

Subspace Identification – A Markov Parameter Approach *

Nelson L.C. Chui J.M. Maciejowski
Cambridge University Engineering Department
Cambridge, CB2 1PZ, England.

22 December 1998

*Technical report CUED/F-INFENG/TR.337
Submitted to IEEE Transactions on Automatic Control.*

Abstract

Estimating observability matrices or state sequences is the central component of existing subspace identification methods. In this paper a different approach, in which Markov parameters are first estimated under general input excitation, is proposed. The prominent difference of this approach is that a three-block arrangement of data matrices is used. It is shown that one advantage of this approach over other subspace algorithms is that several unbiased estimating procedures can be carried out. One immediate application is to obtain balanced or nearly balanced models directly from the estimated Markov parameters. Another application is that with the estimated Markov parameters, consistently initialised Kalman filter state sequences can be obtained, from which the system matrices can be easily determined without bias. Performance of the proposed algorithms is investigated in two case studies which are based on real data taken from two industrial systems.

Keywords: System Identification, Modeling, Subspace Methods, Markov Parameters.

1 Introduction

Various versions of subspace methods for identifying discrete-time linear systems in state-space form have been derived in recent years. Commonly known subspace based algorithms include CVA [19], N4SID [25], MOESP [29, 28], and IV-4SID [30]. All of these methods first estimate the range space of the observability matrix, and then obtain the matrices of the state-space form either by estimating the observability matrix, or by estimating the state sequence. A unified treatment of most of these algorithms has been given in [26], where it is shown that each variant corresponds to a different choice of certain weighting matrices. Some statistical analysis of subspace algorithms is available in [23, 31, 2] and references therein.

*Corresponding author Dr. J.M. Maciejowski. Tel +44 1223 332732 Fax +44 1223 332662; E-mail jmm@eng.cam.ac.uk.

The basis of subspace algorithms is exploitation of the concept of the state as a finite-dimensional ‘interface’ between the past and the future; a concept which dates back to [22] in the deterministic case and to [1] in the stochastic case. An intuitive description of their operation is the following. The input-output data is arranged into two distinct ‘blocks’, one of which can be thought of as ‘inputs and past outputs’ and the other as ‘future outputs’. (The first block can also be thought of as ‘regressors’ in the statistical sense, when there are no measured inputs.) The data in the second block is projected onto the space spanned by the data in the first block. Orthogonal projection corresponds to least-squares prediction of the output, which can be ‘factorized’ into estimation of the observability matrix and estimation of the state sequence. If both measured inputs and unmeasured disturbances are present, further processing of the projected data is necessary, in order to distinguish between the effects of these two kinds of signal. Thus in [26], for instance, one finds a relatively simple algorithm which is known to give biased estimates of the system matrices (unless the measured inputs are white), and a considerably more complex algorithm which is known to be asymptotically unbiased, as the data ‘blocks’ become *doubly* infinite.

In this paper we introduce a class of subspace algorithms in which the data is arranged into *three* blocks rather than two. The additional block can be thought of as a source of ‘instrumental variables’, which are used to remove the effects of the unmeasured noise sources, and thus allow the estimation of some initial Markov parameters (impulse response coefficients) of the system. These Markov parameters are used to modify existing approaches, so that unbiased estimates of the system matrices are obtained, even if the row-dimension of the blocks remains finite. This dimension can be thought of as the length of a ‘sliding window’ passed over the data, which limits the correlation or ‘memory’ length used in subspace algorithms. Thus our three-block approach allows unbiased estimates to be obtained with finite ‘window lengths’, in the presence of arbitrary measured inputs. The tools used in our approach are essentially the same (projections, QR factorizations, etc) as in the existing subspace methods.

The introduction of Markov parameters and of instrumental variables into subspace identification is not new. In particular, the way these appear in [29, 28] is very similar to the way they appear in this paper. But [29] is limited to the ‘output error’ noise structure, while [28] is limited to white noise inputs, and deals with the identification of the deterministic part of the system only. In this paper careful analysis of the relationships between several subspaces which arise in the combined deterministic-stochastic case, with a general noise structure, allows us to give a complete solution for the general case.

The paper is organized as follows. The identification problem is defined in Section 2, and some notations are introduced in 3. Section 4 considers identification of deterministic systems. Section 5 then considers the general case, and introduces the 3-block arrangement of data. Section 6 shows how initial Markov parameters can be estimated in the general case. Section 7 considers the estimation of the state sequences X_k and X_{k+1} , paying particular attention to the requirement that the estimated initial conditions of these sequences are consistent with each other — failure to meet this requirement has been the source of biased estimates in some subspace algorithms. In Section 8, three alternative approaches for identifying the deterministic (input-

output) part of the system are suggested, and Section 9 discusses the estimation of the stochastic part. Numerically efficient implementation is developed in Section 10. Finally, two cases based on industrial data are studied in Section 11.

Some of the results in this paper were previously reported in [10].

2 Problem Setup

Throughout this paper, the sets of integers and non-negative integers are denoted by \mathbb{Z} and \mathbb{Z}_+ , respectively. The Moore-Penrose inverse is written as \cdot^\dagger while the Hermitian as \cdot^* . Denote by $+$, \oplus and \cap the sum, the direct sum and the intersection of two vector spaces. The notation \cdot^\perp denotes the orthogonal complement of a subspace with respect to the predefined ambient space.

Consider a linear time-invariant system with the following state-space realization:

$$x(t+1) = Ax(t) + Bu(t) + w(t), \quad (1a)$$

$$y(t) = Cx(t) + Du(t) + v(t), \quad (1b)$$

where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$, $D \in \mathbb{R}^{p \times m}$. The input and output signals are denoted by u and y , respectively; the process disturbance and output noise are denoted by w and v , respectively. We further assume that u , y , w and v are signals in an ideal probability space. That is, for any instance $t \in \mathbb{Z}_+$, $x(t)$, $u(t)$, $y(t)$, $w(t)$ and $v(t)$ are vectors of real Lebesgue square integrable random variables. We assume that u , y , w and v are stationary, which implies that A has all its eigenvalues strictly inside the unit disk. In addition, \mathbf{E} denotes the usual expectation operator. When used with lower-case letters, u_t , y_t , etc will denote data collected at time t from a particular realization of $u(t)$, $y(t)$, etc.

The identification problem which we address is: Given a sample of a particular realization of input-output data, $\{u_t, y_t : t = 0, \dots, N\}$, estimate the state dimension n , the matrices A , B , C , and D , the Kalman gain K , and the covariance of the process $e(t)$ in the innovations representation

$$x(t+1) = Ax(t) + Bu(t) + Ke(t), \quad (2a)$$

$$y(t) = Cx(t) + Du(t) + e(t), \quad (2b)$$

We assume that N is sufficiently large, that all required sample statistics of the data are arbitrarily close to the population statistics.

Suppose that the system is comprised of two uncorrelated subsystems, a deterministic one denoted by a superscript d , and a stationary stochastic one denoted by a superscript s :

$$x(t) = x^d(t) + x^s(t),$$

$$y(t) = y^d(t) + y^s(t).$$

The deterministic subsystem, describing the behavior due to the input u , has the state-space equations:

$$x^d(t+1) = Ax^d(t) + Bu(t), \quad (3a)$$

$$y^d(t) = Cx^d(t) + Du(t); \quad (3b)$$

whereas the stationary stochastic subsystem, describing the behavior due to the process and output noises w and v , has the state-space equations:

$$x^s(t+1) = Ax^s(t) + w(t), \quad (4a)$$

$$y^s(t) = Cx^s(t) + v(t). \quad (4b)$$

Let the process noises w and v have the following correlation matrices:

$$\mathbf{E} \left[\begin{pmatrix} w(t) \\ v(t) \end{pmatrix} \begin{pmatrix} w(\tau) \\ v(\tau) \end{pmatrix}^* \right] =: \begin{pmatrix} \Sigma^w & \Sigma^{wv} \\ \Sigma^{vw} & \Sigma^v \end{pmatrix} \delta_{t\tau}, \quad (5)$$

for all $t, \tau \in \mathbb{Z}_+$, where δ denotes the Kronecker delta. Moreover, define the correlation matrices Σ^s , Λ_τ and G as:

$$\begin{aligned} \Sigma^s &:= \mathbf{E}(x^s(t) [x^s(t)]^*), \\ \Lambda_\tau &:= \mathbf{E}(y^s(t+\tau) [y^s(t)]^*), \\ G &:= \mathbf{E}(x^s(t+1) [y^s(t)]^*), \end{aligned}$$

where $t \geq \tau$, $t \in \mathbb{Z}_+$, $\tau \in \mathbb{Z}$. Since the stochastic subsystem is assumed stationary, Σ^w , Σ^v , Σ^{wv} , Σ^{vw} , Σ^s , Λ_τ and G are all constant matrices.

3 System in Block Equation Form

The notation introduced in this section mostly follows [25] closely. Let U_t be a matrix composed of a sequence of the input signal:

$$U_t := [u(t) \quad u(t+1) \quad \cdots \quad u(t+q-1)],$$

for some positive integer q . Using similar definitions for Y_t , X_t , etc, it is easy to see that (1) can also be written as:

$$X_{t+1} = AX_t + BU_t + W_t, \quad (6a)$$

$$Y_t = CX_t + DU_t + V_t. \quad (6b)$$

Now, consider the following block-Hankel matrix constructed from the input signal u for some k :

$$\begin{bmatrix} U_p \\ \hline U_f \end{bmatrix} := \begin{bmatrix} u(0) & u(1) & \cdots & u(q-1) \\ \ddots & \ddots & \ddots & \ddots \\ u(k-1) & u(k) & \cdots & u(k+q-2) \\ \hline u(k) & u(k+1) & \cdots & u(k+q-1) \\ \hline \ddots & \ddots & \ddots & \ddots \\ u(2k-1) & u(2k) & \cdots & u(2k+q-2) \end{bmatrix}.$$

Define $Y_p, Y_f, W_p, W_f, V_p,$ and V_f in a similar way, and define X_p and X_f slightly differently as:

$$\begin{bmatrix} X_p \\ X_f \end{bmatrix} := \begin{bmatrix} x(0) & x(1) & \cdots & x(q-1) \\ x(k) & x(k+1) & \cdots & x(k+q-1) \end{bmatrix}.$$

such that each is a sequence of one block row. At this point, we shall introduce a few more matrices. Define the deterministic controllability matrix \mathcal{C}_i^d , the stochastic controllability matrix \mathcal{C}_i^s , and \mathcal{C}_i^w as:

$$\begin{aligned} \mathcal{C}_i^d &:= [A^{i-1}B \quad A^{i-2}B \quad \cdots \quad B], \\ \mathcal{C}_i^s &:= [A^{i-1}G \quad A^{i-2}G \quad \cdots \quad G], \\ \mathcal{C}_i^w &:= [A^{i-1}I \quad A^{i-2}I \quad \cdots \quad I]. \end{aligned}$$

Furthermore, define the observability matrix \mathcal{O}_i , and Toeplitz matrices \mathcal{T}_i^d and \mathcal{T}_i^w as:

$$\mathcal{O}_i := \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{i-1} \end{bmatrix}, \quad \mathcal{T}_i^d := \begin{bmatrix} D & & 0 \\ CB & D & \\ \vdots & \ddots & \ddots \\ CA^{i-2}B & \cdots & CB & D \end{bmatrix}, \quad \mathcal{T}_i^w := \begin{bmatrix} 0 & & 0 \\ C & 0 & \\ \vdots & \ddots & \ddots \\ CA^{i-2} & \cdots & C & 0 \end{bmatrix}.$$

Note that D, CB, CAB, \dots are the Markov parameters of the system, which will also be denoted by h_0, h_1, h_2, \dots in this paper. With these new matrices, it is easy to derive the following ‘block form’ of the system equations:

$$X_f = A^k X_p + \mathcal{C}_k^d U_p + \mathcal{C}_k^w W_p, \tag{7a}$$

$$Y_p = \mathcal{O}_k X_p + \mathcal{T}_k^d U_p + \mathcal{T}_k^w W_p + V_p. \tag{7b}$$

To facilitate the analysis, denote by \mathcal{U}_p the space spanned by all the rows of the block Hankel matrix U_p . That is,

$$\mathcal{U}_p := \text{span}\{ \alpha^* U_p \mid \alpha \in \mathbb{R}^{km} \}.$$

We will use similar notation to represent the row spaces spanned by other block-Hankel matrices. Finally, let Π be the orthogonal projection operator. In this paper, we will use the shorthand

$\Pi_{\mathcal{R}}Q$ to represent the orthogonal projection of each row of Q onto the space \mathcal{R} . For instance, $\Pi_{\mathcal{U}_p}X_0$ is equivalent to:

$$\Pi_{\mathcal{U}_p}X_0 := \begin{bmatrix} \Pi_{\mathcal{U}_p}X_{0;1} \\ \vdots \\ \Pi_{\mathcal{U}_p}X_{0;n} \end{bmatrix}, \quad \text{for} \quad X_0 = \begin{bmatrix} X_{0;1} \\ \vdots \\ X_{0;n} \end{bmatrix}.$$

Moreover, this projection also is equivalent to $\mathbf{E}(X_0U_p^*) \cdot \mathbf{E}(U_pU_p^*)^\dagger \cdot U_p$. Finally, it can be seen without difficulty that $\mathbf{E}(\mathcal{O}_hX_0U_p^*) \cdot \mathbf{E}(U_pU_p^*)^\dagger \cdot U_p$ can also be written as $\mathcal{O}_h\mathbf{E}(X_0U_p^*) \cdot \mathbf{E}(U_pU_p^*)^\dagger \cdot U_p$, or equivalently, $\Pi_{\mathcal{U}_p}\mathcal{O}_hX_0 = \mathcal{O}_h\Pi_{\mathcal{U}_p}X_0$. In other words, the projection operation commutes with real matrix multiplication.

We remark that we shall abuse terminology slightly by speaking of subspaces having ‘empty intersection’ to mean that their intersection contains only the singleton $\{0\}$.

4 Deterministic Identification

First, deterministic identification via Markov parameter (MP) estimation is covered. In the deterministic setup, it will be seen that a two-block configuration is adequate. We shall assume k is greater than the observability index of the system. The data equations have the following form:

$$Y_p = \mathcal{O}_kX_p + \mathcal{T}_k^dU_p; \tag{8}$$

$$Y_f = \mathcal{O}_kX_f + \mathcal{T}_k^dU_f. \tag{9}$$

In addition, the state equation linking the past and future data equations can be written as:

$$X_f = A^kX_p + \mathcal{C}_k^dU_p. \tag{10}$$

It is well known that the quality of a model obtained from an identification experiment depends highly on the degree of excitation of the input signal. Such consideration leads to the study of informative experiments, which are identification experiments which contain sufficient information to discriminate between different models in an intended model set [20]. The study of informative experiments for subspace methods is beyond the scope of this paper, but can be found in [7] (and the references therein) for the deterministic case and [5, 8] (and the references therein) for the combined deterministic-stochastic case. Instead, in this paper we will make certain key assumptions about the state sequences and the input sequences, which enable correct results to be obtained using the algorithms developed here. First, we assume the following for the deterministic case:

$$\mathbf{E} \left[\begin{pmatrix} X_p \\ U_p \\ U_f \end{pmatrix} \begin{pmatrix} X_p \\ U_p \\ U_f \end{pmatrix}^* \right] > 0. \tag{11}$$

With this assumption, the following lemma holds.

Lemma 4.1. *Suppose (11) holds. Then,*

$$\begin{aligned}\mathcal{X}_f &\subset \mathcal{Y}_p + \mathcal{U}_p, \\ X_f + \mathcal{U}_f = \mathcal{X}_f \oplus \mathcal{U}_f &\subset (\mathcal{Y}_p + \mathcal{U}_p) \oplus \mathcal{U}_f.\end{aligned}$$

Proof. From Equation (8), it is easy to see that $\mathcal{Y}_p \subset \mathcal{X}_p + \mathcal{U}_p$. On the other hand, rewriting Equation (8) as $\mathcal{O}_k X_p = Y_p - \mathcal{T}_k^d U_p$ gives $\mathcal{X}_p \subset \mathcal{Y}_p + \mathcal{U}_p$, since \mathcal{O}_k is injective. Now, adding \mathcal{U}_p to both sides of these inclusions gives

$$\mathcal{Y}_p + \mathcal{U}_p \subset \mathcal{X}_p + \mathcal{U}_p, \quad \text{and} \quad \mathcal{X}_p + \mathcal{U}_p \subset \mathcal{Y}_p + \mathcal{U}_p.$$

Thus, clearly $\mathcal{Y}_p + \mathcal{U}_p = \mathcal{X}_p + \mathcal{U}_p$. In addition, with Equation 10 we then have

$$\mathcal{X}_f \subset \mathcal{X}_p + \mathcal{U}_p = \mathcal{Y}_p + \mathcal{U}_p.$$

Finally, (11) guarantees the direct sum property that $\mathcal{X}_f + \mathcal{U}_f = \mathcal{X}_f \oplus \mathcal{U}_f$. Thus, the lemma follows. \square

Lemma 4.1 states that the space \mathcal{X}_f can be observed from $\mathcal{Y}_p + \mathcal{U}_p$ and has empty intersection with \mathcal{U}_f . Therefore, Y_f in Equation (9) has a unique decomposition into $\mathcal{O}_k X_f$ and $\mathcal{T}_k^d U_f$ by the direct sum property. Consequently, \mathcal{T}_k^d can be determined by removing U_f from $\mathcal{T}_k^d U_f$.

Assume \mathcal{T}_k^d is determined. We introduce two Toeplitz matrices:

$$\Upsilon_U := \begin{bmatrix} 0 & h_{k-1} & \cdots & h_1 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & h_{k-1} \\ 0 & \cdots & \cdots & 0 \end{bmatrix}, \quad \Upsilon_L := \begin{bmatrix} h_k & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ h_{2k-1} & \cdots & \cdots & h_k \end{bmatrix}, \quad (12)$$

such that $\mathcal{O}_k \mathcal{C}_k^d = \Upsilon_U + \Upsilon_L$. It can be seen that Υ_U depends only on h_1, \dots, h_{k-1} whereas Υ_L depends only on h_k, \dots, h_{2k-1} . Substituting (10) into (9) gives

$$Y_f - \mathcal{T}_k^d U_f - \Upsilon_U U_p = \mathcal{O}_k A^k X_p + \Upsilon_L U_p, \quad (13)$$

where the terms in the left hand side are all known. Note that \mathcal{X}_p can be found from the equation $\mathcal{O}_k X_p = Y_p - \mathcal{T}_k^d U_p$. Furthermore, by (11) we have $\mathcal{X}_p + \mathcal{U}_p = \mathcal{X}_p \oplus \mathcal{U}_p$. As a result, we can uniquely decompose the left hand side of Equation (13) into $\mathcal{O}_k A^k X_p$ and $\Upsilon_L U_p$. In this way, Υ_L is determined.

Thus, the Markov parameters h_0, \dots, h_{2k-1} , or equivalently, $D, CB, \dots, CA^{2k-2}B$, are acquired. Finally, by Kalman's fundamental realization criterion [17], a unique realization (A, B, C, D) can be obtained.

The following summarises the algorithm for the deterministic case, at a conceptual level:

Algorithm 4.2. *Deterministic Subspace Algorithm via MP Determination*

1. Decompose Y_f into $\mathcal{O}_k X_f$ and $\mathcal{T}_k^d U_f$ using $\mathcal{X}_f \oplus \mathcal{U}_f \subset (\mathcal{Y}_p + \mathcal{U}_p) \oplus \mathcal{U}_f$. Determine \mathcal{T}_k^d .
2. Compute $\mathcal{X}_p = \text{span}\{Y_p - \mathcal{T}_k^d U_p\}$ and construct Υ_U .
3. Decompose $Y_f - \mathcal{T}_k^d U_f - \Upsilon_U U_p$ into $\mathcal{O}_k A^k X_p$ and $\Upsilon_L U_p$ using $\mathcal{X}_p + U_p = X_p \oplus U_p$. Then determine Υ_L .
4. Determine (A, B, C, D) from the $2k$ Markov parameters, for example using Kung's algorithm [18].

5 Deterministic-Stochastic Identification: A Three-Block Configuration

As shown in the previous section, a two-block configuration, splitting the data into ‘past’ (U_p) and ‘future’ (U_f) blocks, is adequate to identify Markov parameters in the deterministic case. However, when it comes to the combined deterministic-stochastic case, a two-block configuration cannot determine the Markov parameters of a system. In the remaining part of this paper, we demonstrate the use of a three-block configuration to estimate initial Markov parameters in a stochastic environment.

We split the data Hankel matrices into three-block configurations, such as:

$$\begin{bmatrix} U_p \\ \hline U_f \\ \hline U_r \end{bmatrix} := \begin{bmatrix} u(0) & u(1) & \cdots & u(q-1) \\ \vdots & \vdots & \ddots & \vdots \\ \hline u(k-1) & u(k) & \cdots & u(k+q-2) \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline u(2k-1) & u(2k) & \cdots & u(2k+q-2) \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline u(3k-1) & u(3k) & \cdots & u(3k+q-2) \end{bmatrix}. \quad (14)$$

The suffices p , f and r are supposed to be mnemonic, representing ‘past’, ‘future’ and ‘remote future’, respectively. We define Y_p , Y_f , and Y_r similarly, and the state sequences are defined as:

$$\begin{bmatrix} X_p \\ X_f \\ X_r \end{bmatrix} := \begin{bmatrix} x(0) & x(1) & \cdots & x(q-1) \\ \hline x(k) & x(k+1) & \cdots & x(k+q-1) \\ \hline x(2k) & x(2k+1) & \cdots & x(2k+q-1) \end{bmatrix}.$$

It can be seen without difficulty that the data equations can be written as:

$$Y_p = \mathcal{O}_k X_p + \mathcal{T}_k^d U_p + \mathcal{T}_k^w W_p + V_p, \quad (15)$$

$$Y_f = \mathcal{O}_k X_f + \mathcal{T}_k^d U_f + \mathcal{T}_k^w W_f + V_f, \quad (16)$$

$$Y_r = \mathcal{O}_k X_r + \mathcal{T}_k^d U_r + \mathcal{T}_k^w W_r + V_r. \quad (17)$$

In addition, the state equations linking the state sequences have the relationships:

$$X_f = A^k X_p + \mathcal{C}_k^d U_p + \mathcal{C}_k^w W_p, \quad (18)$$

$$X_r = A^k X_f + \mathcal{C}_k^d U_f + \mathcal{C}_k^w W_f. \quad (19)$$

As in Section 4, we divide $\mathcal{O}_k \mathcal{C}_k^d$ into two Toeplitz matrices, one in the upper triangular form Υ_U and one in the lower triangular form Υ_L , as in Equations (12). In short, we have $\mathcal{O}_k \mathcal{C}_k^d = \Upsilon_U + \Upsilon_L$.

Due to the presence of noise, a stronger condition than (11) is needed for experiments to be informative in the combined deterministic-stochastic case. One such condition can be expressed as follows:

$$\mathbf{E} \left[\begin{pmatrix} \Pi_{\mathcal{S}} X_f \\ U_f \\ U_r \end{pmatrix} \begin{pmatrix} \Pi_{\mathcal{S}} X_f \\ U_f \\ U_r \end{pmatrix}^* \right] > 0, \quad (20)$$

where $\mathcal{S} := \mathcal{Y}_p + \mathcal{U}_p + \mathcal{U}_f + \mathcal{U}_r$. We refer to [5, 8] for a detailed exposition on informative experiments in the combined deterministic-stochastic case. For the rest of this paper, we shall adopt this assumption.

Now, define two new variables Z_f and Z_r as follows:

$$\begin{bmatrix} Z_f \\ Z_r \end{bmatrix} := \begin{bmatrix} Y_f \\ Y_r \end{bmatrix} - \begin{bmatrix} \mathcal{T}_k^d & 0 \\ \Upsilon_U & \mathcal{T}_k^d \end{bmatrix} \begin{bmatrix} U_f \\ U_r \end{bmatrix}. \quad (21)$$

Using Equations (16), (17) and (19), it is easy to see that Z_f and Z_r can also be written as:

$$Z_f = \mathcal{O}_k X_f + \mathcal{T}_k^w W_f + V_f, \quad (22)$$

$$Z_r = \mathcal{O}_k A^k X_f + \Upsilon_L U_f + \mathcal{O}_k \mathcal{C}_k^w W_f + \mathcal{T}_k^w W_r + V_r. \quad (23)$$

The purpose of introducing Z_r and Z_f is to eliminate any known information relating to the Markov parameters h_0, \dots, h_{k-1} , so that the subsequent Markov parameters h_k, \dots, h_{2k-1} can be determined. The spaces spanned by the rows of Z_f and Z_r will be denoted by \mathcal{Z}_f and \mathcal{Z}_r , respectively.

In this three-block configuration, the projection space will be $\mathcal{S} := \mathcal{Y}_p + \mathcal{U}_p + \mathcal{U}_f + \mathcal{U}_r$. Again, k will be assumed to be greater than the observability index, which implies that \mathcal{O}_{k-1} is of full column rank.

Theorem 5.1. *Suppose (20) holds. Then,*

$$\begin{aligned}\Pi_{\mathcal{S}}\mathcal{X}_r &\subset \Pi_{\mathcal{S}}\mathcal{Y}_f + \mathcal{U}_f, \\ \Pi_{\mathcal{S}}\mathcal{X}_r + \mathcal{U}_r &= \Pi_{\mathcal{S}}\mathcal{X}_r \oplus \mathcal{U}_r \subset (\Pi_{\mathcal{S}}\mathcal{Y}_f + \mathcal{U}_f) \oplus \mathcal{U}_r,\end{aligned}$$

where $\mathcal{S} = \mathcal{Y}_p + \mathcal{U}_p + \mathcal{U}_f + \mathcal{U}_r$.

Proof. Due to the uncorrelated property of past, future and remote future noises, we have $\Pi_{\mathcal{S}}W_f = \Pi_{\mathcal{S}}V_f = \Pi_{\mathcal{S}}W_r = \Pi_{\mathcal{S}}V_r = 0$. Projecting Equation (16) onto \mathcal{S} gives $\Pi_{\mathcal{S}}Y_f = \mathcal{O}_k\Pi_{\mathcal{S}}X_f + \mathcal{T}_k^d U_f$. Thus, we have

$$\begin{aligned}\Pi_{\mathcal{S}}\mathcal{Y}_f &\subset \Pi_{\mathcal{S}}\mathcal{X}_f + \mathcal{U}_f, \\ \Pi_{\mathcal{S}}\mathcal{Y}_f + \mathcal{U}_f &\subset \Pi_{\mathcal{S}}\mathcal{X}_f + \mathcal{U}_f,\end{aligned}\tag{24}$$

where the last inclusion is obtained by adding \mathcal{U}_f to both sides of the first inclusion. On the other hand, it is also easy to see that $\mathcal{O}_k\Pi_{\mathcal{S}}X_f = \Pi_{\mathcal{S}}Y_f - \mathcal{T}_k^d U_f$. Thus, we have

$$\begin{aligned}\Pi_{\mathcal{S}}\mathcal{X}_f &\subset \Pi_{\mathcal{S}}\mathcal{Y}_f + \mathcal{U}_f, \\ \Pi_{\mathcal{S}}\mathcal{X}_f + \mathcal{U}_f &\subset \Pi_{\mathcal{S}}\mathcal{Y}_f + \mathcal{U}_f.\end{aligned}\tag{25}$$

where the last inclusion is obtained by adding \mathcal{U}_f to both sides of the first inclusion. In effect, (24) and (25) together imply

$$\Pi_{\mathcal{S}}\mathcal{X}_f + \mathcal{U}_f = \Pi_{\mathcal{S}}\mathcal{Y}_f + \mathcal{U}_f.$$

Using (20), it can be seen that $\Pi_{\mathcal{S}}\mathcal{X}_f + \mathcal{U}_f + \mathcal{U}_r = \Pi_{\mathcal{S}}\mathcal{X}_f \oplus \mathcal{U}_f \oplus \mathcal{U}_r$. Moreover, we have $\Pi_{\mathcal{S}}\mathcal{X}_r \subset \Pi_{\mathcal{S}}\mathcal{X}_f \oplus \mathcal{U}_f$, which can be seen from projecting Equation (19) onto \mathcal{S} . Thus,

$$\Pi_{\mathcal{S}}\mathcal{X}_r + \mathcal{U}_r = \Pi_{\mathcal{S}}\mathcal{X}_r \oplus \mathcal{U}_r \subset \Pi_{\mathcal{S}}\mathcal{X}_f \oplus \mathcal{U}_f \oplus \mathcal{U}_r = (\Pi_{\mathcal{S}}\mathcal{Y}_f + \mathcal{U}_f) \oplus \mathcal{U}_r,$$

which completes the proof. \square

Remark 5.1. From Theorem 5.1, it can be seen that $\Pi_{\mathcal{S}}\mathcal{Y}_f + \mathcal{U}_f$ provides a subspace which contains the space of the Kalman filter state sequence $\Pi_{\mathcal{S}}\mathcal{X}_r$. \triangle

Theorem 5.2. *Suppose Equation (20) holds. Then,*

$$\begin{aligned}\Pi_{\mathcal{S}}\mathcal{X}_f &= \Pi_{\mathcal{S}}\mathcal{Z}_f, \\ \Pi_{\mathcal{S}}\mathcal{X}_f + \mathcal{U}_f &= \Pi_{\mathcal{S}}\mathcal{X}_f \oplus \mathcal{U}_f = \Pi_{\mathcal{S}}\mathcal{Z}_f \oplus \mathcal{U}_f,\end{aligned}$$

where $\mathcal{S} = \mathcal{Y}_p + \mathcal{U}_p + \mathcal{U}_f + \mathcal{U}_r$.

Proof. Due to the uncorrelated property of past, future and remote future noises, we have $\Pi_{\mathcal{S}}W_f = \Pi_{\mathcal{S}}V_f = \Pi_{\mathcal{S}}W_r = \Pi_{\mathcal{S}}V_r = 0$. Thus, projecting Equation (22) onto \mathcal{S} gives

$$\Pi_{\mathcal{S}}Z_f = \mathcal{O}_k\Pi_{\mathcal{S}}X_f.$$

By Equation (20), we have $\Pi_{\mathcal{S}}\mathcal{X}_f + \mathcal{U}_f + \mathcal{U}_r = \Pi_{\mathcal{S}}\mathcal{X}_f \oplus \mathcal{U}_f \oplus \mathcal{U}_r$. Since \mathcal{O}_k is injective, the statement follows. \square

Remark 5.2. From Theorem 5.2, it can be seen that $\Pi_{\mathcal{S}}\mathcal{Z}_f$ can be used to observe the space of the Kalman filter state sequence $\Pi_{\mathcal{S}}\mathcal{X}_f$. \triangle

6 Markov Parameter Estimation

From Theorems 5.1 and 5.2, a corollary can immediately be obtained. In this corollary, we eliminate by orthogonal projections the state components $\Pi_{\mathcal{S}}X_r$ in Equation (30) and $\Pi_{\mathcal{S}}X_f$ in Equation (31) in order to isolate the input components. It turns out that such an isolation can be done without reducing the richness of the input sequences, as shown in (28), and (29) below.

Corollary 6.1. *Suppose Equation (20) holds. Denote $\mathcal{R} := \Pi_{\mathcal{S}}\mathcal{Y}_f + \mathcal{U}_f$ and $\mathcal{Q} := \Pi_{\mathcal{S}}\mathcal{Z}_f$. Then,*

$$\Pi_{\mathcal{R}^\perp} \Pi_{\mathcal{S}}Y_r = \mathcal{T}_k^d \Pi_{\mathcal{R}^\perp} U_r, \quad (26)$$

$$\Pi_{\mathcal{Q}^\perp} \Pi_{\mathcal{S}}Z_r = \Upsilon_L \Pi_{\mathcal{Q}^\perp} U_f, \quad (27)$$

where $\mathcal{S} = \mathcal{Y}_p + \mathcal{U}_p + \mathcal{U}_f + \mathcal{U}_r$. Furthermore,

$$\mathbf{E} [(\Pi_{\mathcal{R}^\perp} U_r) (\Pi_{\mathcal{R}^\perp} U_r)^*] > 0, \quad (28)$$

$$\mathbf{E} [(\Pi_{\mathcal{Q}^\perp} U_f) (\Pi_{\mathcal{Q}^\perp} U_f)^*] > 0. \quad (29)$$

Proof. Projecting Equations (17) and (23) onto \mathcal{S} gives

$$\Pi_{\mathcal{S}}Y_r = \mathcal{O}_k \Pi_{\mathcal{S}}X_r + \mathcal{T}_k^d U_r, \quad (30)$$

$$\Pi_{\mathcal{S}}Z_r = \mathcal{O}_k A^k \Pi_{\mathcal{S}}X_f + \Upsilon_L U_f. \quad (31)$$

From Theorems 5.1 and 5.2, we have $\Pi_{\mathcal{S}}\mathcal{X}_r \subset \mathcal{R}$ and $\Pi_{\mathcal{S}}\mathcal{X}_f \subset \mathcal{Q}$. Thus, we can isolate \mathcal{T}_k^d and Υ_L by the corresponding orthogonal projection. Furthermore, the direct sum property stated in Theorems 5.1 and 5.2 gives (28) and (29). \square

It can be seen that the matrices \mathcal{T}_k^d and Υ_L can directly be determined from Equations (26) and (27) in Corollary 6.1. However, recall that \mathcal{T}_k^d is a lower triangular Toeplitz matrix composed of h_0, \dots, h_{k-1} , and Υ_L a lower triangular Toeplitz matrix composed of h_k, \dots, h_{2k-1} . In practice such a \mathcal{T}_k^d and a Υ_L may not be a perfect lower triangular Toeplitz matrix. As a result, a direct extraction of the Markov parameters may not be possible since several values of a single Markov parameter may be obtained in the corresponding entries in the Toeplitz matrix. Owing to this consideration, another method of extracting the Markov parameters is proposed.

First, consider Equation (26). Partition the following matrices:

$$\Pi_{(\Pi_{\mathcal{S}}\mathcal{Y}_f + \mathcal{U}_f)^\perp} \Pi_{\mathcal{S}}Y_r =: \begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_k \end{bmatrix} \quad \text{and} \quad \Pi_{(\Pi_{\mathcal{S}}\mathcal{Y}_f + \mathcal{U}_f)^\perp} U_r =: \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_k \end{bmatrix},$$

with obvious dimensions. From Inequality (28), we have $\Pi_{(\Pi_S \mathcal{Y}_f + \mathcal{U}_f)^\perp} U_r$ of full row rank. Define S and P as:

$$S := [S_1 \quad S_2 \quad \cdots \quad S_k] \quad \text{and} \quad P := \begin{bmatrix} P_1 & P_2 & \cdots & P_k \\ & P_1 & \cdots & P_{k-1} \\ & & \ddots & \vdots \\ 0 & & & P_1 \end{bmatrix},$$

where P is of full row rank as well. Furthermore, it can be seen without difficulty that

$$S = HP,$$

where $H := [h_0 \quad h_1 \quad \cdots \quad h_{k-1}]$. Thus least-squares solutions for the first k Markov parameters can be obtained as

$$H = SP^*(PP^*)^{-1}.$$

To solve for the next k Markov parameters h_k, \dots, h_{2k-1} , we can simply repeat this procedure with Equation (27).

7 Kalman Filter State Estimation

Van Overschee and De Moor [25] showed that the state sequence X_k projected onto the input-output space \mathcal{S} was actually a state sequence estimate, running through the following Kalman filter with some initial condition \hat{X}_0 :

$$\hat{X}_t = A\hat{X}_{t-1} + BU_{t-1} + K_{t-1}(Y_{t-1} - C\hat{X}_{t-1} - DU_{t-1}), \quad (32a)$$

$$K_{t-1} = (G + A\tilde{P}_{t-1}C^*)(\Lambda_0 + C\tilde{P}_{t-1}C^*)^{-1}, \quad (32b)$$

$$\tilde{P}_t = A\tilde{P}_{t-1}A^* - (G + A\tilde{P}_{t-1}C^*)(\Lambda_0 + C\tilde{P}_{t-1}C^*)^{-1}(G + A\tilde{P}_{t-1}C^*)^*, \quad (32c)$$

where

$$\tilde{P}_0 + \Sigma^s := \frac{1}{q} \mathbf{E}([X_0 - \hat{X}_0][X_0 - \hat{X}_0]^*). \quad (33)$$

Note also that extension to the partial stochastic excitation case can be found in [5, 9]. In this section, a new method is derived for estimating the Kalman filter state sequences $\Pi_S X_k$ and $\Pi_{\mathcal{S}+\mathcal{Y}_k} X_{k+1}$. This method first determines the initial Markov parameters, then isolates the terms $\mathcal{O}_{2k-1} \Pi_S X_k$ and $\mathcal{O}_{2k-1} \Pi_{\mathcal{S}+\mathcal{Y}_k} X_{k+1}$ from the data equations, and finally factors out \mathcal{O}_{2k-1} using a singular value decomposition.

Suppose that the first $2k$ Markov parameters have already been determined — we have already seen how they can be obtained using a three-block partition of the block-Hankel matrices. Then, it is easy to see that

$$\begin{bmatrix} Y_f \\ Y_r \end{bmatrix} - \mathcal{T}_{2k}^d \begin{bmatrix} U_f \\ U_r \end{bmatrix} = \mathcal{O}_{2k} X_f + \mathcal{T}_{2k}^w \begin{bmatrix} W_f \\ W_r \end{bmatrix} + \begin{bmatrix} V_f \\ V_r \end{bmatrix}.$$

Projecting the above equation onto $\mathcal{S} = \mathcal{Y}_p + \mathcal{U}_p + \mathcal{U}_f + \mathcal{U}_r$ gives

$$\Pi_{\mathcal{S}} \begin{bmatrix} Y_f \\ Y_r \end{bmatrix} - \mathcal{T}_{2k}^d \begin{bmatrix} U_f \\ U_r \end{bmatrix} = \mathcal{O}_{2k} \Pi_{\mathcal{S}} X_k. \quad (34)$$

The main idea now is to extract the terms $\mathcal{O}_{2k-1} \Pi_{\mathcal{S}} X_k$ and $\mathcal{O}_{2k-1} \Pi_{\mathcal{S}} X_{k+1}$ from the data equations. To this end, we introduce two more matrices U_U and U_L , each of which has one block row (U_{3k-1} or U_k) deleted, in the following way:

$$\begin{bmatrix} U_f \\ U_r \end{bmatrix} = \begin{bmatrix} U_k \\ \vdots \\ U_{3k-1} \end{bmatrix} = \begin{bmatrix} U_U \\ \bar{U}_{3k-1} \end{bmatrix} = \begin{bmatrix} \bar{U}_k \\ U_L \end{bmatrix}. \quad (35)$$

We define Y_U and Y_L similarly. Thus, as in Equation (34) we have

$$\Pi_{\mathcal{S}} Y_U - \mathcal{T}_{2k-1}^d U_U = \mathcal{O}_{2k-1} \Pi_{\mathcal{S}} X_k, \quad (36)$$

$$\Pi_{\mathcal{S}+\mathcal{Y}_k} Y_L - \mathcal{T}_{2k-1}^d U_L = \mathcal{O}_{2k-1} \Pi_{\mathcal{S}+\mathcal{Y}_k} X_{k+1}. \quad (37)$$

Finally, we separate the observability matrix and the Kalman filter states using a singular value decomposition:

$$[\Pi_{\mathcal{S}} Y_U \quad \Pi_{\mathcal{S}+\mathcal{Y}_k} Y_L] - \mathcal{T}_{2k-1}^d [U_U \quad U_L] =: [\Gamma_1 \quad \Gamma_2] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Omega_1^* \\ \Omega_2^* \end{bmatrix}. \quad (38)$$

Then, in some particular state coordinates we have

$$\mathcal{O}_{2k-1} = \Gamma_1 \Sigma_1^{\frac{1}{2}}, \quad \text{and} \quad [\Pi_{\mathcal{S}} X_k \quad \Pi_{\mathcal{S}+\mathcal{Y}_k} X_{k+1}] = \Sigma_1^{\frac{1}{2}} \Omega_1^*.$$

An important property of the two Kalman filter state estimate sequences $\Pi_{\mathcal{S}} X_k$ and $\Pi_{\mathcal{S}+\mathcal{Y}_k} X_{k+1}$ is that they are sequences of two consecutive state estimates from the same Kalman filter bank. In [25] it has been shown that for the full stochastic excitation case (or Equation (5) positive definite for $t = \tau$), these two state estimates can be obtained from the same Kalman filter with initial state estimate $\Pi_{\mathcal{U}} X_0$, where $\mathcal{U} := \mathcal{U}_p + \mathcal{U}_f + \mathcal{U}_r$. It is also shown in [5, 8] that even if the system is not fully stochastically excited, the above statement remains true. In general, the initial conditions for $\Pi_{\mathcal{S}} X_k$ and $\Pi_{\mathcal{S}+\mathcal{Y}_k} X_{k+1}$ are respectively $\Pi_{(\mathcal{Y}_p^d \cap \mathcal{Y}_p) + \mathcal{U}} X_0$ and $\Pi_{(\mathcal{Y}_p^{d+} \cap \mathcal{Y}_p^+) + \mathcal{U}} X_0$, where $\mathcal{Y}_p^{d+} := \mathcal{Y}_p^d + \mathcal{Y}_k^d$ and $\mathcal{Y}_p^+ := \mathcal{Y}_p + \mathcal{Y}_k$. However, it is also shown that as long as k is greater than the observability index, we have

$$(\mathcal{Y}_p^d \cap \mathcal{Y}_p) + \mathcal{U} = (\mathcal{Y}_p^{d+} \cap \mathcal{Y}_p^+) + \mathcal{U}.$$

This is also to say that the two state estimates are from the same Kalman filter with the same initial condition.

8 Modeling of the Deterministic Part

In this section, three different approaches are presented for estimating the system matrices A , B , C , D . One method is to extract the system matrices directly from the initial Markov parameters in a similar fashion as in Kung's algorithm. The second method determines system matrices using the shift invariance approach. Finally, the third method, extended from [10], determines system matrices using the state sequence approach.

8.1 A Markov Parameter Approach

One effective way of finding system matrices from Markov parameters is to use Kung's algorithm. The procedure is the following. First determine the initial $2k$ Markov parameters as in Section 6. Then, construct the block-Hankel matrix H as:

$$H := \begin{bmatrix} h_1 & h_2 & \cdots & h_k \\ \cdot & \cdot & \cdot & \cdot \\ h_k & h_{k+1} & \cdots & h_{2k-1} \end{bmatrix}.$$

Decompose the block Hankel matrix using SVD, and follow the standard Kung's algorithm as in [18].

If the block-Hankel matrix is formed instead as

$$H := \begin{bmatrix} h_1 & \cdots & h_{2k-1} \\ \vdots & \cdot & 0 \\ h_{2k-1} & 0 & 0 \end{bmatrix}$$

then a balanced (and hence asymptotically stable) approximation can be obtained in a very simple fashion for rapidly decaying systems. By 'rapidly decaying' we mean that all eigenvalues of A are relatively close to zero, so that the first $2k$ Markov parameters contain all the significant dynamics of the system.

However, to determine the stochastic part of the system, Kalman filter state sequences are needed. Instead of calculating the SVD of (38), we compute the following SVD:

$$\begin{bmatrix} h_1 & \vdots & \vdots \\ \vdots & \Pi_S Y_U - \mathcal{T}_{2k-1}^d U_U & \vdots \\ h_{2k-1} & \vdots & \Pi_{S+\mathcal{Y}_k} Y_L - \mathcal{T}_{2k-1}^d U_L \end{bmatrix} =: [\Gamma_1 \quad \Gamma_2] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Omega_1^* \\ \Omega_2^* \end{bmatrix}.$$

It can be seen from (36) and (37) that this can also be written as:

$$\mathcal{O}_{2k-1} \begin{bmatrix} B & \vdots & \Pi_S X_k & \vdots & \Pi_{S+\mathcal{Y}_k} X_{k+1} \end{bmatrix} = \Gamma_1 \Sigma_1 \Omega_1^*.$$

In some particular state coordinates, we have

$$\mathcal{O}_{2k-1} = \Gamma_1 \Sigma_1^{\frac{1}{2}}, \quad \left[\begin{array}{c} B \\ \Pi_S X_k \\ \Pi_{S+\mathcal{Y}_k} X_{k+1} \end{array} \right] = \Sigma_1^{\frac{1}{2}} \Omega_1.$$

Clearly, B , $\Pi_S X_k$, and $\Pi_{S+\mathcal{Y}_k} X_{k+1}$ are obtained in this way, whereas D is simply equal to h_0 . A and C can be obtained from \mathcal{O}_k by using the shift invariance property, as shown in (39) in the next subsection.

8.2 A Shift Invariance Approach

The identification method using the shift invariance property is more complicated and requires a greater computational effort compared with the other two methods mentioned in this section. However, these shortcomings are usually considered to be outweighed by its ability to provide a better fit to the data.

In this method, we first determine \mathcal{O}_{2k-1} and $\Pi_S X_f$ as in Section 7. In practice, there may not exist matrices A and C which fit \mathcal{O}_{2k-1} perfectly, due to sampling error. A commonly used remedy is to find the best A and C in the least-squares sense with the use of the shift invariance property:

$$C = \gamma_1 \quad \text{and} \quad A = \begin{bmatrix} \gamma_1 \\ \vdots \\ \gamma_{2k-2} \end{bmatrix}^\dagger \begin{bmatrix} \gamma_2 \\ \vdots \\ \gamma_{2k-1} \end{bmatrix}, \quad \text{where} \quad \mathcal{O}_{2k-1} = \begin{bmatrix} \gamma_1 \\ \vdots \\ \gamma_{2k-1} \end{bmatrix}. \quad (39)$$

Reconstruct \mathcal{O}_{2k} from A and C . Now, from Equation (34) it is not hard to see

$$\Pi_S \begin{bmatrix} Y_f \\ Y_r \end{bmatrix} - \mathcal{O}_{2k} \Pi_S X_f = \mathcal{T}_{2k}^d \begin{bmatrix} U_f \\ U_r \end{bmatrix}.$$

Again, in practice there may not exist matrices B and D which fit the above equation perfectly. To solve for B and D from the above, we first note that

$$\begin{aligned} \mathcal{T}_{2k}^d \begin{bmatrix} U_f \\ U_r \end{bmatrix} &= \begin{bmatrix} I & & 0 \\ 0 & \ddots & \\ 0 & & \mathcal{O}_{2k-1} \end{bmatrix} \begin{bmatrix} D \\ B \end{bmatrix} U_k + \begin{bmatrix} 0 & & 0 \\ I & & \\ 0 & & \mathcal{O}_{2k-2} \end{bmatrix} \begin{bmatrix} D \\ B \end{bmatrix} U_{k+1} + \dots \\ &\quad + \begin{bmatrix} 0 & & 0 \\ 0 & \ddots & \\ I & & 0 \end{bmatrix} \begin{bmatrix} D \\ B \end{bmatrix} U_{3k-1}. \end{aligned}$$

Furthermore, note that B and D appear linearly in the equation. Using the Kronecker product

In [25], it is pointed out that $Y_k - C\hat{X}_k - DU_k$ is orthogonal to \hat{X}_k and U_k . More precisely, we have

$$\mathbf{E}[(Y_k - C\hat{X}_k - DU_k)(\hat{X}_k)^*] = \mathbf{E}[(Y_k - C\hat{X}_k - DU_k)(U_k)^*] = 0.$$

Consequently, unbiased estimates can be obtained from a least-squares solution:

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \mathbf{E} \left[\begin{pmatrix} \hat{X}_{k+1} \\ Y_k \end{pmatrix} \begin{pmatrix} \hat{X}_k \\ U_k \end{pmatrix}^* \right] \mathbf{E} \left[\begin{pmatrix} \hat{X}_k \\ U_k \end{pmatrix} \begin{pmatrix} \hat{X}_k \\ U_k \end{pmatrix}^* \right]^{-1}.$$

Note that the conditions for the above covariance matrix to be invertible are covered in [5, 8]. In practice, with real data, this unbiasedness is obtained asymptotically as $N \rightarrow \infty$, although k can remain fixed.

Again, for a discussion of stability enforcement with the state-sequence approach we refer to [6].

9 Modeling of the Stochastic Part

In this section, we will follow the method proposed in [24, 25] to model the stochastic subsystem. With A, B, C, D, \hat{X}_k and \hat{X}_{k+1} determined, this method uses (41) to estimate the steady state covariances Σ^w, Σ^{wv} and Σ^v . However, since Equations (41) are not in steady state, there is some approximation error. Nonetheless, as pointed out in [27] this approximation error will grow smaller, as the number of blocks k grows larger.

The stochastic identification procedure is summarized as follows. First determine A, B, C, D, \hat{X}_k and \hat{X}_{k+1} as in the previous sections. Now, let

$$\begin{bmatrix} \varepsilon_w \\ \varepsilon_v \end{bmatrix} = \begin{bmatrix} \hat{X}_{k+1} \\ Y_k \end{bmatrix} - \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \hat{X}_k \\ U_k \end{bmatrix}.$$

Then, obtain the approximation:

$$\begin{bmatrix} \Sigma^w & \Sigma^{wv} \\ \Sigma^{vw} & \Sigma^v \end{bmatrix} \approx \frac{1}{q} \mathbf{E} \left[\begin{pmatrix} \varepsilon_w \\ \varepsilon_v \end{pmatrix} \begin{pmatrix} \varepsilon_w \\ \varepsilon_v \end{pmatrix}^* \right]. \quad (42)$$

Solve Σ^s, G , and Λ_0 from:

$$\Sigma^s = A\Sigma^s A^* + \Sigma^w; \quad (43a)$$

$$G = A\Sigma^s C^* + \Sigma^{wv}; \quad (43b)$$

$$\Lambda_0 = C\Sigma^s C^* + \Sigma^v. \quad (43c)$$

Finally, the Kalman filter gain K and the associated covariance matrix P can be solved from:

$$K = (G + APC^*)(\Lambda_0 + CPC^*)^{-1}, \quad (44a)$$

$$P = APA^* - (G + APC^*)(\Lambda_0 + CPC^*)^{-1}(G + APC^*)^*. \quad (44b)$$

A final remark will be drawn about the positive realness of the resulting stochastic subsystem. From (42), it can be seen that covariance of the process and output noises is nonnegative definite. Thus, as shown in [13] positive realness of the resulting stochastic subsystem is guaranteed.

10 Numerically Efficient Implementation by QR Factorization

In practice, QR factorization has been shown to be a reliable and efficient way of implementing subspace algorithms¹. Both the MOESP algorithms in [28] and the N4SID algorithms in [25] use QR factorization to speed up numerical computation. To implement an efficient algorithm for the identification scheme developed in this paper, the following QR factorization is computed:

$$\begin{bmatrix} U_f \\ U_r \\ U_p \\ Y_p \\ Y_f \\ Y_r \end{bmatrix} =: \begin{bmatrix} R_{11} & & & & & & 0 \\ R_{21} & R_{22} & & & & & \\ R_{31} & R_{32} & R_{33} & & & & \\ R_{41} & R_{42} & R_{43} & R_{44} & & & \\ R_{51} & R_{52} & R_{53} & R_{54} & R_{55} & & \\ R_{61} & R_{62} & R_{63} & R_{64} & R_{65} & R_{66} & \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \\ Q_5 \\ Q_6 \end{bmatrix} =: RQ. \quad (45)$$

Note that the above arrangement of the data blocks shares a similar structure with that used in the MOESP PO scheme [28], both having the past data blocks placed in the center.

The major advantage of this factorization is that sequences projected onto $\mathcal{S} = \mathcal{Y}_p + \mathcal{U}_p + \mathcal{U}_f + \mathcal{U}_r$ are easily obtained by eliminating the Q_5 and Q_6 components. That is,

$$\Pi_{\mathcal{S}} \begin{bmatrix} Y_f \\ Y_r \end{bmatrix} = \begin{bmatrix} R_{51} & R_{52} & R_{53} & R_{54} & 0 & 0 \\ R_{61} & R_{62} & R_{63} & R_{64} & 0 & 0 \end{bmatrix} Q.$$

The following subsections show how this factorization facilitates the computation of Markov parameters and Kalman filter state sequences.

10.1 Determination of the First k Markov Parameters

From the QR factorization in Equation (45), a smaller QR factorization is computed:

$$\begin{bmatrix} R_{52} & R_{53} & R_{54} \\ R_{22} & 0 & 0 \\ R_{62} & R_{63} & R_{64} \end{bmatrix} =: \begin{bmatrix} \hat{R}_{52} & 0 & 0 \\ \hat{R}_{22} & \hat{R}_{23} & 0 \\ \hat{R}_{62} & \hat{R}_{63} & \hat{R}_{64} \end{bmatrix} \begin{bmatrix} \hat{Q}_2 \\ \hat{Q}_3 \\ \hat{Q}_4 \end{bmatrix}. \quad (46)$$

¹In the literature of subspace identification it is common to use the term RQ factorization, because the factorization used is one in which the triangular R factor comes before the orthogonal Q factor. This is sometimes called the LQ factorization. We prefer to use the term QR factorization, since this is the commonly used terminology in the linear algebra literature

It will be seen in Proposition 10.1 that \mathcal{T}_k^d can in fact be determined from this factorization. Also, with this smaller factorization we immediately obtain the following factorization:

$$\Pi_S \begin{bmatrix} U_f \\ Y_f \\ U_r \\ Y_r \end{bmatrix} = \begin{bmatrix} R_{11} & 0 & 0 & 0 \\ R_{51} & R_{52} & R_{53} & R_{54} \\ R_{21} & R_{22} & 0 & 0 \\ R_{61} & R_{62} & R_{63} & R_{64} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \end{bmatrix} = \begin{bmatrix} R_{11} & 0 & 0 & 0 \\ R_{51} & \hat{R}_{52} & 0 & 0 \\ R_{21} & \hat{R}_{22} & \hat{R}_{23} & 0 \\ R_{61} & \hat{R}_{62} & \hat{R}_{63} & \hat{R}_{64} \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & \hat{Q}_2 \\ 0 & \hat{Q}_3 \\ 0 & \hat{Q}_4 \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \end{bmatrix} =: \hat{R}\hat{Q}. \quad (47)$$

Proposition 10.1. *Suppose Equation (20) holds. Then, for \hat{R}_{23} and \hat{R}_{63} defined in Equation (46), we have*

$$\hat{R}_{63} = \mathcal{T}_k^d \hat{R}_{23}.$$

Proof. By Theorem 5.1 we have $\Pi_S \mathcal{Y}_r \subset \Pi_S \mathcal{Y}_f + \mathcal{U}_f + \mathcal{U}_r$. This implies $\hat{R}_{64} = 0$. Now, let $\mathcal{R} := \Pi_S \mathcal{Y}_f + \mathcal{U}_f$. From Equation (47), it is easy to see that

$$\begin{aligned} \Pi_{\mathcal{R}^\perp} U_r &= [0 \ 0 \ \hat{R}_{23} \ 0] \hat{Q}, \\ \Pi_{\mathcal{R}^\perp} \Pi_S Y_r &= [0 \ 0 \ \hat{R}_{63} \ 0] \hat{Q}. \end{aligned}$$

Thus, by Corollary 6.1 the result follows. \square

Using the QR factorization in (46) and Proposition 10.1, we can determine h_0, \dots, h_{k-1} by following the procedure described in Section 6.

Remark 10.1. It is noteworthy that the QR factorization in Equation (46) requires much less numerical calculation than that in Equation (45). \triangle

10.2 Determination of the Next k Markov Parameters

With Markov parameters h_0, \dots, h_{k-1} , we can construct Υ_U as in Equation (12). The next step is to determine $\Pi_S Z_f$ and $\Pi_S Z_r$ as in Equation (21). From Equation (47), we have

$$\begin{aligned} \Pi_S Z_f &= \Pi_S Y_f - \mathcal{T}_k^d U_f \\ &= [R_{51} \ \hat{R}_{52} \ 0 \ 0] \hat{Q} - \mathcal{T}_k^d [R_{11} \ 0 \ 0 \ 0] \hat{Q} \\ &=: [\bar{R}_{51} \ \hat{R}_{52} \ 0 \ 0] \hat{Q}; \end{aligned}$$

and

$$\begin{aligned} \Pi_S Z_r &= \Pi_S Y_r - [\Upsilon_U \ \mathcal{T}_k^d] \begin{bmatrix} U_f \\ U_r \end{bmatrix} \\ &= [R_{61} \ \hat{R}_{62} \ \hat{R}_{63} \ \hat{R}_{64}] \hat{Q} - [\Upsilon_U \ \mathcal{T}_k^d] \begin{bmatrix} R_{11} & 0 & 0 & 0 \\ R_{21} & \hat{R}_{22} & \hat{R}_{23} & 0 \end{bmatrix} \hat{Q} \\ &=: [\bar{R}_{61} \ \bar{R}_{62} \ \bar{R}_{63} \ \hat{R}_{64}] \hat{Q}. \end{aligned}$$

To obtain the next k Markov parameters, another small QR factorization, similar to that in Equation (46), is computed:

$$\begin{bmatrix} \bar{R}_{51} & \hat{R}_{52} \\ R_{11} & 0 \\ \bar{R}_{61} & \bar{R}_{62} \end{bmatrix} =: \begin{bmatrix} \tilde{R}_{51} & 0 \\ \tilde{R}_{11} & \tilde{R}_{12} \\ \tilde{R}_{61} & \tilde{R}_{62} \end{bmatrix} \begin{bmatrix} \tilde{Q}_1 \\ \tilde{Q}_2 \end{bmatrix}. \quad (48)$$

Note that Υ_L can be determined from the above QR factorization. To see how, we first observe the following factorization which can be obtained immediately from the above QR factorization:

$$\Pi_S \begin{bmatrix} Z_f \\ U_f \\ Z_r \end{bmatrix} = \begin{bmatrix} \bar{R}_{51} & \hat{R}_{52} & 0 & 0 \\ R_{11} & 0 & 0 & 0 \\ \bar{R}_{61} & \bar{R}_{62} & \bar{R}_{63} & \hat{R}_{64} \end{bmatrix} \hat{Q} = \begin{bmatrix} \tilde{R}_{51} & 0 & 0 & 0 \\ \tilde{R}_{11} & \tilde{R}_{12} & 0 & 0 \\ \tilde{R}_{61} & \tilde{R}_{62} & \bar{R}_{63} & \hat{R}_{64} \end{bmatrix} \begin{bmatrix} \tilde{Q}_1 & 0 \\ \tilde{Q}_2 & 0 \\ 0 & I \end{bmatrix} \hat{Q} =: \tilde{R}\tilde{Q}. \quad (49)$$

Proposition 10.2. *Suppose Equation (20) holds. Then, for \tilde{R}_{12} and \tilde{R}_{62} defined in Equation (48), we have*

$$\tilde{R}_{62} = \Upsilon_L \tilde{R}_{12}.$$

Proof. By Theorem 5.2 we have $\Pi_S Z_r \subset \Pi_S Z_f + \mathcal{U}_f$. This implies $\bar{R}_{63} = \hat{R}_{64} = 0$. Now, let $Q := \Pi_S Z_f$. From Equation (49), it is easy to see that

$$\begin{aligned} \Pi_{Q^\perp} U_f &= [0 \ \tilde{R}_{12} \ 0 \ 0] \tilde{Q}, \\ \Pi_{Q^\perp} \Pi_S Z_r &= [0 \ \tilde{R}_{62} \ 0 \ 0] \tilde{Q}. \end{aligned}$$

Thus, by Corollary 6.1 the result follows. \square

Using the QR factorization in (48) and Proposition 10.2, we can determine h_k, \dots, h_{2k-1} by following the procedure described in Section 6.

10.3 Determination of Kalman Filter State Sequences

The determination of Kalman filter state sequences is relatively straightforward. The first step is to extract U_U, U_L, Y_U and Y_L , which are defined in Equation (35). To obtain these block Hankel matrices, we can simply repartition the QR factorization in Equation (45) as:

$$\begin{bmatrix} U_U \\ U_{3k-1} \\ \bar{U}_p \\ Y_p \\ \bar{Y}_U \\ Y_{3k-1} \end{bmatrix} =: \begin{bmatrix} R_{11}^+ & & & & & & \\ R_{21}^- & R_{22}^- & & & & & \\ \bar{R}_{31} & \bar{R}_{32} & \bar{R}_{33} & & & & \\ R_{41} & R_{42} & R_{43} & R_{44} & & & \\ \bar{R}_{51}^+ & \bar{R}_{52}^+ & \bar{R}_{53}^+ & \bar{R}_{54}^+ & \bar{R}_{55}^+ & & \\ R_{61}^- & R_{62}^- & R_{63}^- & R_{64}^- & R_{65}^- & R_{66}^- & \end{bmatrix} Q := RQ, \quad (50)$$

and

$$\begin{bmatrix} U_k \\ U_L \\ \bar{U}_p \\ Y_p \\ \bar{Y}_k \\ Y_L \end{bmatrix} =: \begin{bmatrix} R_{11}^- \\ R_{21}^+ & R_{22}^+ \\ \bar{R}_{31}^- & \bar{R}_{32}^- & R_{33}^- \\ R_{41}^- & R_{42}^- & R_{43}^- & R_{44}^- \\ \bar{R}_{51}^- & \bar{R}_{52}^- & \bar{R}_{53}^- & \bar{R}_{54}^- & \bar{R}_{55}^- \\ R_{61}^+ & R_{62}^+ & R_{63}^+ & R_{64}^+ & R_{65}^+ & R_{66}^+ \end{bmatrix} Q := RQ. \quad (51)$$

Using the first $2k$ Markov parameters determined in the previous subsections, we reconstruct \mathcal{T}_{2k-1}^d . Consequently, Equations (36) and (37) can be rewritten as:

$$\begin{aligned} \mathcal{O}_{2k-1} \Pi_S X_k &= \Pi_S Y_U - \mathcal{T}_{2k-1}^d U_U \\ &= [R_{51}^+ \ R_{52}^+ \ R_{53}^+ \ R_{54}^+ \ 0 \ 0] Q - \mathcal{T}_{2k-1}^d [R_{11}^+ \ 0 \ 0 \ 0 \ 0 \ 0] Q \\ &=: [\bar{R}_{51}^+ \ R_{52}^+ \ R_{53}^+ \ R_{54}^+ \ 0 \ 0] Q, \end{aligned}$$

and

$$\begin{aligned} \mathcal{O}_{2k-1} \Pi_{S+\mathcal{Y}_k} X_{k+1} &= \Pi_{S+\mathcal{Y}_k} Y_L - \mathcal{T}_{2k-1}^d U_L \\ &= [R_{61}^+ \ R_{62}^+ \ R_{63}^+ \ R_{64}^+ \ R_{65}^+ \ 0] Q - \mathcal{T}_{2k-1}^d [R_{21}^+ \ R_{22}^+ \ 0 \ 0 \ 0 \ 0] Q \\ &=: [\bar{R}_{61}^+ \ \bar{R}_{62}^+ \ R_{63}^+ \ R_{64}^+ \ R_{65}^+ \ 0] Q. \end{aligned}$$

Finally, the following SVD is computed:

$$[\bar{R}_{51}^+ \ R_{52}^+ \ R_{53}^+ \ R_{54}^+ \ 0 \ 0 \ \bar{R}_{61}^+ \ \bar{R}_{62}^+ \ R_{63}^+ \ R_{64}^+ \ R_{65}^+ \ 0] =: [\Gamma_1 \ \Gamma_2] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Omega_1^* \\ \Omega_2^* \end{bmatrix}.$$

Then, we have

$$\mathcal{O}_{2k-1} = \Gamma_1 \Sigma_1^{\frac{1}{2}}, \quad \text{and} \quad [\Pi_S X_k \ \Pi_{S+\mathcal{Y}_k} X_{k+1}] = \Sigma_1^{\frac{1}{2}} \Omega_1^* Q.$$

Finally, schematic overviews of the algorithms developed in this paper can be found in Figures 1, 2, and 3.

11 Case Studies

Various aspects of the performance of the subspace algorithms presented in this paper are investigated in two case studies. These case studies are based on data taken from industrial systems. Both sets of data are available in the *DAISY* database [11], and both have also been investigated in [4].

- The first example, presented in Section 11.1, is based on step responses of a simulation model of a fractional distillation column.

- The second example, presented in Section 11.2, is based on a set of process measurements recorded from an industrial dryer.

\mathcal{H}_∞ and \mathcal{H}_2 model errors are not used commonly as performance indicators in the identification literature, since the true generating system is not usually available. But in the first example it is possible to obtain a very high-order model which fits the step response data exactly, and consider that model to be the ‘true system’. We are therefore able to investigate \mathcal{H}_∞ and \mathcal{H}_2 measures of performance. One purpose of such a study is to reveal how well the algorithms perform in extracting system dynamics within a certain model order. In the second example performance is measured in a more classical fashion, namely in terms of simulation and prediction accuracy.

As it is difficult to judge the performance of identification algorithms in an absolute sense, throughout the case studies a rather well-known subspace identification algorithm, N4SID Robust Combined Algorithm [27, p. 131], is used as a reference method, whose performance is provided for comparison with our algorithms. Note that this reference algorithm has been encoded as a MATLAB m-file `subid.m`, which can be found in [27].

11.1 A Distillation Column

The data is the step response of a very detailed nonlinear simulation model of a fractional distillation column with 3 input channels: the input cooling temperature, the reboiling temperature, and the pressure, and 2 output channels: the top product flow rate and the C4 concentration. Step responses were determined by applying a step to each of the input channels, while keeping the other two input channels at zero. The responses from the output channels were then sampled at a rate of once every second for a period of 2500 seconds, giving a total of 2501 samples.

The ‘True Model’

Since the dynamics of the column are very slow, we re-sampled the data with a sampling interval of 30 seconds, leaving a total of 84 samples. We then converted the step responses to impulse responses by taking successive differences, and used these to generate a high-order ‘true system’ that reproduced these impulse responses perfectly.

As pointed out in [3], it is important to make sure that the the input-output channels are scaled, so that they are all weighted comparably. A good basis for appropriate scaling is the energy content of each input-output channel, calculated as the sum of squares of all data points of the corresponding impulse response. Thus the input-output channels were rescaled according to the scheme developed in [3]. The final scaled step responses are depicted in Figure 4.

Identification Data

Order	Algorithm 10.3		Algorithm 10.4		Algorithm 10.5		N4SID Algorithm	
	\mathcal{H}_∞ Err	\mathcal{H}_2 Err	\mathcal{H}_∞ Err	\mathcal{H}_2 Err	\mathcal{H}_∞ Err	\mathcal{H}_2 Err	\mathcal{H}_∞ Err	\mathcal{H}_2 Err
4	39.2	16.96	26.4	13.84	21.8	13.65	35.0	13.86
6	7.9	3.44	8.1	3.27	7.9	3.25	12.3	3.54
8	7.8	2.61	7.3	2.61	7.6	2.62	11.1	2.98
10	8.2	2.25	9.8	2.34	8.5	2.26	14.7	2.80
12	6.6	1.84	6.3	1.83	6.8	1.86	7.8	1.95

Table 1: Accuracy of distillation column models in the noise free case.

Two sets of data, each of length $N = 1000$, were generated from the ‘true system’, with an initial condition vector generated as normal random variables and an input signal generated by uniform random variables distributed in $[0, 1]$. Furthermore, one set of simulation data was left noise-free, whereas the other was corrupted by white process and output noises of variance 0.15.

Performance Measurement

The proposed identification algorithms, Algorithms 10.3–10.5, were applied to these two sets of data. Note that the original model has a \mathcal{H}_∞ norm of 247.6 and a \mathcal{H}_2 norm of 75.63. The performance of the proposed algorithms and of the N4SID algorithm is given in Table 1 for the noise-free case and Table 2 for the noise-corrupted case. Note that algorithms which give the smallest \mathcal{H}_∞ or \mathcal{H}_2 error are highlighted for each model order.

Figures 5 and 6 show the frequency responses of the original model and the 8-th order identified models in the noise free and noise corrupted case respectively. For a clear view, only models produced by Algorithm 10.5 and N4SID algorithm are displayed for comparison. On the other hand, the errors of the identified models, measured as the maximum singular value of the difference from the original model, are plotted for each frequency in Figures 7 and 8. From these figures, it is easy to verify the \mathcal{H}_∞ errors calculated earlier. At this point, some observations can be drawn from the tables and figures as follows:

- The proposed algorithms seems to be less sensitive to noise and provide better models for most estimates of the state dimension n (the ‘order’).
- Algorithm 10.5 seems to be least sensitive to the choice of state dimension.

As regards computational complexity, Algorithms 10.3–10.5 needed about $135M$, $134M$, and $156M$ floating point operations respectively for the 8-th order models, while the N4SID algorithm required about $142M$ floating point operations. Finally, note that the truncation index k for

Order	Algorithm 10.3		Algorithm 10.4		Algorithm 10.5		N4SID Algorithm	
	\mathcal{H}_∞ Err	\mathcal{H}_2 Err	\mathcal{H}_∞ Err	\mathcal{H}_2 Err	\mathcal{H}_∞ Err	\mathcal{H}_2 Err	\mathcal{H}_∞ Err	\mathcal{H}_2 Err
4	42.6	16.80	32.3	14.14	28.7	13.92	36.1	14.13
6	20.1	5.05	42.7	7.03	26.8	5.86	23.6	5.31
8	17.2	4.15	14.7	4.31	14.6	4.29	31.2	5.38
10	13.9	4.08	13.6	3.96	14.7	3.98	83.7	11.18
12	21.1	4.49	22.9	4.42	17.7	3.96	86.1	9.79

Table 2: Accuracy of distillation column models in the noise corrupted case.

Algorithms 10.3–10.5 was chosen to be 15, whereas for the N4SID algorithm it was chosen to be 23. The rationale for choosing these values is explained below.

The results with these values of k can be seen in Tables 1 and 2. In each case, it can be observed that as we increase the model order, initially the model quality improves, as judged by both the \mathcal{H}_∞ norm and the \mathcal{H}_2 norm. Beyond some model order, however, the performance deteriorates. As is to be expected, this point is reached sooner for the noise-corrupted data set than for the noise-free data set.

Determination of the Truncation Index

A difficulty in using subspace based methods is choosing a suitable truncation index k . As the model order is determined from some matrix closely related to $\mathcal{O}_k X_f$, increasing k will provide more information about \mathcal{O}_k , or equivalently the dynamics of the pair (A, C) , which is certainly of importance when principal components are to be extracted from this pair. Nevertheless, an increase in k obviously decreases the number of columns q in X_f , creating two shortcomings. First of all, since X_f intrinsically contains information about the controllability pair (A, B) , decreasing q will reduce the information relating to the pair (A, B) , giving a similar problem mentioned above. On the other hand, these columns in X_f actually have an averaging effect on the data in each row, so they have the capability of smoothing the observations. Therefore, if q is small, there is relatively little smoothing. The effects of various k values on the accuracy of pole estimation have been made in [32, 15, 16], while some rigorous statistical treatments of this issue can be found in [12, 23].

When Algorithms 10.3–10.5 were applied to the two sets of simulation data, the value of k was chosen to give approximately optimal performance for both data sets, and a range of model orders. As remarked above, the best choice of k was markedly different for our 3-block algorithms than for the N4SID algorithm.

Order	Algorithm 10.3		Algorithm 10.4		Algorithm 10.5		N4SID Algorithm	
	Sim Err	Pre Err	Sim Err	Pre Err	Sim Err	Pre Err	Sim Err	Pre Err
2	35.26	22.18	39.30	19.77	31.90	16.55	32.72	15.19
4	31.69	10.36	32.40	9.90	30.66	9.12	33.33	9.26
6	33.96	9.22	32.21	8.86	30.72	8.18	31.17	8.11
8	48.09	8.58	31.10	8.03	29.72	7.60	31.83	7.68
10	36.77	8.00	33.62	7.62	30.05	7.73	—	9.22

Table 3: Simulation errors and prediction errors for the identification data set of an industrial dryer.

11.2 An Industrial Dryer

For the second example, data are taken from an industrial drying process with 3 input channels: fuel flow rate, hot gas exhaust fan speed, and rate of flow of raw material, and 3 output channels: dry bulb temperature, wet bulb temperature, and moisture content of the raw material leaving the dryer. The first two inputs could be manipulated, and were generated as pseudo-random binary signals. The third input is an external disturbance which could not be manipulated but could be measured. Values of the input and output signals were recorded at a sampling interval of 1 second to obtain 11665 samples.

Pretreatment of Data

The first 3000 data points were discarded to ensure that the process had reached a steady operating point. The means of the samples were subtracted from the data, then both input and output data were filtered through an anti-aliasing filter, before re-sampling at an interval of 10 seconds. As a result, a data set of 867 data points was obtained, which is depicted in Figure 9. As in [4], we divided the data set into two subsets, the first one containing the first 600 data points being used for identification, while the second one containing the rest being used for model validation.

Performance Measurement

Algorithms 10.3–10.5 were applied to the identification data set to obtain state-space models of various orders. A commonly used performance indicator, which measures the discrepancy between the original output and the model output as

$$\% \text{ Error} = \frac{1}{p} \sum_{i=1}^p \sqrt{\frac{\sum_{j=0}^{N-1} (y_i(j) - \hat{y}_i(j))^2}{\sum_{j=0}^{N-1} (y_i(j))^2}} \times 100,$$

was used to judge the fitness of each model produced by each algorithm in this example. Here,

$y_i(t)$ is the i -th channel of the original output, N is the number of the data points in the data set, and p is the number of output channels, which is 3 in this case. On the other hand, $\hat{y}_i(t)$ represents either the simulated i -th output generated by the deterministic part of the model, with which simulation errors are measured, or the one-step ahead predicted i -th output, with which prediction errors are measured. As the data set is divided into the identification and validation data sets, simulation errors and prediction errors can be evaluated for each of these sets, producing four performance indices. These indices are given in Table 3 for the identification data set and Table 4 for the validation data set. Note that an entry with “—” in the tables indicates that the identified model is unstable, thus giving a large error. Moreover, algorithms with best performance in each category are highlighted for easier comparison.

To illustrate the effect of Kalman filters, Figure 10 shows the simulation outputs (generated by the deterministic part of the models) and the prediction outputs (generated by the full models with Kalman filters) for the identification data set. From this figure, it is clear that models with Kalman filters can produce much better predictions. To avoid confusion, only models produced by Algorithm 10.5 and the N4SID algorithm are displayed for comparison. On the other hand, Figure 11 shows simulation outputs and cumulative simulation errors for the validation data set, where the cumulative error at time t is the error computed up to time t :

$$\sum_{j=0}^{t-1} (y_i(j) - \hat{y}_i(j))^2.$$

The reason for plotting these cumulative errors is that it is difficult to distinguish between the performance of both algorithms just by inspecting the simulation outputs. Here, it can be seen that the model produced by Algorithm 10.5 is slightly better than that produced by the N4SID algorithm. Finally, prediction outputs and cumulative prediction errors for the validation data set are depicted in Figure 12. This time, Algorithm 10.5 gives a better fit to the first two outputs, whereas the N4SID algorithm gives a better fit to the third output. At this point, we observe that:

- Algorithm 10.5 seems to produce the best model in terms of simulation and prediction errors, both in identification and validation, and more distinctly in terms of simulation errors for the identification data set.
- models obtained by Algorithms 10.3–10.5 improve, in terms of prediction errors, each time the model order increases up to an order of 10.
- Algorithm 10.3 does not seem to be as robust as the others, as it is not consistent in producing good estimates of various orders.

Algorithms 10.3–10.5 required about $12.3M$, $9.0M$, and $4.3M$ floating point operations respectively for the 6-th order models, while the N4SID algorithm used about $6.0M$ floating point operations for the same order. The number of floating point operations in fact depends heavily on the truncation index k used in the algorithms. For Algorithms 10.3–10.5, the truncation indices were set to 5, 4, and 3 respectively whereas for the N4SID algorithm it was set to 5.

Order	Algorithm 10.3		Algorithm 10.4		Algorithm 10.5		N4SID Algorithm	
	Sim Err	Pre Err	Sim Err	Pre Err	Sim Err	Pre Err	Sim Err	Pre Err
2	59.20	29.59	57.02	25.19	59.72	21.17	58.72	18.52
4	61.56	12.25	64.13	11.92	60.00	11.01	64.84	11.82
6	68.23	11.24	61.20	10.35	60.20	9.53	61.19	9.27
8	88.99	11.11	58.19	9.12	58.22	8.81	56.89	9.07
10	65.09	9.44	53.58	8.22	59.05	8.13	—	10.15

Table 4: Simulation errors and prediction errors for the validation data set of an industrial dryer.

12 Conclusions

In this paper, a new approach to subspace identification, using three-block partitions of data matrices, has been presented. Each of the three blocks can be seen as serving a different function — either an instrumental variable, a Kalman filter state observer, or a computational data set. Numerically efficient implementation using QR factorization has also been developed.

A significant advantage of the new approach is that it gives unbiased estimates of the system matrices with the state sequence approach in a straightforward way. It is also noteworthy that in estimating the observability matrix and the Kalman filter state sequence, all components relating to the input sequences are first removed, as in Equations (36) and (37). As a result, any negative effects due to ill conditioning of the input signals are minimized. One drawback of the proposed method is that it has a relatively complicated overall procedure, in which two additional QR factorizations (46) and (48) are required. Nonetheless, one can argue that these QR factorizations should not adversely affect the performance of the algorithms, as they are numerically robust and of relatively small size.

The proposed algorithms have been assessed by means of two case studies. These suggest that Algorithm 10.3 is less robust than the other two algorithms. The reason for this is that in determining the system matrices, this algorithm computes the matrices directly from the estimated Markov parameters without referring to the input-output data for a second time. Regarding Algorithm 10.4, it is demonstrated that this algorithm could identify the deterministic part of a system fairly well, while identification of the Kalman filter could be improved. The studies in this paper also show that the performance of Algorithm 10.5 is satisfactory, at a level comparable to that of the N4SID algorithm. Furthermore, in quite a number of cases the algorithm indeed produces better models under various measures, namely the \mathcal{H}_∞ error, the \mathcal{H}_2 error, the simulation error, and the prediction error.

13 Acknowledgement

The authors are indebted to K. Glover and B. De Moor for valuable discussions and suggestions. This work has been supported by the Natural Sciences and Engineering Research Council of Canada, the European Research Network in System Identification, the Committee of Vice-Chancellors and Principals of the Universities of the United Kingdom, Cambridge University Engineering Department, and Pembroke College, Cambridge.

References

- [1] H. Akaike. Stochastic theory of minimal realization. *IEEE Trans. on Automatic Control*, 19:667–674, 1974.
- [2] D. Bauer. *Some asymptotic theory for the estimation of linear systems using maximum likelihood methods or subspace algorithms*. PhD thesis, Technical University of Vienna, Austria, 1998.
- [3] C.T. Chou. *Geometry of Linear Systems and Identification*. PhD thesis, Cambridge University, U.K., 1994.
- [4] C.T. Chou and J.M. Maciejowski. System identification using balanced parameterizations. *IEEE Trans. on Automatic Control*, 42:956–974, 1997.
- [5] N.L.C. Chui. *Subspace Methods and Informative Experiments for System Identification*. PhD thesis, Cambridge University, U.K., 1997. Also available at URL:”<http://www-control.eng.cam.ac.uk/nlcc/report/thesis.ps>”.
- [6] N.L.C. Chui and J.M. Maciejowski. Realization of stable models with subspace methods. *Automatica*, 32:1587–1595, 1996.
- [7] N.L.C. Chui and J.M. Maciejowski. Uniqueness of minimal partial realizations and markov parameter identification. In *35th IEEE Conf. on Decision and Control*, pages 3642–3647, Kobe, 1996.
- [8] N.L.C. Chui and J.M. Maciejowski. Criteria for informative experiments with subspace identification. Technical Report CUED/F-INFENG/TR.338, Cambridge University, U.K., 1998. Submitted to SIAM J. Control and Optimization. Also available at URL:”<http://www-control.eng.cam.ac.uk/jmm/papers/TR338.ps>”.
- [9] N.L.C. Chui and J.M. Maciejowski. A geometric interpretation for subspace identification. Technical Report CUED/F-INFENG/TR.339, Cambridge University, U.K., 1998. Submitted to Syst. and Contr. Letters. Also available at URL:”<http://www-control.eng.cam.ac.uk/jmm/papers/TR339.ps>”.

- [10] N.L.C. Chui and J.M. Maciejowski. An unbiased subspace algorithm with the state sequence approach. In A. Beghi, L. Finesso, and G. Picci, editors, *Mathematical Theory on Networks and Systems*, Padova, Italy, 1999. Il Poligrafo. To appear.
- [11] B.L.R. De Moor. *DAISY: Database for the Identification of Systems*. Department of Electrical Engineering, ESAT/SISTA, K.U.Leuven, Belgium, URL: "http://www.esat.kuleuven.ac.be/sista/daisy/", 1997.
- [12] M. Deistler, K. Peternell, and W. Scherrer. Consistency and relative efficiency of subspace methods. *Automatica*, 31:1865–1875, 1995.
- [13] P.L. Faurre. Stochastic realization algorithms. In R. Mehra and D. Lainiotis, editors, *System Identification: Advances and Case Studies*. Academic Press, New York, 1976.
- [14] R.A. Horn and C.R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, Cambridge, U.K., 1994.
- [15] M. Jansson and B. Wahlberg. On weighting in state-space subspace system identification. In *3rd European Control Conf.*, volume 1, pages 435–440, Rome, Italy, 1995.
- [16] M. Jansson and B. Wahlberg. A linear regression approach to state-space subspace system identification. *Signal Processing*, 52:103–129, 1996.
- [17] R.E. Kalman. On minimal partial realizations of a linear input/output map. In R.E. Kalman and N. DeClaris, editors, *Aspects of Network and System Theory*. Holt, Reinhart and Winston, New York, 1971.
- [18] S.Y. Kung. A new low-order approximation algorithm via singular value decomposition. In *12th Asilomar Conf. on Circuits, Systems and Computers*, pages 705–714, Asilomar, 1978.
- [19] W.E. Larimore. Canonical variate analysis in identification, filtering, and adaptive control. In *29th IEEE Conf. on Decision and Control*, pages 596–604, Honolulu, Hawaii, 1990.
- [20] L. Ljung. *System Identification - Theory for the Users*. Prentice-Hall, Inc., N.J., U.S.A., 1987.
- [21] M. Moonen, B. De Moor, L. Vandenberghe, and J. Vandewalle. On- and off-line identification of linear state-space models. *Int. J. Control*, 49:219–232, 1989.
- [22] A. Nerode. Linear automaton transformations. *Proc. Amer. Math. Soc.*, 9:541–544, 1958.
- [23] K. Peternell, W. Scherrer, and M. Deistler. Statistical analysis of novel subspace identification methods. *Signal Processing*, 52:161–177, 1996.
- [24] P. Van Overschee and B. De Moor. Subspace algorithms for the stochastic identification problem. *Automatica*, 29:649–660, 1993.
- [25] P. Van Overschee and B. De Moor. N4SID: Subspace algorithms for the identification of combined deterministic stochastic systems. *Automatica*, 30:75–93, 1994.

- [26] P. Van Overschee and B. De Moor. A unifying theorem for three subspace system-identification algorithms. *Automatica*, 31:1853–1864, 1995.
- [27] P. Van Overschee and B. De Moor. *Subspace Identification for Linear Systems: Theory, Implementation, Applications*. Kluwer Academic Publishers, The Netherlands, 1996.
- [28] M. Verhaegen. Identification of the deterministic part of MIMO state space models given in innovations form from input-output data. *Automatica*, 30:61–74, 1994.
- [29] M. Verhaegen and P. Dewilde. Subspace model identification – Part 1. *Int. J. Control*, 56:1187–1210, 1992.
- [30] M. Viberg. Subspace-based methods for the identification of linear time-invariant systems. *Automatica*, 31:1835–1851, 1995.
- [31] M. Viberg, B. Wahlberg, and B. Ottersten. Analysis of state space system identification methods based on instrumental variables and subspace fitting. *Automatica*, 33:1603–1616, 1997.
- [32] B. Wahlberg and M. Jansson. 4SID linear regression. In *33rd IEEE Conf. on Decision and Control*, pages 2858–2863, Florida, 1994.

Algorithm 10.3. ID via MP Estimation — A Markov Parameter Appr.

1. Construct U_p, U_f, U_r and Y_p, Y_f, Y_r as in Equation (14).

2. Determine h_0, \dots, h_{k-1} from:

$$\Pi_{\mathcal{R}^\perp} \Pi_S Y_r = \mathcal{T}_k^d \Pi_{\mathcal{R}^\perp} U_r, \quad \text{for } \mathcal{R} := \Pi_S \mathcal{Y}_f + \mathcal{U}_f.$$

3. Compute Z_f and Z_r as in (21). Then determine h_k, \dots, h_{2k-1} from:

$$\Pi_{\mathcal{Q}^\perp} \Pi_S Z_r = \Upsilon_L \Pi_{\mathcal{Q}^\perp} U_f, \quad \text{for } \mathcal{Q} := \Pi_S \mathcal{Z}_f.$$

4. Extract U_U, U_L and Y_U and Y_L as in (35). Calculate the following SVD and partition accordingly by selecting an model order:

$$\begin{bmatrix} h_1 & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ h_{2k-1} & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \Pi_S Y_U - \mathcal{T}_{2k-1}^d U_U \\ \vdots \\ \Pi_{S+\mathcal{Y}_k} Y_L - \mathcal{T}_{2k-1}^d U_L \end{bmatrix} =: [\Gamma_1 \quad \Gamma_2] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Omega_1^* \\ \Omega_2^* \end{bmatrix}.$$

5. Assign $D = h_0$. Determine $\mathcal{O}_{2k-1}, B, \Pi_S X_k$ and $\Pi_{S+\mathcal{Y}_k} X_{k+1}$ as:

$$\mathcal{O}_{2k-1} = \Gamma_1 \Sigma_1^{\frac{1}{2}}, \quad \text{and} \quad \begin{bmatrix} B & \Pi_S X_k & \Pi_{S+\mathcal{Y}_k} X_{k+1} \end{bmatrix} = \Sigma_1^{\frac{1}{2}} \Omega_1^*.$$

6. Determine A and C from \mathcal{O}_{2k-1} using the shift invariance property, as in (39).

7. Determine Σ^w, Σ^v , and Σ^{wv} by taking the covariance of:

$$\begin{bmatrix} \varepsilon_w \\ \varepsilon_v \end{bmatrix} := \begin{bmatrix} \Pi_{S+\mathcal{Y}_k} X_{k+1} \\ Y_k \end{bmatrix} - \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \Pi_S X_k \\ U_k \end{bmatrix}.$$

8. Determine Kalman filter gain K and the error covariance matrix P using (43) and (44).

Figure 1: A schematic overview of the subspace identification via Markov parameter estimation — a Markov parameter approach.

Algorithm 10.4. ID via MP Estimation — A Shift Invariance Approach

1. Construct U_p, U_f, U_r and Y_p, Y_f, Y_r as in Equation (14).
2. Determine h_0, \dots, h_{k-1} from:

$$\Pi_{\mathcal{R}^\perp} \Pi_{\mathcal{S}} Y_r = \mathcal{T}_k^d \Pi_{\mathcal{R}^\perp} U_r, \quad \text{for } \mathcal{R} := \Pi_{\mathcal{S}} \mathcal{Y}_f + \mathcal{U}_f.$$

3. Compute Z_f and Z_r as in (21). Then determine h_k, \dots, h_{2k-1} from:

$$\Pi_{\mathcal{Q}^\perp} \Pi_{\mathcal{S}} Z_r = \Upsilon_L \Pi_{\mathcal{Q}^\perp} U_f, \quad \text{for } \mathcal{Q} := \Pi_{\mathcal{S}} Z_f.$$

4. Extract U_U, U_L and Y_U and Y_L as in (35). Calculate the following SVD and partition accordingly by selecting an model order:

$$[\Pi_{\mathcal{S}} Y_U \quad \Pi_{\mathcal{S}+\mathcal{Y}_k} Y_L] - \mathcal{T}_{2k-1}^d [U_U \quad U_L] =: [\Gamma_1 \quad \Gamma_2] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} \Omega_1^* \\ \Omega_2^* \end{bmatrix}.$$

5. Determine $\mathcal{O}_{2k-1}, \Pi_{\mathcal{S}} X_k$ and $\Pi_{\mathcal{S}+\mathcal{Y}_k} X_{k+1}$ as:

$$\mathcal{O}_{2k-1} = \Gamma_1 \Sigma_1^{\frac{1}{2}}, \quad \text{and} \quad [\Pi_{\mathcal{S}} X_k \quad \Pi_{\mathcal{S}+\mathcal{Y}_k} X_{k+1}] = \Sigma_1^{\frac{1}{2}} \Omega_1^*.$$

6. Determine A and C from \mathcal{O}_{2k-1} using the shift invariance property, as in (39).
7. Reconstruct \mathcal{O}_{2k} . Solve for B and D as in (40) from:

$$B, D = \arg \min_{B, D} \left\| \Pi_{\mathcal{S}} \begin{bmatrix} Y_f \\ Y_r \end{bmatrix} - \mathcal{O}_{2k} \Pi_{\mathcal{S}} X_k - \mathcal{T}_{2k}^d(B, D) \begin{bmatrix} U_f \\ U_r \end{bmatrix} \right\|_F.$$

8. Determine Σ^w, Σ^v , and Σ^{wv} by taking the covariance of:

$$\begin{bmatrix} \varepsilon_w \\ \varepsilon_v \end{bmatrix} := \begin{bmatrix} \Pi_{\mathcal{S}+\mathcal{Y}_k} X_{k+1} \\ Y_k \end{bmatrix} - \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \Pi_{\mathcal{S}} X_k \\ U_k \end{bmatrix}.$$

9. Determine Kalman filter gain K and the error covariance matrix P using (43) and (44).

Figure 2: A schematic overview of the subspace identification via Markov parameter estimation — a shift invariance approach.

Algorithm 10.5. ID via MP Estimation — A State Sequence Approach

1. Construct U_p, U_f, U_r and Y_p, Y_f, Y_r as in Equation (14).

2. Determine h_0, \dots, h_{k-1} from:

$$\Pi_{\mathcal{R}^\perp} \Pi_S Y_r = \mathcal{T}_k^d \Pi_{\mathcal{R}^\perp} U_r, \quad \text{for } \mathcal{R} := \Pi_S \mathcal{Y}_f + \mathcal{U}_f.$$

3. Compute Z_f and Z_r as in (21). Then determine h_k, \dots, h_{2k-1} from:

$$\Pi_{\mathcal{Q}^\perp} \Pi_S Z_r = \Upsilon_L \Pi_{\mathcal{Q}^\perp} U_f, \quad \text{for } \mathcal{Q} := \Pi_S Z_f.$$

4. Extract U_U, U_L and Y_U and Y_L as in (35). Calculate the following SVD and partition accordingly by selecting an model order:

$$[\Pi_S Y_U \quad \Pi_{\mathcal{S}+\mathcal{Y}_k} Y_L] - \mathcal{T}_{2k-1}^d [U_U \quad U_L] =: [\Gamma_1 \quad \Gamma_2] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} \Omega_1^* \\ \Omega_2^* \end{bmatrix}.$$

5. Determine $\Pi_S X_k$ and $\Pi_{\mathcal{S}+\mathcal{Y}_k} X_{k+1}$ as:

$$[\Pi_S X_k \quad \Pi_{\mathcal{S}+\mathcal{Y}_k} X_{k+1}] = \Sigma_1^{\frac{1}{2}} \Omega_1^*.$$

6. Determine A, B, C and D from:

$$A, B, C, D = \arg \min_{A, B, C, D} \left\| \begin{bmatrix} \Pi_{\mathcal{S}+\mathcal{Y}_k} X_{k+1} \\ Y_k \end{bmatrix} - \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \Pi_S X_k \\ U_k \end{bmatrix} \right\|_F.$$

7. Determine Σ^w, Σ^v , and Σ^{wv} by taking the covariance of:

$$\begin{bmatrix} \varepsilon_w \\ \varepsilon_v \end{bmatrix} := \begin{bmatrix} \Pi_{\mathcal{S}+\mathcal{Y}_k} X_{k+1} \\ Y_k \end{bmatrix} - \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \Pi_S X_k \\ U_k \end{bmatrix}.$$

8. Determine Kalman filter gain K and the error covariance matrix P using (43) and (44).

Figure 3: A schematic overview of the subspace identification via Markov parameter estimation — a state sequence approach.

Figure 4: Scaled step responses of the distillation column simulation model. Unit time interval = 30 sec.

Figure 5: Comparison of frequency responses of alternative 8-state models in the noise free case: original model (solid); model from Algorithm 10.5 (dashdot); model from N4SID algorithm (dotted).

Figure 6: Comparison of frequency responses of alternative 8-state models in the noise corrupted case: original model (solid); model from Algorithm 10.5 (dashdot); model from N4SID algorithm (dotted).

Figure 7: The maximum singular value of the difference from the original model in the noise free case: model from Algorithm 10.5 (dashdot); model from N4SID algorithm (dotted).

Figure 8: The maximum singular value of the difference from the original model in the noise corrupted case: model from Algorithm 10.5 (dashdot); model from N4SID algorithm (dotted).

Figure 9: Input and output signals of an industrial dryer. Unit time interval = 10 sec.

Figure 10: Comparison of alternative 6-state models for identification data: original model (solid); model from Algorithm 10.5 (dashdot); model from N4SID algorithm (dotted). Unit time interval = 10 sec.

Figure 11: Comparison of simulation outputs of alternative 6-state models for validation data: original model (solid); model from Algorithm 10.5 (dashdot); model from N4SID algorithm (dotted). Unit time interval = 10 sec.

Figure 12: Comparison of prediction outputs of alternative 6-state models for validation data: original model (solid); model from Algorithm 10.5 (dashdot); model from N4SID algorithm (dotted). Unit time interval = 10 sec.