lemma 7 to linearise the bilinear system. The data equations, and the state equation linking current and future data, can be written as shown in Theorem 1. We refer to this as a 'three-block' form of the equations, because data from 'past', 'current' and 'future' blocks are used (as can be seen from the definitions given in section 3). Most subspace algorithms for linear systems are based on a 'two-block' form of the equations at this point. The inspiration for the 'three-block' form is taken from [6], in which a third block is used to estimate some initial Markov parameters, which are then used in the estimation of the system matrices. In the bilinear case the introduction of a third block allows the estimation, in Step 1 of Algorithm I, of the matrix $\mathcal{D}_k$ which appears in Theorem 1. In Algorithm II this is replaced by the estimation of the matrix $\mathcal{D}_{k,2}$ ($\mathcal{D}_{k,1}$) which appears in Theorem 3 (Theorem 5). (In the linear case each of these matrices would simplify to a Toeplitz matrix containing Markov parameters.) Step 2 of each algorithm then uses this estimate to estimate two consecutive state sequences, and these are used in Step 3 to estimate the system matrices.

Theorem 1 (Three Block Form 1) *The system (1) can be written in the following form, if condition (4) holds:*

$$Y^c = \mathcal{C}_k X_c + \mathcal{D}_k U^{c,u,y} + \varepsilon(\lambda^{k-1}) \tag{20}$$
$$Y^f = \mathcal{C}_k X_f + \mathcal{D}_k U^{f,u,y} + \varepsilon(\lambda^{k-1}) \tag{21}$$
$$X_f = \mathcal{A}_k X_c + \mathcal{B}_k U^{c,u,y} + \varepsilon(\lambda^{k-1}) \tag{22}$$

where $\mathcal{A}_k, \mathcal{B}_k, \mathcal{C}_k$ and $\mathcal{D}_k$ are system-dependent constant matrices.

Proof: Recall that $Y^c = Y_{2k-1|k}$. From Lemmas 5 and 3 (specialised to the deterministic case), we know that

$$\begin{aligned} Y^c &= \mathcal{L}_k^X X^c + \mathcal{L}_k^U U^c \\ &= \mathcal{L}_k^X \begin{pmatrix} X_c \\ U^{+c} \odot X_c \end{pmatrix} + \mathcal{L}_k^U U^c \end{aligned} \tag{23}$$

Let $\mathcal{L}_k^X = \left[\mathcal{C}_k, \mathcal{L}_{k,2}^X\right]$, where $\mathcal{C}_k$ is the first n columns of the matrix $\mathcal{L}_k^X$, and $\mathcal{L}_{k,2}^X$ is the last $\left[(m+1)^k - 1\right] n$ columns of $\mathcal{L}_k^X$. Then equation (23) can be written as follows

$$\begin{aligned} Y^c &= \mathcal{C}_k X_c + \mathcal{L}_{k,2}^X (U^{+c} \odot X_c) + \mathcal{L}_k^U U^c \\ &= \mathcal{C}_k X_c + \mathcal{L}_{k,2}^X \Big(U^{+c} \odot [C^\dagger (Y_c - DU_c) + (I - C^\dagger C) X_c]\Big) + \mathcal{L}_k^U U^c \quad \text{(by Lemma 4)} \\ &= \mathcal{C}_k X_c + \mathcal{L}_{k,2}^X \Big(U^{+c} \odot \Big[C^\dagger Y_c - C^\dagger D U_c + (I - C^\dagger C)(\triangle_k^U U^p + \varepsilon(\lambda^{k-1}))\Big]\Big) + \mathcal{L}_k^U U^c \\ &\qquad \text{(by Lemma 7)} \\ &= \mathcal{C}_k X_c + \mathcal{L}_{k,2}^X \Big((I \otimes C^\dagger)(U^{+c} \odot Y_c) - (I \otimes C^\dagger D)(U^{+c} \odot U_c) + [I \otimes (I - C^\dagger C)\triangle_k^U](U^{+c} \odot U^p)\Big) \\ &\quad + \mathcal{L}_k^U U^c + \varepsilon_1(\lambda^{k-1}) \qquad \text{(by Lemma 1)} \end{aligned} \tag{24}$$

Note that the second and third terms of equation (24) form a linear combination of vectors in the row space of the matrices $\mathcal{U}^{+c} \odot \mathcal{Y}_c$, $\mathcal{U}^{+c} \odot \mathcal{U}_c$, $\mathcal{U}^{+c} \odot \mathcal{U}^p$ and $\mathcal{U}^c$. We divide $(\mathcal{U}^{+c} \odot \mathcal{U}_c)$ into two parts, namely $\mathcal{U}^{++c} \oplus [(\mathcal{U}^{+c} \odot \mathcal{U}_c) \setminus \mathcal{U}^{++c}]$ and according to Lemma 2, we know that $[(\mathcal{U}^{+c} \odot \mathcal{U}_c \setminus \mathcal{U}^{++c})] = \mathcal{U}^{+}_{2k-1|k+1} \odot \mathcal{U}_k \subset \mathcal{U}^c$. Hence there exists a matrix $\mathcal{D}_k$ such that

$$\mathcal{D}_k U^{c,u,y} = \mathcal{L}_{k,2}^X \Big((I \otimes C^\dagger)(U^{+c} \odot Y_c) - (I \otimes C^\dagger D)(U^{++c}) + [I \otimes (I - C^\dagger C)\triangle_k^U](U^{+c} \odot U^p)\Big) + \mathcal{L}_k^U U^c$$

so equation (20) holds. (21) and (22) of Theorem 1 can be proved similarly. ∎

From Theorem 1 we deduce that the block data matrices Y^c and Y^f are asymptotically linearly related to the state block matrices X_c and X_f. Also X_f is asymptotically linearly related to X_c. This is achieved by putting all the bilinear terms of the system into the data matrices $U^{c,u,y}$ and $U^{f,u,y}$ (which are defined in this paper for the first time). Note that the system equations have now been written in a quasi-linear form. This is the key to lifting the restrictions on the input which were required by earlier algorithms. As a result of this quasi-linear form, we do not require the input to be white, or even have zero mean value. In contrast with equations (20)–(22), equations (3.32)–(3.34) in [9] contain input-state products such as $U_p \odot X_p$, and the 'whiteness' of the input is required precisely in order to deal with such products.

Theorem 2 *If the pair (A, C) in (1) is observable, condition (4) holds, and*

$$\begin{pmatrix} Y^c \\ U^{c,u,y} \\ U^{f,u,y} \end{pmatrix} \text{ has full row rank,} \tag{25}$$

then, denoting $\mathcal{S} := \mathcal{Y}^c + \mathcal{U}^{c,u,y}$,

$$\|\Pi_{\mathcal{S}^\perp} Y^f - \mathcal{D}_k \Pi_{\mathcal{S}^\perp} U^{f,u,y}\|_1 = o(\lambda^{k-1}) \tag{26}$$

Proof: Substituting (22) into (21), we have

$$Y^f = \mathcal{C}_k(\mathcal{A}_k X_c + \mathcal{B}_k U^{c,u,y}) + \mathcal{D}_k U^{f,u,y} + \varepsilon_1(\lambda^{k-1}) \tag{27}$$

Since the linear part of the system is observable, then from (20), we have

$$X_c = \mathcal{C}_k^\dagger (Y^c - \mathcal{D}_k U^{c,u,y}) + \varepsilon_2(\lambda^{k-1}) \tag{28}$$

Substituting (28) into (27) shows that there exist two matrices C_1, C_2 such that

$$Y^f - \mathcal{D}_k U^{f,u,y} = C_1 Y^c + C_2 U^{c,u,y} + \varepsilon_3(\lambda^{k-1}) \tag{29}$$

Since $\mathcal{S} = \mathcal{Y}^c + \mathcal{U}^{c,u,y}$ we have, from Lemma 8:

$$\|\Pi_{\mathcal{S}^\perp} Y^f - \mathcal{D}_k \Pi_{\mathcal{S}^\perp} U^{f,u,y}\|_1 = o(\lambda^{k-1})$$

This proves Theorem 2. ∎

Remark 6: It is well known that the quality of a model obtained from an identification experiment depends on the degree of excitation of the input signal. Condition (25) of Theorem 2 is a kind of 'persistent excitation' condition. It guarantees that the matrix $\Pi_{\mathcal{S}^\perp} U^{f,u,y}$ has full row rank, and therefore $(\Pi_{\mathcal{S}^\perp} U^{f,u,y})(\Pi_{\mathcal{S}^\perp} U^{f,u,y})^\dagger = I$.

Algorithm I:

Step 1. On the basis of (26), estimate $\mathcal{D}_k$ as:

$$\hat{\mathcal{D}}_k = (\Pi_{\mathcal{S}^\perp} Y^f)(\Pi_{\mathcal{S}^\perp} U^{f,u,y})^\dagger \tag{30}$$

Step 2. Choose a threshold τ. Obtain the following SVD decomposition and partition as

$$\left[\Pi_{\mathcal{S}^+} Y_{3k-1|2k} \; \Pi_{\mathcal{S}^{++}} Y_{3k|2k+1}\right] - \hat{\mathcal{D}}_k \left[U^{u,y}_{3k-1|2k} \; U^{u,y}_{3k|2k+1}\right] =: \Gamma\Sigma\Omega^* = \begin{bmatrix} \Gamma_1 & \Gamma_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} \Omega_1^* \\ \Omega_2^* \end{bmatrix}$$

where $\mathcal{S}^+ = \mathcal{S} + \mathcal{U}^{f,u,y}$, $\mathcal{S}^{++} := \mathcal{Y}_{2k|k+1} + \mathcal{U}_{2k|k+1} + \mathcal{U}_{3k|2k+1}$ and $\|\Sigma_2\| < \tau$.

Since we expect from (20) and (21) that $\mathrm{rank}(\Sigma_1) = n$ and $\mathrm{rank}(\Sigma_2) = 0$, and hence that

$$\Gamma\Sigma\Omega^* = \Gamma_1\Sigma_1\Omega_1^* = \mathcal{C}_k \left[X_{2k} \quad X_{2k+1}\right], \tag{31}$$

form the estimates $\hat{\mathcal{C}}_k = \Gamma_1 \Sigma_1^{1/2}$ and $[\hat{X}_{2k} \quad \hat{X}_{2k+1}] = \Sigma_1^{1/2}\Omega_1^T$. ($\hat{\mathcal{C}}_k$ is not needed later.)

Step 3. Estimate the parameters A, B, C, D, N on the basis of equation (7), by solving the equation

$$\begin{bmatrix} \hat{X}_{2k+1} \\ Y_{2k} \end{bmatrix} = \begin{bmatrix} A & N & B \\ C & 0 & D \end{bmatrix} \begin{bmatrix} \hat{X}_{2k} \\ U_{2k} \odot \hat{X}_{2k} \\ U_{2k} \end{bmatrix} \tag{32}$$

in a least-squares sense.

Remark 7 Other estimates could be obtained by using other right-inverses in steps 1 and 3, and another factorisation in step 2. In [12] it is suggested that constrained least-squares could be used in step 3, because of the known structure of the solution. Our experience to date is that this does not have much effect on the estimated eigenvalues of matrices A and N (see Table 4).

Remark 8 The presence of the $\varepsilon(\lambda^{k-1})$ terms in (20)-(22) implies that this algorithm gives inexact results, even if the data is generated by a bilinear system of the form (1). The error decreases as the block size k increases, providing that $\lambda < 1$ and that $k \geq n$. The value of k is limited by the amount of data available, and by computational complexity and memory requirements, both of which grow very rapidly with k — see section 7.3.

5.2 Algorithm II

Here we propose an alternative algorithm for identifying deterministic bilinear systems, which consists of two sub-algorithms. The two sub-algorithms deal with the two cases $l < n$ and $l \geq n$, respectively. There are two possible advantages of Algorithm II over Algorithm I. One is that in the case $l \geq n$, Algorithm II.2 is exact. The other is that, in the case $l < n$, Algorithm II.1 has a lower computational cost, due to using (11) and (12) of Lemma 6, instead of Lemma 7, to linearise the bilinear system. But this lower cost arises from the use of a cruder approximation of the bilinear system's input-ouptut relationship, so one can expect Algorithm II.1 to be less accurate than Algorithm I.

5.2.1 The case $l < n$

If $l < n$, (11) and (12) of Lemma 6 give the following block-form equations:

Theorem 3 (Three Block Form 2) *The system (1) can be written in the following 'three block' form, if condition (4) holds:*

$$Y^c = \mathcal{C}_k X_c + \mathcal{D}_{k,2} U^{c,u} + \varepsilon(\lambda^{k-1}) \tag{33}$$

$$Y^f = \mathcal{C}_k X_f + \mathcal{D}_{k,2} U^{f,u} + \varepsilon(\lambda^{k-1}) \tag{34}$$

$$X_f = \mathcal{A}_k X_c + \mathcal{B}_{k,2} U^{c,u} + \varepsilon(\lambda^{k-1}) \tag{35}$$

where $\mathcal{A}_k, \mathcal{B}_{k,2}, \mathcal{C}_k$ and $\mathcal{D}_{k,2}$ are system-dependent constant matrices.

Proof: From (23) we have,

$$\begin{aligned} Y^c &= \mathcal{C}_k X_c + \mathcal{L}^X_{k,2}(U^{+c} \odot X_c) + \mathcal{L}^U_k U^c \\ &= \mathcal{C}_k X_c + \mathcal{L}^X_{k,2}(I \otimes \triangle^U_k)(U^{+c} \odot U^p) + \mathcal{L}^U_k U^c + \varepsilon(\lambda^k) \end{aligned} \tag{36}$$

The second and third terms of equation (36) are a linear combination of vectors in the row spaces of the matrices $\mathcal{U}^{+c} \odot \mathcal{U}^p$ and $\mathcal{U}^{+c}$. As before, it follows that there exists a matrix $\mathcal{D}_{k,2}$ such that $\mathcal{D}_{k,2} U^{c,u} = \mathcal{L}^X_{k,2}(I \otimes \triangle^U_k)(U^{+c} \odot U^p) + \mathcal{L}^U_k U^c + \varepsilon(\lambda^k)$, so equation (33) holds. (34) and (35) of Theorem 3 can be proved similarily. ∎

Theorem 4 *If the pair (A, C) in (1) is observable, condition (4) holds, and*

$$\begin{pmatrix} Y^c \\ U^{c,u} \\ U^{f,u} \end{pmatrix} \text{ has full row rank,} \tag{37}$$

then, denoting $\mathcal{S}_2 := \mathcal{Y}^c + \mathcal{U}^{c,u}$,

$$\|\Pi_{\mathcal{S}_2^\perp} Y^f - \mathcal{D}_{k,2}\Pi_{\mathcal{S}_2^\perp} U^{f,u}\|_1 = o(\lambda^{k-1}) \tag{38}$$

Proof: Substituting (35) into (34) gives

$$Y^f = \mathcal{C}_k(\mathcal{A}_k X_c + \mathcal{B}_{k,2} U^{c,u}) + \mathcal{D}_{k,2} U^{f,u,y} + \varepsilon_1(\lambda^{k-1}) \tag{39}$$

Since the linear part of the system is observable, we have, from (33),

$$X_c = \mathcal{C}_k^\dagger (Y^c - \mathcal{D}_{k,2} U^{c,u}) + \varepsilon_2(\lambda^{k-1}) \tag{40}$$

Substituting (40) into (39) shows that there exist two matrices C_1, C_2 such that

$$Y^f - \mathcal{D}_{k,2} U^{f,u} = C_1 Y^c + C_2 U^{c,u} + \varepsilon_3(\lambda^{k-1}) \tag{41}$$

Since $\mathcal{S}_2 = \mathcal{Y}^c + \mathcal{U}^{c,u}$ we have, from Lemma 8,

$$\|\Pi_{\mathcal{S}_2^\perp} Y^f - \mathcal{D}_k \Pi_{\mathcal{S}_2^\perp} U^{f,u}\|_1 = o(\lambda^{k-1})$$

This proves Theorem 4. ∎

Algorithm II.1:

Step 1. Decompose Y^f into $\mathcal{C}_k X_f$ and $\mathcal{D}_{k,2} U^{f,u}$ using orthogonal projection. From (38), estimate $\mathcal{D}_{k,2}$ as:

$$\hat{\mathcal{D}}_{k,2} = (\Pi_{\mathcal{S}_2^\perp} \mathcal{Y}^f)(\Pi_{\mathcal{S}_2^\perp} U^{f,u})^\dagger \tag{42}$$

Step 2. Obtain the SVD decomposition and partition by selecting a model order, as in Algorithm I.

$$\left[\Pi_{\mathcal{S}_2^+} Y_{3k-1|2k}\ \Pi_{\mathcal{S}_2^{++}} Y_{3k|2k+1}\right] - \hat{\mathcal{D}}_{k,2}\left[U^u_{3k-1|2k}\ U^u_{3k|2k+1}\right] =: \Gamma\Sigma\Omega^* = \begin{bmatrix} \Gamma_1 & \Gamma_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} \Omega_1^* \\ \Omega_2^* \end{bmatrix}$$

where $\mathcal{S}_2^+ = \mathcal{S}_2 + \mathcal{U}^{f,u}$ and $\mathcal{S}_2^{++} = \mathcal{Y}_{2k|k+1} + \mathcal{U}^u_{2k|k+1} + \mathcal{U}^u_{3k|2k+1}$.

Form the estimates $\hat{C}_k$ and $[\hat{X}_{2k}\ \hat{X}_{2k+1}]$, as in Algorithm I.

Step 3. Estimate the parameters A, B, C, D, N on the basis of equation (7), as in step 3 of Algorithm I.

5.2.2 The case $l \geq n$

As mentioned in Remark 4, (17) holds when $l \geq n$. This results in the following theorem:

Theorem 5 (Three Block Form 3) *If $l \geq n$ and rank$(C) = n$, system (1) can be written in the form, :*

$$Y^c = \mathcal{C}_k X_c + \mathcal{D}_{k,1}\tilde{U}^{c,u,y} \tag{43}$$
$$Y^f = \mathcal{C}_k X_f + \mathcal{D}_{k,1}\tilde{U}^{f,u,y} \tag{44}$$
$$X_f = \mathcal{A}_k X_c + \mathcal{B}_{k,1}\tilde{U}^{c,u,y} \tag{45}$$

where $\mathcal{A}_k, \mathcal{B}_{k,1}, \mathcal{C}_k$ and $\mathcal{D}_{k,1}$ are system-dependent constant matrices.

Proof: From (23), we have

$$\begin{aligned} Y^c &= \mathcal{C}_k X_c + \mathcal{L}^X_{k,2}(U^{+c} \odot X_c) + \mathcal{L}^U_k U^c \\ &= \mathcal{C}_k X_c + \mathcal{L}^X_{k,2}(U^{+c} \odot [C^\dagger(Y_c - DU_c)]) + \mathcal{L}^U_k U^c \\ &= \mathcal{C}_k X_c + \mathcal{L}^X_{k,2}\Big((I \otimes C^\dagger)(U^{+c} \odot Y_c) - (I \otimes C^\dagger D)(U^{+c} \odot U_c)\Big) + \mathcal{L}^U_k U^c \end{aligned} \tag{46}$$

The last two terms of the equation (46) are a linear combination of vectors in the row spaces of the matrices $\mathcal{U}^{+c} \odot \mathcal{Y}_c$, $\mathcal{U}^{+c} \odot \mathcal{U}_c$ and $\mathcal{U}^{+c}$. From the above, and as in the proof of Theorem 1, there exists a matrix $\mathcal{D}_{k,1}$ such that $\mathcal{D}_{k,1}\tilde{U}^{c,u,y} = \mathcal{L}^X_{k,2}\left((I \otimes C^\dagger)(U^{+c} \odot Y_c) - (I \otimes C^\dagger D)(U^{+c} \odot U_c)\right) + \mathcal{L}^U_k U^c$, so (43) holds. (44) and (45) of Theorem 5 can be proved similarly. ∎

Theorem 6 *If $l \geq n$, the pair (A, C) of (1) is observable, and*

$$\begin{pmatrix} Y^c \\ \tilde{U}^{c,u,y} \\ \tilde{U}^{f,u,y} \end{pmatrix} \text{ has full row rank,} \tag{47}$$

then , denoting $\mathcal{S}_1 := \mathcal{Y}^c + \tilde{\mathcal{U}}^{c,u,y}$,

$$\Pi_{\mathcal{S}_1^\perp} Y^f = \mathcal{D}_{k,1}\Pi_{\mathcal{S}_1^\perp}\tilde{U}^{f,u,y} \tag{48}$$

Proof: From Theorem 5, we know that $\mathcal{X}_f \subset \mathcal{X}^c + \tilde{\mathcal{U}}^{c,u,y} \subset \mathcal{Y}^c + \tilde{\mathcal{U}}^{c,u,y}$.
Since condition (47) holds, $\mathcal{X}_f + \tilde{\mathcal{U}}^{f,u,y} = \mathcal{X}_f \oplus \tilde{\mathcal{U}}^{f,u,y} \subset (\mathcal{Y}^c + \mathcal{U}^{c,u,y}) \oplus \tilde{\mathcal{U}}^{f,u,y}$. By projecting both sides of equation (44) onto $\mathcal{S}_1^\perp$ the statement of Theorem 6 is obtained. ∎

Algorithm II.2:

Step 1. Decompose Y^f into $\mathcal{C}_k X_f$ and $\mathcal{D}_{k,1}\tilde{U}^{f,u,y}$ using orthogonal projection. From (48), estimate $\mathcal{D}_{k,1}$ as:

$$\hat{\mathcal{D}}_{k,1} = (\Pi_{\mathcal{S}_1^\perp}\mathcal{Y}^f)(\Pi_{\mathcal{S}_1^\perp}\tilde{U}^{f,u,y})^\dagger \tag{49}$$

Step 2. Obtain the SVD decomposition and partition by selecting a model order, as in Algorithm I.

$$\left[\Pi_{\mathcal{S}_1^+} Y_{3k-1|2k}\ \Pi_{\mathcal{S}_1^{++}} Y_{3k|2k+1}\right] - \hat{\mathcal{D}}_{k,1}\left[\tilde{U}^{u,y}_{3k-1|2k}\ \tilde{U}^{u,y}_{3k|2k+1}\right] =: \Gamma\Sigma\Omega^* = \left[\begin{array}{cc} \Gamma_1 & \Gamma_2 \end{array}\right] \left[\begin{array}{cc} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{array}\right] \left[\begin{array}{c} \Omega_1^* \\ \Omega_2^* \end{array}\right]$$

where $\mathcal{S}_1^+ = \mathcal{S}_1 + \tilde{\mathcal{U}}^{f,u,y}$ and $\mathcal{S}_1^{++} = \mathcal{Y}_{2k|k+1} + \tilde{\mathcal{U}}^{u,y}_{2k|k+1} + \tilde{\mathcal{U}}^{u,y}_{3k|2k+1}$.

Form the estimates $\hat{C}_k$ and $[\hat{X}_{2k}\ \hat{X}_{2k+1}]$, as in Algorithm I.

Step 3. Estimate the parameters A, B, C, D, N on the basis of equation (7), as in step 3 of Algorithm I.

Remark 9 We envisage that one would usually start by using Algorithm II.1. If the singular values (in step 2) indicated that $l \geq n$ might be a possibility, then one could try Algorithm II.2.

Remark 10 Although Algorithms I and II.1 give the same results asymptotically as k increases, with a given k one should expect a trade-off between greater accuracy of Algorithm I, due to its more complete use of the assumed bilinear structure of the system, and the lower computational complexity of Algorithm II.1.

Remark 11 The 'full row rank' requirement in Theorems 2, 4 and 6 can only be met if $k \geq n$.

6 The stochastic case

In this section, two 'four-block' subspace algorithms for the identification of stochastic bilinear systems are developed. The fourth block of data which is used here is the 'remote future' data block (matrices U_r, X_r, Y_r, and matrices constructed from these).

Lemma 9 *Let $\{x_n; n \in \mathbf{Z}_+\}$ be a sequence of independent identically distributed random variables with $Ex_0 = 0$, $\{y_n; n \in \mathbf{Z}_+\}$ be a sequence, which is uncorrelated with $\{x_n; n \in \mathbf{Z}_+\}$, and with $E\|y_0\| < \infty, E\|y_0^2\| < \infty$. Let $X_i = [x_i, x_{i+1}, ..., x_{i+j}]$ and $Y_i = [y_i, y_{i+1}, ..., y_{i+j}]$. Then*

$$\Pi_{Y_i} X_i \quad \rightarrow \quad 0 \quad w.p.1 \quad as \quad j \rightarrow \infty \tag{50}$$

Proof: See Appendix.

6.1 Algorithm III

Theorem 7 (Four Block Form 1) *The system (2) can be written in the following form if condition (4) holds:*

$$\begin{aligned} Y^c &= \mathcal{O}_k X_c + \mathcal{T}_k^u U^{c,u,y} + \mathcal{T}_k^v (U^{+c} \odot V_c) + \mathcal{L}_k^W W^c + \mathcal{L}_k^V V^c + \varepsilon(\lambda^{k-1}) & (51)\\ Y^f &= \mathcal{O}_k X_f + \mathcal{T}_k^u U^{f,u,y} + \mathcal{T}_k^v (U^{+f} \odot V_f) + \mathcal{L}_k^W W^f + \mathcal{L}_k^V V^f + \varepsilon(\lambda^{k-1}) & (52)\\ Y^r &= \mathcal{O}_k X_r + \mathcal{T}_k^u U^{r,u,y} + \mathcal{T}_k^v (U^{+r} \odot V_r) + \mathcal{L}_k^W W^r + \mathcal{L}_k^V V^r + \varepsilon(\lambda^{k-1}) & (53) \end{aligned}$$

$$\begin{aligned} X_f &= \mathcal{F}_k X_c + \mathcal{G}_k^u U^{c,u,y} + \mathcal{G}_k^v (U^{+c} \odot V_c) + \triangle_k^W W^c + \varepsilon(\lambda^{k-1}) & (54)\\ X_r &= \mathcal{F}_k X_f + \mathcal{G}_k^u U^{f,u,y} + \mathcal{G}_k^v (U^{+f} \odot V_f) + \triangle_k^W W^f + \varepsilon(\lambda^{k-1}) & (55) \end{aligned}$$

where $\mathcal{O}_k, \mathcal{T}_k^u, \mathcal{T}_k^v, \mathcal{F}_k, \mathcal{G}_k^u$ *and* $\mathcal{G}_k^v$ *are system-dependent constant matrices.*

Proof: From equation (9) of Lemma 5 and from Lemma 3, we know that

$$\begin{aligned} Y^c &= \mathcal{L}_k^X X^c + \mathcal{L}_k^U U^c + \mathcal{L}_k^W W^c + \mathcal{L}_k^V V^c \\ &= \mathcal{L}_k^X \begin{pmatrix} X_c \\ U^{+c} \odot X_c \end{pmatrix} + \mathcal{L}_k^U U^c + \mathcal{L}_k^W W^c + \mathcal{L}_k^V V^c & (56) \end{aligned}$$

Let $\mathcal{L}_k^X = \left[\mathcal{O}_k, \mathcal{L}_{k,2}^X\right]$, where $\mathcal{O}_k$ is the first n columns of the matrix $\mathcal{L}_k^X$ and $\mathcal{L}_{k,2}^X$ is the last $(m+1)^k - 1$ columns of $\mathcal{L}_k^X$. Then, as in the proof of Theorem 1, equation (56) can be written as:

$$\begin{aligned} Y^c &= \mathcal{O}_k X_c + \mathcal{L}_{k,2}^X (U^{+c} \odot X_c) \\ &\quad + \mathcal{L}_k^U U^c + \mathcal{L}_c^W W^c + \mathcal{L}_k^V V^c \\ &= \mathcal{O}_k X_c + \mathcal{L}_{k,2}^X \left(U^{+c} \odot \left[C^\dagger (Y_c - DU_c - V_c) + (I - C^\dagger C) X_c\right]\right) \\ &\quad + \mathcal{L}_k^U U^c + \mathcal{L}_k^W W^c + \mathcal{L}_k^V V^c \\ &= \mathcal{O}_k X_c + \mathcal{L}_{k,2}^X \left(U^{+c} \odot \left[C^\dagger (Y_c - DU_c - V_c) + (I - C^\dagger C) \triangle_k^U U^p\right]\right) \\ &\quad + \mathcal{L}_k^U U^c + \mathcal{L}_k^W W^c + \mathcal{L}_k^V V^c + \varepsilon(\lambda^{k-1}) \\ &= \mathcal{O}_k X_c + \mathcal{L}_{k,2}^X \Big((I \otimes C^\dagger)(U^{+c} \odot Y_c) - (I \otimes C^\dagger D)(U^{+c} \odot U_c) + [I \otimes (I - C^\dagger C) \triangle_k^U](U^{+c} \odot U^p)\Big) \\ &\quad + \mathcal{L}_{k,2}^X (I \otimes C^\dagger)(U^{+c} \odot V_c) + \mathcal{L}_k^U U^c + \mathcal{L}_k^W W^c + \mathcal{L}_k^V V^c + \varepsilon(\lambda^{k-1}) & (57) \end{aligned}$$

The second term of equation (57) is a linear combination of vectors in the spaces spanned by the rows of the matrices $\mathcal{U}^{+c} \odot \mathcal{Y}_c, \mathcal{U}^{+c} \odot \mathcal{U}_c, \mathcal{U}^{+c} \odot \mathcal{U}^p$ and $\mathcal{U}^{+c}$. According to Lemma 2, $(\mathcal{U}^{+c} \odot \mathcal{U}_c)$ can be decomposed into the direct sum of two subspaces, namely $\mathcal{U}^{++c} \oplus \left(\mathcal{U}_{2k-1|k}^+ \odot \mathcal{U}_c\right)$ and the latter subspace is contained in $\mathcal{U}^c$. Hence there exists a matrix $\mathcal{T}_k^u$ such that
$\mathcal{T}_k^u U^{c,u,y} = \mathcal{L}_{k,2}^X \left((I \otimes C^\dagger)(U^{+c} \odot Y_c) - (I \otimes C^\dagger D)(U^{++c}) + (I \otimes (I - C^\dagger C \triangle_k^U)(U^{+c} \odot U^p) + \mathcal{L}_k^U U^c\right)$.
So (51) holds, if we take $\mathcal{T}_k^v = \mathcal{L}_{k,2}^X (I \otimes C^\dagger)$. (52), (53), (54) and (55) can be proved similarily. ∎

Theorem 8 *If the pair (A, C) in system (2) is observable, condition (4) holds, and*

$$\begin{pmatrix} Y^c \\ U^{c,u,y} \\ U^{f,u,y} \\ U^{r,u,y} \end{pmatrix} \quad \textit{has full row rank,} \tag{58}$$

then, denoting $\mathcal{S}_3 := \mathcal{Y}^c + \mathcal{U}^{c,u,y} + \mathcal{U}^{f,u,y} + \mathcal{U}^{r,u,y}$ and $\mathcal{R}_1 = \Pi_{\mathcal{S}_3}\mathcal{Y}^f + \mathcal{U}^{f,u,y}$, =

$$\|\Pi_{\mathcal{R}_1^\perp}\Pi_{\mathcal{S}_3}\mathcal{Y}^r - \mathcal{T}_k^u\Pi_{\mathcal{R}_1^\perp}U^{r,u,y}\|_1 \to o(\lambda^{k-1}) \quad w.p.1 \quad as \quad j \to \infty \tag{59}$$

Proof: From its structure, we know that all the rows of W^r are generated by sequences of the form $\Pi_{\nu=1}^{\mu} u_{i_\nu} \times w_i$ where $3k \leq i \leq 4k-1, \mu \leq k, 3k \leq i_\nu \leq 4k-1$. $\mathcal{S}_3$ is independent of w_i, hence from Lemma 9 we have $\Pi_{\mathcal{S}_3}W^r \to 0$ w.p.1 as $j \to \infty$. From the the structure of V^r and $U^{+r} \odot V_r$ we have

$$\Pi_{\mathcal{S}_3}V^r = \Pi_{\mathcal{S}_3}(U^{+r} \odot V_r)$$

and hence, by a similar argument,

$$\Pi_{\mathcal{S}_3}V^r \to 0 \quad w.p.1 \quad \text{as} \quad j \to \infty$$

Hence, using (53) and (55), we have

$$\begin{aligned} \Pi_{\mathcal{S}_3}Y^r &\to \mathcal{O}_k\Pi_{\mathcal{S}_3}X_r + \mathcal{T}_k^u U^{r,u,y} + \varepsilon(\lambda^{k-1}) \quad w.p.1 \quad \text{as} \quad j \to \infty \\ &\to \mathcal{O}_k\Pi_{\mathcal{S}_3}(\mathcal{F}_k X_f + \mathcal{G}_k^u U^{f,u,y}) + \mathcal{T}_k^u U^{r,u,y} + \varepsilon_1(\lambda^{k-1}) \quad w.p.1 \quad \text{as} \quad j \to \infty \end{aligned} \tag{60}$$

Since the linear part of the system is observable, we have

$$\Pi_{\mathcal{S}_3}X_f \to \mathcal{O}_k^\dagger\Pi_{\mathcal{S}_3}(Y^f - \mathcal{T}_k^u U^{f,u,y}) + \varepsilon_2(\lambda^{k-1}) \quad w.p.1 \quad \text{as} \quad j \to \infty \tag{61}$$

Substituting (61) into (60), we deduce that there are matrices C_1, C_2 such that

$$\Pi_{\mathcal{S}_3}Y^r - \mathcal{T}_k^u U^{r,u,y} \to C_1 Y^f + C_2 U^{f,u,y} + \varepsilon_3(\lambda^{k-1}) \quad w.p.1 \quad \text{as} \quad j \to \infty \tag{62}$$

Now applying Lemma 8 proves Theorem 8. ∎

Theorem 8 uses two consecutive projections to remove the effects of the stochastic disturbances w_t and v_t. The 'current' data block can be regarded as providing an instrumental variable for this purpose, the remaining three data blocks being available for the estimation of the system matrices, as before. Algorithm III, based on this result, has four steps; the matrix $\mathcal{T}_k^u$ is estimated in Step 1, by using the two projections. Steps 2 and 3 are used to estimate state sequences and the system matrices, respectively, as in the deterministic case. Step 4 is concerned only with estimating the covariance matrix of the stochastic disturbances, if this is required. Algorithm IV differs from Algorithm III only in that the projections are on slightly different spaces (see Theorems 9 and 11), and therefore different matrices ($\mathcal{T}_{k,2}$, $\mathcal{T}_{k,1}$) are estimated.

Algorithm III:

Step 1. Decompose Y^r into $\mathcal{O}_k X_r$ and $\mathcal{T}_k^u U^{r,u,y}$ using orthogonal projection: from (59) of Theorem 8, estimate $\mathcal{T}_k^u$ as:

$$\hat{\mathcal{T}}_k^u = (\Pi_{\mathcal{R}_1^\perp}\Pi_{\mathcal{S}_3}\mathcal{Y}^r)(\Pi_{\mathcal{R}_1^\perp}U^{r,u,y})^\dagger \tag{63}$$

Step 2. Obtain the SVD decomposition and partition by selecting a model order as in Algorithm I.

$$\left[\Pi_{\mathcal{S}_3}Y_{4k-1|3k}\ \Pi_{\mathcal{S}_3^+}Y_{4k|3k+1}\right]-\hat{\mathcal{T}}_k^u\left[U^{u,y}_{4k-1|3k}\ U^{u,y}_{4k|3k+1}\right] =: \Gamma\Sigma\Omega^* = \left[\begin{array}{cc}\Gamma_1 & \Gamma_2\end{array}\right]\left[\begin{array}{cc}\Sigma_1 & 0\\ 0 & \Sigma_2\end{array}\right]\left[\begin{array}{c}\Omega_1^*\\ \Omega_2^*\end{array}\right]$$

where $\mathcal{S}_3^+ := \mathcal{Y}_{2k|k+1} + \mathcal{U}^{u,y}_{2k|k+1} + \mathcal{U}^{u,y}_{3k|2k+1} + \mathcal{U}^{u,y}_{4k|4k+1}$. Form the estimates $\hat{C}_k$ and $[\hat{X}_{3k}\ \hat{X}_{3k+1}]$, as in Algorithm I.

Stepm 3. Estimate the parameters A, B, C, D, N on the basis of equation (7)

$$\left[\begin{array}{c}\hat{X}_{3k+1}\\ Y_{3k}\end{array}\right] = \left[\begin{array}{ccc}A & N & B\\ C & 0 & D\end{array}\right]\left[\begin{array}{c}\hat{X}_{3k}\\ U_{3k}\odot\hat{X}_{3k}\\ U_{3k}\end{array}\right] \tag{64}$$

in a least-squares sense, as in step 3 of Algorithm I. Denote the resulting estimates by $\hat{A}, \hat{B}, \hat{C}, \hat{D}$ and $\hat{N}$.

Step 4. Estimate the covariance matrix (if needed) by calculating

$$\begin{aligned}\left[\begin{array}{c}\epsilon_w\\ \epsilon_v\end{array}\right] &= \left[\begin{array}{c}\hat{X}_{3k+1}\\ Y_{3k}\end{array}\right] - \left[\begin{array}{ccc}\hat{A} & \hat{N} & \hat{B}\\ \hat{C} & 0 & \hat{D}\end{array}\right]\left[\begin{array}{c}\hat{X}_{3k}\\ U_{3k}\odot\hat{X}_{3k}\\ U_{3k}\end{array}\right]\\ \left[\begin{array}{cc}\hat{Q} & \hat{S}\\ \hat{S}^T & \hat{R}\end{array}\right] &= \frac{1}{j}\left[\left(\begin{array}{c}\epsilon_w\\ \epsilon_v\end{array}\right)\left(\begin{array}{c}\epsilon_w\\ \epsilon_v\end{array}\right)^*\right]\end{aligned}$$

6.2 Algorithm IV

Here we propose an alternative algorithm for the stochastic case. As for the deterministic case, this consists of two sub-algorithms, one for the case $l < n$, and the other for the case $l \geq n$. There are similar advantages and trade-offs for Algorithm IV, relative to Algorithm III, as regards computational complexity and accuracy, as in the deterministic case for Algorithms II and I. In particular, Algorithm IV gives unbiased results if $l \geq n$.

6.2.1 The case $l < n$

In the case of $l < n$, equations (11-13) of Lemma 6 give the following block-form equations:

Theorem 9 (Four Block Form 2) *The system (2) can be written in the following form if condition (4) holds:*

$$
\begin{aligned}
Y^c &= \mathcal{O}_k X_c + \mathcal{T}^u_{k,2} U^{c,u} + \mathcal{T}^v_{k,2}(U^{+c} \odot V_c) + \mathcal{L}^W_k W^c + \mathcal{L}^V_k V^c + \varepsilon(\lambda^{k-1}) && (65)\\
Y^f &= \mathcal{O}_k X_f + \mathcal{T}^u_{k,2} U^{f,u} + \mathcal{T}^v_{k,2}(U^{+f} \odot V_f) + \mathcal{L}^W_k W^f + \mathcal{L}^V_k V^f + \varepsilon(\lambda^{k-1}) && (66)\\
Y^r &= \mathcal{O}_k X_r + \mathcal{T}^u_{k,2} U^{r,u} + \mathcal{T}^v_{k,2}(U^{+r} \odot V_r) + \mathcal{L}^W_k W^r + \mathcal{L}^V_k V^r + \varepsilon(\lambda^{k-1}) && (67)\\
X_f &= \mathcal{F}_k X_c + \mathcal{G}^u_{k,2} U^{c,u} + \mathcal{G}^v_{k,2}(U^{+c} \odot V_c) + \triangle^W_k W^c + \varepsilon(\lambda^{k-1}) && (68)\\
X_r &= \mathcal{F}_k X_f + \mathcal{G}^u_{k,2} U^{f,u} + \mathcal{G}^v_{k,2}(U^{+f} \odot V_f) + \triangle^W_k W^f + \varepsilon(\lambda^{k-1}) && (69)
\end{aligned}
$$

where $\mathcal{O}_k, \mathcal{T}^u_{k,2}, \mathcal{T}^v_{k,2}$, $\mathcal{F}_k, \mathcal{G}^u_{k,2}$ and $\mathcal{G}^v_{k,2}$ are system-dependent constant matrices.

Proof: From (56) we have:

$$
\begin{aligned}
Y^c &= \mathcal{O}_k X_c + \mathcal{L}^X_{k,2}(U^{+c} \odot X_c)\\
&\quad + \mathcal{L}^U_k U^c + \mathcal{L}^W_c W^c + \mathcal{L}^V_k V^c\\
&= \mathcal{O}_k X_c + \mathcal{L}^X_{k,2}\left(U^{+c} \odot (\triangle^U_k U^p + \triangle^W_k W^p)\right)\\
&\quad + \mathcal{L}^U_k U^c + \mathcal{L}^W_k W^c + \mathcal{L}^V_k V^c + \varepsilon(\lambda^{k-1})\\
&= \mathcal{O}_k X_c + \mathcal{L}^X_{k,2}\left(I \otimes \triangle^U_k (U^{+c} \odot U^p)\right)\\
&\quad + \mathcal{L}^X_{k,2}(I \otimes C^\dagger)(U^{+c} \odot V_c) + \mathcal{L}^U_k U^c + \mathcal{L}^W_k W^c + \mathcal{L}^V_k V^c + \varepsilon(\lambda^{k-1}) && (70)
\end{aligned}
$$

The second term of equation (70) is a linear combination of vectors in the spaces spanned by the rows of the matrices $\mathcal{U}^{+c} \odot \mathcal{U}^p$ and $\mathcal{U}^{+c}$. According to Lemma 2, $(\mathcal{U}^{+c} \odot \mathcal{U}_c)$ can be decomposed into the direct sum of two subspaces: $\mathcal{U}^{++c} \oplus \left(\mathcal{U}^+_{2k-1|k} \odot \mathcal{U}_c\right)$ and the latter subspace is contained in $\mathcal{U}^c$. Hence there exists a matrix $\mathcal{T}^u_{k,2}$ such that $\mathcal{T}^u_{k,2} U^{c,u} = \mathcal{L}^X_{k,2}\left((I \otimes \triangle^U_k)(U^{+c} \odot U^p) + \mathcal{L}^U_k U^c\right)$. So (65) holds, if we take $\mathcal{T}^v_k = \mathcal{L}^X_{k,2}(I \otimes C^\dagger)$. (66), (67), (68) and (69) can be proved similarly. ∎

Theorem 10 *Suppose that the pair (A, C) of the system (2) is observable, and that condition (4) holds, and*

$$
\begin{pmatrix} Y^c \\ U^{c,u} \\ U^{f,u} \\ U^{r,u} \end{pmatrix} \text{ has full row rank,} \qquad (71)
$$

then, denoting $\mathcal{S}_5 = \mathcal{Y}^c + \mathcal{U}^{c,u} + \mathcal{U}^{f,u} + \mathcal{U}^{r,u}$ and $\mathcal{R}_3 = \Pi_{\mathcal{S}_5} \mathcal{Y}^f + \mathcal{U}^{f,u}$. we have:

$$
\|\Pi_{\mathcal{R}_3^\perp} \Pi_{\mathcal{S}_5} \mathcal{Y}^r - \mathcal{T}^u_{k,2} \Pi_{\mathcal{R}_3^\perp} U^{r,u}\|_1 \to o(\lambda^{k-1}) \quad w.p.1 \quad as \quad j \to \infty \qquad (72)
$$

Proof: From Lemma 9 it follows that

$$
\Pi_{\mathcal{S}_5} W^r \to 0 \quad \text{and} \quad \Pi_{\mathcal{S}_5} V^r \to 0 \quad w.p.1 \quad \text{as} \quad j \to \infty
$$

From (67) and (69), we have

$$\begin{aligned} \Pi_{\mathcal{S}_5} Y^r &\rightarrow \mathcal{O}_k \Pi_{\mathcal{S}_5} X_r + \mathcal{T}_k^u U^{r,u} + \varepsilon(\lambda^{k-1}) \quad w.p.1 \quad \text{as} \quad j \rightarrow \infty \\ &\rightarrow \mathcal{O}_k \Pi_{\mathcal{S}_3} (\mathcal{F}_k X_f + \mathcal{G}_{k,2}^u U^{f,u}) + \mathcal{T}_{k,2}^u U^{r,u} + \varepsilon_1(\lambda^{k-1}) \quad w.p.1 \quad \text{as} \quad j \rightarrow \infty \end{aligned} \tag{73}$$

Since the linear part of the system is observable, we have

$$\Pi_{\mathcal{S}_5} X_f \rightarrow \mathcal{O}_k^{\dagger} \Pi_{\mathcal{S}_5} (Y^f - \mathcal{T}_{k,2}^u U^{f,u}) + \varepsilon_2(\lambda^{k-1}) \quad w.p.1 \quad \text{as} \quad j \rightarrow \infty \tag{74}$$

Substituting (74) into (73) shows that there are matrices C_1, C_2 such that

$$\Pi_{\mathcal{S}_5} Y^r - \mathcal{T}_{k,2}^u U^{r,u} \rightarrow C_1 Y^f + C_2 U^{f,u} + \varepsilon_3(\lambda^{k-1}) \quad w.p.1 \quad \text{as} \quad j \rightarrow \infty \tag{75}$$

Now applying Lemma 8 proves Theorem 10. ∎

Algorithm IV.1:

Step 1. Decompose Y^r into $\mathcal{O}_k X_r$ and $\mathcal{T}_{k,2}^u U^{r,u}$ using orthogonal projection. From (72) of Theorem 10, estimate $\mathcal{T}_{k,2}^u$ as

$$\hat{\mathcal{T}}_{k,2}^u = (\Pi_{\mathcal{R}_3^{\perp}} \Pi_{\mathcal{S}_5} \mathcal{Y}^r)(\Pi_{\mathcal{R}_3^{\perp}} U^{r,u})^{\dagger} \tag{76}$$

Step 2. Obtain the SVD decomposition and partition by selecting a model order as in Algorithm I.

$$\left[\Pi_{\mathcal{S}_5} Y_{4k-1|3k} \; \Pi_{\mathcal{S}_5^+} Y_{4k|3k+1}\right] - \hat{\mathcal{T}}_{k,2}^u \left[U_{4k-1|3k}^u \; U_{4k|3k+1}^u\right] =: \Gamma \Sigma \Omega^* = \begin{bmatrix} \Gamma_1 & \Gamma_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} \Omega_1^* \\ \Omega_2^* \end{bmatrix}$$

where $\mathcal{S}_5^+ := \mathcal{Y}_{2k|k+1} + \mathcal{U}_{2k|k+1}^u + \mathcal{U}_{3k|2k+1}^u + \mathcal{U}_{4k|4k+1}^u$.

Form estimates $\hat{C}_k$ and $[\hat{X}_{3k} \; \hat{X}_{3k+1}]$, as in Algorithm III.

Step 3. Estimate the parameters A, B, C, D, N as in step 3 of Algorithm III.

Step 4. Estimate Q, R, S (if needed) as in step 4 of Algorithm III.

6.2.2 The case $l \geq n$

As before, when $l \geq n$ then (17) holds, which results in:

Theorem 11 (Four Block Form 3) *If $l \geq n$, the system (2) can be written in the following form:*

$$Y^c = \mathcal{O}_k X_c + \mathcal{T}_{k,1}^u \tilde{U}^{c,u,y} + \mathcal{T}_k^v U^{+c} \odot V_c + \mathcal{L}_k^W W^c + \mathcal{L}_k^V V^c \tag{77}$$

$$Y^f = \mathcal{O}_k X_f + \mathcal{T}_{k,1}^u \tilde{U}^{f,u,y} + \mathcal{T}_k^v U^{+f} \odot V_f + \mathcal{L}_k^W W^f + \mathcal{L}_k^V V^f \tag{78}$$

$$Y^r = \mathcal{O}_k X_r + \mathcal{T}_{k,1}^u \tilde{U}^{r,u,y} + \mathcal{T}_k^v U^{+r} \odot V_r + \mathcal{L}_k^W W^r + \mathcal{L}_k^V V^r \tag{79}$$

$$
\begin{aligned}
X_f &= \mathcal{F}_k X_c + \mathcal{G}^u_{k,1}\tilde{U}^{c,u,y} + \mathcal{G}^v_k U^{+c} \odot V_c + \triangle^W_k W^c && (80)\\
X_r &= \mathcal{F}_k X_f + \mathcal{G}^u_{k,1}\tilde{U}^{f,u,y} + \mathcal{G}^v_k U^{+f} \odot V_f + \triangle^W_k W^f && (81)
\end{aligned}
$$

where $\mathcal{O}_k, \mathcal{T}^u_{k,1}, \mathcal{T}^v_k, \mathcal{F}_k, \mathcal{G}^u_{k,1}$ and $\mathcal{G}^v_k$ are system-dependent constant matrices.

Proof: From (56), we have:

$$
\begin{aligned}
Y^c &= \mathcal{O}_k X_c + \mathcal{L}^X_{k,2}(U^{+c} \odot X_c) \\
&\quad + \mathcal{L}^U_k U^c + \mathcal{L}^W_c W^c + \mathcal{L}^V_k V^c \\
&= \mathcal{O}_k X_c + \mathcal{L}^X_{k,2}\left(U^{+c} \odot \left[C^\dagger (Y_c - DU_c - V_c)\right]\right) \\
&\quad + \mathcal{L}^U_k U^c + \mathcal{L}^W_k W^c + \mathcal{L}^V_k V^c \\
&= \mathcal{O}_k X_c + \mathcal{L}^X_{k,2}\left(U^{+c} \odot \left[C^\dagger (Y_c - DU_c - V_c)\right]\right) \\
&\quad + \mathcal{L}^U_k U^c + \mathcal{L}^W_k W^c + \mathcal{L}^V_k V^c \\
&= \mathcal{O}_k X_c + \mathcal{L}^X_{k,2}\left((I \otimes C^\dagger)(U^{+c} \odot Y_c) - (I \otimes C^\dagger D)(U^{+c} \odot U_c)\right) \\
&\quad + \mathcal{L}^X_{k,2}(I \otimes C^\dagger)(U^{+c} \odot V_c) + \mathcal{L}^U_k U^c + \mathcal{L}^W_k W^c + \mathcal{L}^V_k V^c && (82)
\end{aligned}
$$

The second term of equation (82) is a linear combination of vectors in the spaces spanned by the rows of the matrices $\mathcal{U}^{+c} \odot \mathcal{Y}_c$,$\mathcal{U}^{+c} \odot \mathcal{U}_c$ and $\mathcal{U}^{+c}$. Hence, as above, there exists a matrix $\mathcal{T}^u_{k,1}$ such that
$\mathcal{T}^u_{k,1}\tilde{U}^{c,u,y} = \mathcal{L}^X_{k,2}\left((I \otimes C^\dagger)(U^{+c} \odot Y_c) - (I \otimes C^\dagger D)(U^{++c}) + \mathcal{L}^U_k U^c\right)$. $\mathcal{T}^v_k = \mathcal{L}^X_{k,2}(I \otimes C^\dagger)$. So (77) holds if we take $\mathcal{T}^v_k = \mathcal{L}^X_{k,2}(I \otimes C^\dagger)$. (78), (79), (80) and (81) can be proved similarly. ∎

We also have:

Theorem 12 *If $l \geq n$ and rank$(C) = n$, the pair (A, C) in (2) is observable, and if*

$$
\begin{pmatrix} Y^c \\ \tilde{U}^{c,u,y} \\ \tilde{U}^{f,u,y} \\ \tilde{U}^{r,u,y} \end{pmatrix} \quad \textit{has full row rank,} \tag{83}
$$

then, denoting $\mathcal{S}_4 := \mathcal{Y}^c + \tilde{\mathcal{U}}^{c,u,y} + \tilde{\mathcal{U}}^{f,u,y} + \tilde{\mathcal{U}}^{r,u,y}$ and $\mathcal{R}_2 = \Pi_{\mathcal{S}_4}\mathcal{Y}^f + \tilde{\mathcal{U}}^{f,u,y}$, we have

$$
\Pi_{\mathcal{R}_2^\perp}\Pi_{\mathcal{S}_4}\mathcal{Y}^r \to \mathcal{T}^u_{k,1}\Pi_{\mathcal{R}_2^\perp}\tilde{U}^{r,u,y} \quad w.p.1 \quad as \quad j \to \infty \tag{84}
$$

Proof: From Lemma 9, and arguing as before, it follows that

$$
\Pi_{\mathcal{S}_4} W^r \to 0 \quad \text{and} \quad \Pi_{\mathcal{S}_4} V^r \to 0 \quad w.p.1 \quad \text{as} \quad j \to \infty
$$

From (80) and (78) we have

$$
\Pi_{\mathcal{S}_4}\mathcal{X}_r \subset \Pi_{\mathcal{S}_4}\mathcal{X}_f + \tilde{\mathcal{U}}^{f,u,y} \subset \Pi_{\mathcal{S}_4}\mathcal{Y}^f + \tilde{\mathcal{U}}^{f,u,y} \quad w.p.1 \quad \text{as} \quad j \to \infty \tag{85}
$$

and from (80) and (77) we get

$$\begin{aligned}\Pi_{\mathcal{S}_4}\mathcal{Y}^f &\subset \Pi_{\mathcal{S}_4}\mathcal{X}_f + \tilde{\mathcal{U}}^{f,u,y} \subset \Pi_{\mathcal{S}_4}\mathcal{X}_c + \tilde{\mathcal{U}}^{c,u,y} + \tilde{\mathcal{U}}^{f,u,y} \quad w.p.1 \quad \text{as} \quad j \to \infty \\ &\subset \Pi_{\mathcal{S}_4}\mathcal{Y}^c + \tilde{\mathcal{U}}^{c,u,y} + \tilde{\mathcal{U}}^{f,u,y} \subset \mathcal{Y}^c + \tilde{\mathcal{U}}^{c,u,y} + \tilde{\mathcal{U}}^{f,u,y} \quad w.p.1 \quad \text{as} \quad j \to \infty \end{aligned} \tag{86}$$

Hence

$$\begin{aligned}\Pi_{\mathcal{S}_4}\mathcal{X}_r + \tilde{\mathcal{U}}^{r,u,y} &= \Pi_{\mathcal{S}_4}\mathcal{X}_r \oplus \tilde{\mathcal{U}}^{r,u,y} \\ &\subset (\Pi_{\mathcal{S}_4}\mathcal{Y}^f + \tilde{\mathcal{U}}^{f,u,y}) \oplus \tilde{\mathcal{U}}^{r,u,y} \quad w.p.1 \quad \text{as} \quad j \to \infty\end{aligned}$$

But $\Pi_{\mathcal{S}_4}X_r \subset \mathcal{R}_2$ w.p.1 as $j \to \infty$ so, since (83) holds, projecting both sides of equation (79) onto $\mathcal{S}_4$ and then onto $\mathcal{R}_2^{\perp}$ proves Theorem 12.

Algorithm IV.2:

Step 1. Decompose Y^r into $\mathcal{O}_k X_r$ and $\mathcal{T}_{k,1}^u \tilde{U}^{r,u,y}$ using orthgonal projection. From (84) of Theorem 12, estimate $\mathcal{T}_{k,1}^u$ as:

$$\hat{\mathcal{T}}_{k,1}^u = (\Pi_{\mathcal{R}_2^{\perp}}\Pi_{\mathcal{S}_4}\mathcal{Y}^r)(\Pi_{\mathcal{R}_2^{\perp}}\tilde{U}^{r,u,y})^{\dagger} \tag{87}$$

Step 2. Obtain the SVD decomposition and partition by selecting a model order as in Algorithm I:

$$\left[\Pi_{\mathcal{S}_4}Y_{4k-1|3k}\ \Pi_{\mathcal{S}_4^+}Y_{4k|3k+1}\right] - \hat{\mathcal{D}}_{k,2}\left[\tilde{U}^u_{4k-1|3k}\ \tilde{U}^u_{4k|3k+1}\right] =: \Gamma\Sigma\Omega^* = \left[\begin{array}{cc}\Gamma_1 & \Gamma_2\end{array}\right]\left[\begin{array}{cc}\Sigma_1 & 0\\ 0 & \Sigma_2\end{array}\right]\left[\begin{array}{c}\Omega_1^*\\ \Omega_2^*\end{array}\right]$$

where $\mathcal{S}_4^+ := \mathcal{Y}_{2k|k+1} + \tilde{\mathcal{U}}^u_{2k|k+1} + \tilde{\mathcal{U}}^u_{3k|2k+1} + \tilde{\mathcal{U}}^u_{4k|4k+1}$.

Form estimates $\hat{C}_k$ and $[\hat{X}_{3k}\ \hat{X}_{3k+1}]$, as in Algorithm III.

Step 3. Estimate the parameters A, B, C, D, N on the basis of equation (7) as in step 3 of Algorithm III.

Step 4. Estimate Q, R, S, as in step 4 of Algorithm III.

Remark 12 As with Algorithm II, we envisage that one would usually start by using Algorithm IV.1. If the singular values indicated that $l \geq n$ might be a possibility, then one could try Algorithm IV.2.

Remark 13 The 'full row rank' requirement in Theorems 8, 10 and 12 can only be met if $k \geq n$.

7 Examples

In this section we compare the performance of the algorithms introduced in this paper with each other and with the bilinear N4SID algorithm introduced in [10], using both white and non-white inputs applied to known systems to generate the data. The results shown in Tables 1, 2, 4, and 6 are each based on 100 realisations of the input process $\mathbf{u_t}$. The values given in these tables are the sample means obtained, and are followed in parentheses by the sample standard deviations (of the modulus in those cases for which the values are complex).

We examine first the performance of Algorithms I and II in the deterministic case ($w_t = 0$ and $v_t = 0$). Then we examine the performance of Algorithms III and IV in the stochastic case.

7.1 The deterministic case: Algorithms I and II

We consider two second-order bilinear systems which were introduced in [10, 13], and one third-order system newly introduced here.

Example 1 The true system is

$$A = \begin{pmatrix} 0 & 0.5 \\ -0.5 & 0 \end{pmatrix}, \; B = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \; C = \begin{pmatrix} 1 & 1 \end{pmatrix}, \tag{88}$$

$$D = 2, \quad N = [N_1 \; N_2], \quad N_1 = [0.4 \; 0]^T, \quad N_2 = [0 \; 0.3]^T \tag{89}$$

Table 1 shows the eigenvalues of the true and estimated A and N in various cases. The row labelled '[10]' gives the result reported in [10] using the bilinear N4SID algorithm, with a white input, $k = 3$, and $j = 8191$. (This result appears to have been based on a single realisation of the input process.) 'Case I' is for a white input with uniform distribution, mean 0, standard deviation 0.5 and $\lambda = 0.83$ (see (4)). 'Case II' is for a white input with normal distribution $N(0, 0.01)$ and $\lambda = 0.73$. 'Case III' is for a coloured input with mean 0, standard deviation $\sigma_u = 0.08$, autocovariance $r_q = Eu_t u_{t+q} = 0.5^q \sigma_u^2$, and $\lambda = 0.93$. 'Case IV' is for a white input with exponential distribution with standard deviation 0.05 and $\lambda = 0.77$.

In all the cases I–IV the row block size is again $k = 3$, but the number of columns is only $j = 595$ (hence the sample size is $\tilde{N} = 2k + j - 1 = 600$), compared with $j = 8191$ ($\tilde{N} = 8196$), which was used in [10]. It can be seen that, with the smaller sample size, the bilinear N4SID algorithm gives significantly worse results than our Algorithms I and II, even when the input is white.

Table 2 shows how the eigenvalues of the estimated A and N depend on the 'row block size' k and on the number of columns j when the input of case I is used.

Step 2 of each algorithm we have proposed includes the selection of a suitable state dimension n. In principle this can be done by examining the singular values obtained in this step, since the number of non-zero singular values should indicate the correct state dimension. However, it is known from experience with the application of linear subspace methods that this is not a very reliable indicator in practice when noise is present, or when the data has not been generated by a system in the assumed model class. In this example there is no noise and the data has been generated by a bilinear system, but both Algorithms I and II.1 are known to be inaccurate (as shown by Theorems 2 and 4). It is therefore of interest to see the effect on the relative output error of various choices

	Algorithms	Eigenvalues of A	Eigenvalues of N
True		$\pm 0.5i$	0.4, 0.3
[10]	N4SID	$-0.0027 \pm 0.4975i$	0.4011, 0.3055
Case I	N4SID	$-0.0003 \pm 0.4923i(0.0108)$	0.1857(0.0474), 0.1409 (0.0325)
	Algorithm I	$0.0001 \pm 0.4998i(0.0058)$	0.3988(0.0087), 0.2996(0.0147)
	Algorithm II.1	$0.0041 \pm 0.4998i(0.0347)$	0.3959(0.0259), 0.2945 (0.0239)
Case II	N4SID	$-0.0012 \pm 0.4813i(0.0138)$	0.2800(0.0412), 0.2206(0.0237)
	Algorithm I	$0.0000 \pm 0.5000i(0.0047)$	0.3993(0.0129) 0.3009(0.0113)
	Algorithm II.1	$0.0002 \pm 0.4999i(0.0086)$	0.3962(0.0235), 0.2963(0.0127)
Case III	N4SID	$-0.0002 \pm 0.47742i(0.0387)$	0.3046(0.0358), 0.2446(0.0236)
	Algorithm I	$-0.0000 \pm 0.5000i(0.0456)$	0.3993(0.0102) 0.3018(0.0097)
	Algorithm II.1	$-0.0001 \pm 0.5000i(0.0637)$	0.4091(0.0237), 0.3004(0.0132)
Case IV	N4SID	$-0.0003 \pm 0.4927i(0.0382)$	0.1857(0.0397), 0.1409(0.0272)
	Algorithm I	$-0.0002 \pm 0.4999i(0.0157)$	0.3996(0.0147), 0.3089(0.0129)
	Algorithm II.1	$0.0002 \pm 0.4997i(0.0316)$	0.4104(0.0527), 0.3024(0.0321)

Table 1: Example 1: Results with different inputs and algorithms

of n, and compare these with the singular values. Table 3 shows, for different block sizes and algorithms, the effects of different choices of $\hat{n}$ (estimates of n) on the relative output-error. The singular values which are significantly different from 0 are also shown (as determined by the default criterion of Matlab's 'rank(M)' function for this purpose: σ_i is taken to be 0 if $\sigma_i < j \times \|M\|_2 \times \epsilon$, where ϵ is the machine precision). The system input in this case was the same as for Case I, but these results were obtained from only one realisation.

As to the order selection, apart from the method mentioned in Algorithm I, the relative output error is another criterion for choosing the model order. We take an example here for illustration. The system is the same as in Example 5.1.1 and two system inputs are used. In case I, the system input is $u_1$with mean -2.7691e-16, standard deviation 0.5 and $\|u_1\| = 12.2372$ and output y_1 with mean -0.0012, standard deviation 1.4454 and $\|y_1\| = 35.3742$. In case II, the system input is u_2 with mean 0.0051, standard deviation 0.0029 and $\|u_2\| = 0.1430$ and output y_2 with mean 0.0183, standard deviation 0.0083 and $\|y_2\| = 0.4919$. The effects of different order selections on the relative output error given in the following Table 3. In Table 3, only those singular values which are significantly different from 0 are shown. We use the default criterion of Matlab's 'rank(M)' function for this purpose: σ_i is taken to be 0 if $\sigma_i < j \times \|M\|_2 \times \epsilon$, where ϵ is the machine precision.

(k,j)	Algorithms	Eigenvalues of A	Eigenvalues of N
True		$\pm 0.5i$	0.4, 0.3
(2,297)	Algorithm I	$0.0008 \pm 0.5004i(0.0041)$	$0.3980(0.0077),\ 0.2994(0.0059)$
	Algorithm II.1	$-0.0121 \pm 0.4974i(0.0329)$	$0.3926(0.0198),\ 0.3014(0.0177)$
(3,295)	Algorithm I	$0.0002 \pm 0.4996i(0.0085)$	$0.3976(0.0095),\ 0.2984(0.0104)$
	Algorithm II.1	$0.0065 \pm 0.4993i(0.0423)$	$0.3918(0.0317),\ 0.2922(0.273)$
(2,597)	Algorithm I	$-0.0001 \pm 0.5005i(0.0027)$	$0.3929(0.0043),\ 0.2961(0.0038)$
	Algorithm II.1	$0.0030 \pm 0.5008i(0.0077)$	$0.3761(0.0120),\ 0.2700(0.0087)$
(3,595)	Algorithm I	$0.0001 \pm 0.4998i(0.0058)$	$0.3988(0.0087),\ 0.2996(0.0147)$
	Algorithm II.1	$0.0041 \pm 0.4998i(0.0347)$	$0.3959(0.0259),\ 0.2945(0.0239)$

Table 2: Example 1: Effect of block size and sample size

Example 2 The true system is:

$$A = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.3 \end{pmatrix}, B = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, C = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix},$$

$$D = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},\ N = [N_1\ N_2],\ N_1 = \begin{pmatrix} 0.6 & 0 \\ 0 & 0.4 \end{pmatrix},\ N_2 = \begin{pmatrix} 0.2 & 0 \\ 0 & 0.5 \end{pmatrix}$$

For this example the input was two-dimensional and coloured, with $Eu_i u_{i+q} = 0.9^q \sigma_u^2 I_2$, and $\lambda = 0.63$. The sample size was $\tilde{N} = 600$ and the row block size was $k = 2$ (hence $j = 597$). Table 4 summarises the results, including a comparison with the results obtained in [12], where $\tilde{N} = 4095$ and $k = 2$ were used. Since $l = n$, Algorithm II.2 can be used in this case. The rows labelled 'OLS' show the results obtained by using ordinary least-squares to solve (32) in Algorithms I and II.2, while those labelled 'CLS' show results obtained using constrained least-squares to take account of the known structure of the solution (the zero block). It can be seen that in this case the results do not depend much on either the choice of Algorithms I or II.2, or on the version of least-squares which is employed.

Example 3 The system is

$$A = \begin{pmatrix} 0.0251 & 0.4923 & -0.2620 \\ -0.5560 & -0.4083 & -0.2044 \\ 0.7928 & 0.3862 & 0.7832 \end{pmatrix}, B = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, C = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$$

$$D = 3, N_1 = \begin{pmatrix} 0.9023 \\ -0.1685 \\ -0.6581 \end{pmatrix}, N_2 = \begin{pmatrix} 0.5268 \\ 0.0768 \\ -0.6460 \end{pmatrix}, N_3 = \begin{pmatrix} 0.5340 \\ -0.1244 \\ -0.4791 \end{pmatrix}$$

A coloured input with mean 0, standard deviation 0.12, $Eu_t u_{t+q} = 0.9^q \sigma_u^2$, and $\lambda = 0.86$ was used. Algorithms I and II.1 were used, since $l < n$. Results with different block (k) and sample ($\tilde{N}$) sizes are given in Table 5. These results were obtained using a single realisation of the input.

k	$\hat{n}$	Algorithms	Singular values	Relative Output-error
2	1	Algorithm I	36.0948	0.0021
	2		9.2613	6.3818×10^{-8}
	1	Algorithm II.1	36.0903	0.0021
	2		9.3538	1.9242×10^{-6}
	3		3.3637	1.7164×10^{-6}
3	1	Algorithm I	37.11	0.0107
	2		9.22	7.09×10^{-8}
	3		0.91	6.41×10^{-8}
	4		0.43	6.4136×10^{-9}
	1	Algorithm II.1	37.10	0.0107
	2		9.21	2.98×10^{-7}
	3		1.42	2.96×10^{-7}
	4		1.30	3.0863×10^{-7}
	5		0.94	3.2179×10^{-7}
	6		0.62	3.2870×10^{-7}
	7		0.44	4.0066×10^{-7}

Table 3: Example 1: Effects of order selection

Note that with $k = 4$ and $\tilde{N} = 600$ the 'persistent excitation' rank conditions were not satisfied. This illustrates that, although a large row block size is desirable for high accuracy, it is limited by the available sample size. The likelihood of the rank condition failing increases as the row block size increases.

7.2 The stochastic case: Algorithms III and IV

Example 4 The system matrices are the same as in Example 1, but process and measurement noises are added. Four cases will be considered. Cases I and II will be with white measured input u_t, while Cases III and IV will be with coloured input ($Eu_k u_{k+q} = 0.5^q \sigma_u^2$). $\lambda = 0.79$ in all cases.

For cases I and III the covariance matrices are:

$$Q = \begin{pmatrix} 0.16 & 0 \\ 0 & 0.04 \end{pmatrix}, \; R = 0.09, \; S = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{90}$$

while for Cases II and IV they are 100 times smaller.

Table 6 shows the results with $k = 3$ and $\tilde{N} = 600$. The Table also shows the results reported in [10] for a single input realisation, with $\tilde{N} = 8191$.

Table 7 shows the results obtained from a single input realisation with various block (k) and sample

	Algorithm	eig(A)	eig(N_1)	eig(N_2)
True		0.5, 0.3	0.6, 0.4	0.2, 0.5
[10] ($\tilde{N} = 4095$)	N4SID	0.5001 0.2979	0.5994 0.4020	0.5016 0.1914
OLS ($\tilde{N} = 600$)	Algorithm I	0.5000(0.0001) 0.3000(0.0002)	0.6000(0.0012) 0.4000(0.0011)	0.5000(0.0011) 0.2000(0.0007)
	Algorithm II.2	0.5000(0.0001) 0.3000(0.0001)	0.6000(0.0011) 0.4000(0.0010)	0.5000 (0.0012) 0.2000(0.0009)
CLS ($\tilde{N} = 600$)	Algorithm I	0.5000(0.0001) 0.3000(0.0001)	0.6000(0.0009) 0.4000(0.0008)	0.5000(0.0007) 0.2000(0.0008)
	Algorithm II.2	0.5000(0.0001) 0.3000(0.0002)	0.6000(0.0012) 0.4000(0.0009)	0.4000(0.0008) 0.2000(0.0007)

Table 4: Example 2: Performance with different algorithms

$(k, \tilde{N})$	Algorithm	Eigenvalues of A	Eigenvalues of N
True		$\pm 0.5i$, 0.4	0.5, ± 0.2
(3,600)	Algorithm I Algorithm II.1	$-0.0025 \pm 0.4964i$, 0.4005 $0.0043 \pm 0.4896i$, 0.4025	0.4795, 0.2036 , -0.1981 0.4120, 0.1410, -0.2102
(4,600)	Algorithm I Algorithm II.1	condition (25) not met condition (37) not met	
(3,1000)	Algorithm I Algorithm II.1	$-0.0005 \pm 0.4989i$, 0.4016 $0.0018 \pm 0.4911i$, 0.4021	0.4881, 0.2051, -0.2100 0.4268, 0.1482, -0.2095
(4,1000)	Algorithm I Algorithm II.1	$-0.0019 \pm 0.4960i$, 0.4007 $0.0023 \pm 0.5006i$, 0.3999	0.4803, 0.1980, -0.1987 0.5149, 0.2170, -0.2100

Table 5: Example 3: Effect of block size and sample size.

$(\tilde{N})$ sizes.

Example 5 The system is the same as in Example 2. White process and measurement noises are added with covariances $Q = R = 0.01I_2$, $S = 0_{2,2}$.

Again four cases are considered. For Case V and VI, the input is white, with a two-dimensional uniform distribution. For Case VII and VIII, the input is coloured with $Eu_t u_{t+q} = 0.9^q \sigma_u^2 I_2$. In each case, $\tilde{N} = 1000$ and $k = 2$. Algorithm IV.2 is used, since $l = n$. Ordinary least-squares is used to solve (32) in Cases V and VII, while constrained least-squares is used in Cases VI and VIII. Note that in [10], the sample size was $\tilde{N} = 4095$. Table 8 shows the results for the various cases (for a single input realisation). Again it is seen that the choice of ordinary or constrained least-squares has little effect on the estimates. As remarked also in [10, 12], the estimates of Q, R and S are not

	Algorithms	Eigenvalues of A	Eigenvalues of N
True		$\pm 0.5i$	0.4, 0.3
[10]	N4SID	$-0.0027 \pm 0.4975i$	0.4011, 0.3055
Case I	N4SID	$-0.0171 \pm 0.4794i(0.0108)$	0.2769(0.0133), 0.2189(0.0102)
	Algorithm III	$-0.0078 \pm 0.4864i(0.0192)$	0.4128(0.0145), 0.3035(0.0157)
	Algorithm IV.1	$-0.0076 \pm 0.4860i(0.0174)$	0.3838(0.0174), 0.2829(0.0105)
Case II	N4SID	$-0.002 \pm 0.4817i(0.0137)$	0.2798(0.0102), 0.2200(0.0112)
	Algorithm III	$0.0000 \pm 0.5000i(0.0098)$	0.4005(0.0078), 0.3030(0.0083)
	Algorithm IV.1	$0.0043 \pm 0.5005i(0.0103)$	0.4128(0.0097), 0.2992(0.0085)
Case III	N4SID	$-0.0056 \pm 0.4274i(0.0625)$	0.2796(0.0138), 0.2063(0.0149)
	Algorithm III	$0.0089 \pm 0.4945i(0.0238)$	0.3906(0.0155), 0.3149(0.0234)
	Algorithm IV.1	$0.0044 \pm 0.4947i(0.0302)$	0.4048(0.0123), 0.2688(0.0201)
Case IV	N4SID	$-0.0155 \pm 0.4727i(0.0108)$	0.2779(0.0128), 0.2456(0.0137)
	Algorithm III	$0.0005 \pm 0.4980(0.0057)$	0.4006(0.0107), 0.2976(0.0103)
	Algorithm IV.1	$0.0015 \pm 0.4974(0.0098)$	0.3876(0.0257), 2893(0.0233)

Table 6: Example 4: Results with different inputs and signal-to-noise ratios

very accurate, because of the small value of k.

Example 6 The system is:

$$\begin{aligned} A &= \begin{pmatrix} 0 & 0.5 & 0 \\ -0.5 & 0 & 0 \\ 0 & 0 & 0.4 \end{pmatrix}, B = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, C = B^T, \\ D &= 3, \; N = \text{diag}[0.5, -0.2, 0.2] \end{aligned}$$

and the noise is the same as for Case II of Example 4. A single realisation of a coloured input with mean 0, variance 0.01, $Eu_tu_{t+q} = 0.5^q\sigma_u^2$ and $\lambda = 0.87$ was applied. Algorithm IV.1 was used, since $l < n$. Results with different block and sample sizes are given in Table 9.

7.3 Remarks on computational cost

Our new algorithms have considerably lower computational complexity than the algorithms proposed in [12]. The major computational load is involved in finding the right-inverse in (30) and (49). The row dimensions of the relevant matrices which appear in the algorithms presented here, in [12], and in [3], are shown in Table 10 for the three examples, where $k = 2$ for examples 1 and 2, and $k = 3$ for example 3. For the bilinear N4SID algorithm of [12] the row dimension is $(d_k + 2e_k + e_kd_k + e_k^2)$. In the case of our Algorithms I and III, the row dimension is $g_k = e_k + (m/2)(m+1)^k + l[(m+1)^k - 1] + e_k^2$. For our Algorithms II and IV, the row dimension

(k, j)	Algorithm	Eigenvalues of A	Eigenvalues of N
True		$\pm 0.5i$	0.4, 0.3
(2,297)	Algorithm III	$-0.0543 \pm 0.4628i$	0.4680, 0.2613
	Algorithm IV.1	$-0.1182 \pm 0.4725i$	0.4876, 0.2251
(3,295)	Algorithm III	$-0.0133 \pm 0.4744i$	0.4597, 0.2504
	Algorithm IV.1	$0.0163 \pm 0.4697i$	0.3606, 0.2492
(2,597)	Algorithm III	$0.0024 \pm 0.4966i$	0.4298, 0.3022
	Algorithm IV.1	0.0038 ± 0.5075	0.3725, 0.2842
(3,597)	Algorithm III	$0.0005 \pm 0.4980i$	0.4006, 0.2976
	Algorithm IV.1	0.0015 ± 0.4974	0.3876, 0.2893

Table 7: Example 4: Effect of sample size and block size

	eig(A)	$eig(N_1)$	$eig(N_2)$	eig(Q)	eig(R)	eig(S)
Original	0.5, 0.3	0.6, 0.4	0.5, 0.2	0.01, 0.01	0, 0	0.01, 0.01
N4SID [10]	0.5001	0.5994	0.5016	N.A.	N.A.	N.A.
($\tilde{N} = 4095$)	0.2979	0.4020	0.1914			
Case V	0.4998	0.5998	0.5000	0.0013	0.0477	0.0001
($\tilde{N} = 1000$)	0.3002	0.4000	0.2001	0.0009	0.0197	0.0001
Case VI	0.5004	0.5998	0.4999	0.6732×10^{-3}	0.0675	0.0014
($\tilde{N} = 1000$)	0.2900	0.3997	0.1997	0.4051×10^{-3}	0.0246	-0.0013
Case VII	0.4992	0.6028	0.5070	0.6681×10^{-5}	0.0857	0.8975×10^{-4}
($\tilde{N} = 1000$)	0.2968	0.4007	0.2019	0.0141×10^{-5}	0.0403	0.0413×10^{-4}
Case VIII	0.5000	0.6000	0.5003	0.7542×10^{-5}	0.1030	0.9754×10^{-4}
($\tilde{N} = 1000$)	0.2996	0.3998	0.2000	0.0006×10^{-5}	0.0527	0.0322×10^{-4}

Table 8: Example 5: Comparisons with different inputs and least-squares algorithms

is $h_k = e_k + e_k^2$ for Examples 1 and 3 ($l < n$), and $f_k = e_k + (m/2)(m+1)^k + l[(m+1)^k - 1]$ for Example 2 ($l = n$).

We find the performances of Algorithms I and III are a little better than those of Algorithms II and IV in the case of $l < n$, and a little worse in the case of $l \geq n$. This is in line with expectations. In the case of $l < n$ one can trade off the better accuracy of Algorithms I and III against the lower computational cost of Algorihtms II.1 and IV.1.

Although increasing the block size k should give more accurate estimation, experience shows that large k not only causes high computational cost, but also increases the likelihood that the persistent excitation (rank) conditions in Theorems 2, 4, 6, 8, 10, and 12 will not be met.

$(k, \tilde{N})$	Eigenvalues of A	Eigenvalues of N
True	$\pm 0.5i$, 0.4	0.5, ± 0.2
(3,800)	$0.00 \pm 0.49i$, 0.29	0.47, 0.17, -0.08
(4,800)	$-0.01 \pm 0.49i$,0.41	0.47, 0.22, -0.12
(3,1200)	$0.00 \pm 0.50i$, 0.40	0.49, 0.19, -0.09
(4,1200)	$0.00 \pm 0.50i$,0.40	0.48, 0.22, -0.20
(3,1500)	$0.00 \pm 0.50i$, 0.40	0.49, 0.18, -0.12
(4,1500)	$0.00 \pm 0.50i$, 0.40	0.51, 0.19, -0.19

Table 9: Example 6: Effect of block size and sample size

	Algorithms I and III	Algorithms II and IV	N4SID
Example 1	17	12	27
Example 2	97	33	152
Example 3	67	56	119

Table 10: Comparison of row dimensions of matrices for Examples 1–3 and various algorithms

8 Conclusion

Some subspace algorithms for the identification of bilinear systems have been developed. Their main advantage is that the system input does not have to be white. All the algorithms proposed here also have lower computational complexity than previously proposed algorithms, because the dimensions of the matrices involved in them are much smaller. Their wider applicability has been demonstrated by several examples, which also show that even with coloured inputs the new algorithms converge to correct estimates relatively quickly. The presumed reason for this is that, since the algorithms do not depend on whiteness of the input, they are insensitive to the large errors in the sample spectrum which are inevitable with small sample sizes.

9 Acknowledgement

The work reported in this paper was supported by the UK Engineering and Physical Science Research Council under Research Grant GR/M08332, and by the European Research Network on System Identification (ERNSI) under TMR contract ERB FMRX CT98 0206. We would like to thank Vincent Verdult for providing us with his software implementing the bilinear N4SID algorithm described in [10].

References

[1] Caines P.E., *Linear Stochastic Systems*, Wiley, 1988.

[2] Chen H. and Maciejowski J.M., Subspace identification of deterministic bilinear systems, *ACC 2000, Chicago, Illinois, USA*, June 2000.

[3] Chen H. and Maciejowski J.M., Subspace identification of combined deterministic-stochastic bilinear systems, *IFAC Symp. on System Identification, SYSID 2000, Santa Barbara*, June 2000.

[4] Chen H. and Maciejowski J.M., An Improved Subspace Identification Method for Bilinear Systems, *39th IEEE Conference on Decision and Control, Sydney, Australia*, December 2000.

[5] Chou C.T. and Maciejowski J.M., System Identification Using Balanced Parameterizations, *IEEE Trans. Auto. Contr.*, Vol. 42, No. 7. July, 1997 pp.956-974.

[6] Chui N.L.C. and Maciejowski J. M., *Subspace Identification – a Markov Parameter Approach* Technical Report CUED/F-INFENG/TR.337, 1998 Cambridge University

[7] Chen H., Zinober A.S.I. and Ruan R., Strong Consistency and Convergence Rate of Parameter Identification for Bilinear Systems, *Int.J.Control*, 1996, vol. 63, No.5 pp. 907-919

[8] Desai U. B., *Realization of Bilinear Stochastic Systems* IEEE Trans on AC, Vol. AC-31, No.2, Feb, 1986.

[9] Favoreel W, *Subspace Methods for Identification and Control of Linear and Bilinear Systems*, Ph.D Thesis, Katholieke Universiteit Leuven, November, 1999.

[10] Favoreel W, De Moor B. and Van Overschee P., *Subspace identification of bilinear systems subject to white inputs*, ESAT-SISTA/TR 1996-53I, Katholieke Universiteit Leuven, 1996.

[11] Favoreel W, De Moor B. and Van Overschee P., Subspace identification of Balanced deterministic bilinear systems subject to white inputs, *Proc. European Ccontrol Conf, Brussels*, 1997.

[12] Favoreel W and De Moor B, Subspace identification of bilinear systems, *Proc. MTNS, Padova*, 1998. pp. 787-790.

[13] Favoreel W, De Moor B. and Van Overschee P., Subspace identification of bilinear systems subject to white inputs, *IEEE Trans. Auto. Contr.*, Vol 44., No. 6, June 1999. pp. 1157–1165.

[14] Fnaiech F. and Ljung L., Recursive Identification of Bilinear System, *Int.J. Control*, vol. 45, pp. 453-470., 1987.

[15] Isidori A., *Nonlinear Control Systems* Springer-Verlag, 1989

[16] Khatri C.G. and Rao C.R., Solutions to some functional equations and their applications to characterization of probability distributions, *Sankhya : The Indian J. Stat., series A*, 30, pp. 167-180, 1968.

[17] Krener A.J., Bilinear and nonlinear realizations of input-output maps. *SIAM Journal on Control*, vol. 13, pp.827-834., 1975.

[18] Larimore,W.E, System identification, reduced-order filtering and modeling via canonical variate analysis, *Proc. American Contr.Conf., San Francisco*, (1983).

[19] Papoulis, A, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, 1984.

[20] Van Overschee P and De Moor B, N4SID: subspace algorithms of combined deterministic and stochastic systems, *Automatica*, vol.30, No. 1, pp. 75-93, 1994

[21] Suda N., Kodama S. and Ikeda M., *Matrix Theory in Automatical Control,* Japanese Automatical Control Association, 1973.

[22] Verhaegen,M, and Dewilde,P.M, Subspace model identification, Parts 1 and 2, *Int. J. Contr.*, **56**, 1187-1241, (1992).

[23] Verdult V, Verhaegen M, Chou C.T, Lovera, *Proc. IEEE CDC Conf.,* 1998.

[24] Verdult V., and Verhaegen M., *Subspace-based Identification of MIMO Bilinear Systems*, ECC 1999, 31.August-3.September, 1999, Karlsruhe, Germany

10 Appendix

Proof of Lemma 1:

The proof of (5) is in [21]. Now we prove (6). Let

$$G = [g_1, g_2, ..., g_m], \; J = [j_1, j_2, ..., j_m]$$

From (5), and the definitions of the Kronecker and Khatri-Rao products, we obtain

$$\begin{aligned}
(FG \odot HJ) &= [Fg_1 \otimes Hj_1, Fg_2 \otimes Hj_2, \ldots, Fg_m \otimes Hj_m] \\
&= (F \otimes H)\,[g_1 \otimes j_1, g_2 \otimes j_2, \ldots, g_m \otimes j_m] \\
&= (F \otimes H)(G \odot J)
\end{aligned}$$

This proves Lemma 1. ∎

Proof of Lemma 2:

We prove Lemma 2 by induction.

From the definition of $U^{+}_{q|q}$ and $U^{++}_{q|q}$, we know that $\mathcal{U}^{+}_{q|q} \odot \mathcal{U}_q = \mathcal{U}_q \odot \mathcal{U}_q$. First we prove that $\mathcal{U}^{+}_{q|q} \odot \mathcal{U}_q = \mathcal{U}_q \odot \mathcal{U}_q = \mathcal{U}^{++}_{q|q}$

$$
\begin{aligned}
U_q \odot U_q &= \begin{pmatrix} U_{q,1} \odot U_q \\ U_{q,2} \odot U_q \\ \vdots \\ U_{q,m} \odot U_q \end{pmatrix} \\
&= \begin{pmatrix} U_{q,1} \odot U_{q,1} \\ U_{q,1} \odot U_{q,2} \\ \vdots \\ U_{q,1} \odot U_{q,m} \\ U_{q,2} \odot U_{q,1} \\ U_{q,2} \odot U_{q,2} \\ \vdots \\ U_{q,2} \odot U_{q,m} \\ \vdots \\ U_{q,m} \odot U_{q,1} \\ U_{q,m} \odot U_{q,2} \\ \vdots \\ U_{q,m} \odot U_{q,m} \end{pmatrix}
\end{aligned} \tag{91}
$$

Since for any i, j, we have $U_{q,i} \odot U_{q,j} = U_{q,j} \odot U_{q,i}$, by removing some repeated rows from matrix (91), we see that $\mathcal{U}^{+}_{q|q} \odot \mathcal{U}_q = \mathcal{U}_q \odot \mathcal{U}_q = \mathcal{U}^{++}_{q|q}$

Now we prove that Lemma 2 holds for $i = 1$.

$$
\mathcal{U}^{+}_{1+q|q} \odot \mathcal{U}_q = \begin{pmatrix} \mathcal{U}^{+}_{q|q} \odot \mathcal{U}_q \\ \mathcal{U}_{q+1} \odot \mathcal{U}_q \\ \mathcal{U}_{q+1} \odot \mathcal{U}_q \odot \mathcal{U}_q \end{pmatrix}
$$

and

$$
\mathcal{U}^{++}_{1+q|q} = \begin{pmatrix} \mathcal{U}^{++}_{q|q} \\ \mathcal{U}_{q+1} \odot \mathcal{U}^{++}_{q|q} \end{pmatrix}
$$

We have

$$
\begin{aligned}
(\mathcal{U}^{+}_{1+q|q} \odot \mathcal{U}_q) \setminus \mathcal{U}^{++}_{1+q|q} &= \begin{pmatrix} \mathcal{U}_{q+1} \odot \mathcal{U}_q \end{pmatrix} \\
&= \mathcal{U}^{+}_{q+1|q+1} \odot \mathcal{U}_q \\
&= \mathcal{U}_{q+1|q+1} \odot \mathcal{U}_q \subset \mathcal{U}_{q+1|q}
\end{aligned}
$$

Now suppose that Lemma 2 holds for $i = n$. Then for $i = n + 1$,

$$\mathcal{U}^+_{n+1+q|q} \odot \mathcal{U}_q = \begin{pmatrix} \mathcal{U}^+_{n+q|q} \odot \mathcal{U}_q \\ \mathcal{U}_{n+q+1} \odot \mathcal{U}_q \\ \mathcal{U}_{n+q+1} \odot \mathcal{U}^+_{n+q|q} \odot \mathcal{U}_q \end{pmatrix}$$

and

$$\mathcal{U}^{++}_{n+1+q|q} = \begin{pmatrix} \mathcal{U}^{++}_{n+q|q} \\ \mathcal{U}_{n+q+1} \odot \mathcal{U}^{++}_{n+q|q} \end{pmatrix}$$

Since $\mathcal{U}^+_{n+q|q} \odot \mathcal{U}_q \setminus \mathcal{U}^{++}_{n+q|q} = \mathcal{U}^+_{n+q|q+1} \odot \mathcal{U}_q$, hence

$$\begin{aligned}(\mathcal{U}^+_{n+q+1|q} \odot \mathcal{U}_q) \setminus \mathcal{U}^{++}_{n+1+q|q} &= \begin{pmatrix} \mathcal{U}^+_{n+q|q+1} \odot \mathcal{U}_q \\ \mathcal{U}_{n+q+1} \odot \mathcal{U}_q \\ \mathcal{U}_{n+q+1} \odot \mathcal{U}^+_{n+q|q+1} \odot \mathcal{U}_q \end{pmatrix} \\ &= \mathcal{U}^+_{n+q+1|q+1} \odot \mathcal{U}_q\end{aligned}$$

From the definition and structure of $\mathcal{U}_{i-1|q}$, we have the following fact: for all j, j_1, j_2 such that $q \le j_1 \le j_2 \le j \le i-1$

$$\begin{aligned}\mathcal{U}_j &\subset \mathcal{U}_{i-1|q} \\ \mathcal{U}_{j_2} \odot \mathcal{U}_{j_1} &\subset \mathcal{U}_{i-1|q}\end{aligned}$$

since

$$\begin{aligned}\mathcal{U}^+_{n+q|q+1} \odot \mathcal{U}_q \subset \mathcal{U}^+_{n+q|q} &\subset \mathcal{U}^+_{n+q+1|q} \\ \mathcal{U}_{n+q+1} \odot \mathcal{U}_q &\subset \mathcal{U}^+_{n+q+1|q} \\ \mathcal{U}_{n+q+1} \odot \mathcal{U}^+_{n+q|q+1} \odot \mathcal{U}_q \subset \mathcal{U}_{n+q+1} \odot \mathcal{U}^+_{n+q|q} &\subset \mathcal{U}^+_{n+q+1|q}\end{aligned}$$

then $\mathcal{U}^+_{n+q+1|q+1} \odot \mathcal{U}_q \subset \mathcal{U}^+_{n+q+1|q}$

This proves Lemma 2. $\blacksquare$

Proof of Lemma 3

We prove Lemma 3 by induction.

First we prove that Lemma 3 holds for $k = 1$:

$$\begin{aligned}X_{j|j} &= \begin{pmatrix} X_j \\ U_j \odot X_j \end{pmatrix} \\ &= \begin{pmatrix} X_j \\ U^+_{j|j} \odot X_j \end{pmatrix}\end{aligned}$$

So,

$$\begin{aligned}\mathcal{X}_{j|j} &= \begin{pmatrix} \mathcal{X}_j \\ \mathcal{U}_j \odot \mathcal{X}_j \end{pmatrix} \\ &= \begin{pmatrix} \mathcal{X}_j \\ \mathcal{U}^+_{j|j} \odot \mathcal{X}_j \end{pmatrix}\end{aligned}$$

Suppose that Lemma 3 holds for $k = n$. Then, for $k = n + 1$,

$$\begin{aligned}\mathcal{X}_{n+j|j} &= \begin{pmatrix} \mathcal{X}_{n-1+j|j} \\ \mathcal{U}_{n+j} \odot \mathcal{X}_{n-1+j|j} \end{pmatrix} \\ &= \begin{pmatrix} \mathcal{X}_j \\ \mathcal{U}^+_{n-1+j|j} \odot \mathcal{X}_j \\ \mathcal{U}_{n+j} \odot \mathcal{X}_j \\ \mathcal{U}_{n+j} \odot \mathcal{U}^+_{n-1+j|j} \odot \mathcal{X}_j \end{pmatrix} \\ &= \begin{pmatrix} \mathcal{X}_j \\ \mathcal{U}^+_{n+j|j} \odot \mathcal{X}_j \end{pmatrix}\end{aligned}$$

This proves Lemma 3. ∎

Proof of Lemma 8:

Since $X = PY + M$, and Y is full row rank, there exist matrices L, Q, such that

$$Y = LQ = [L_1 \;\; 0] \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} \tag{92}$$

where Q is an orthogonal matrix. From (92), we have $\mathcal{Y} = \mathcal{Q}_1$ and $\mathcal{Y}^\perp = \mathcal{Q}_2$, where $\mathcal{Y}, \mathcal{Q}_1$ and $\mathcal{Q}_2$ denote spaces spanned by the rows of matrices Y, Q_1 and Q_2, respectively.

$$\begin{aligned}\|\Pi_{\mathcal{Y}^\perp} X\|_1 &= \|\Pi_{\mathcal{Q}_2} X\|_1 \\ &= \|\Pi_{\mathcal{Q}_2} M\|_1 = \|M Q_2^T Q_2\|_1 = o(\delta)\end{aligned}$$

Lemma 8 is thus proved. ∎

Proof of Lemma 9:

From (18),

$$\begin{aligned}\Pi_{Y_i} X_i &= X_i Y_i^T (Y_i Y_i^T)^{-1} Y_i \\ &= \frac{1}{j+1} X_i Y_i^T (\frac{1}{j+1} Y_i Y_i^T)^{-1} Y_i\end{aligned}$$

Hence, by ergodicity

$$\begin{aligned}\Pi_{Y_i} X_i &\to (E x_0 y_0^T)(E y_0 y_0^T)^{-1} Y_i \quad w.p.1 \quad as \quad j \to \infty \\ &= 0 \quad w.p.1 \quad as \quad j \to \infty\end{aligned}$$

where the last step follows from the uncorrelatedness of x_n and y_n. ∎