Chapter 10

Subspace Identification

> "Given observations of $m \geq 1$ input signals, and $p \geq 1$ signals resulting from those when fed into a dynamical system under study, can we estimate the internal dynamics regulating that system?"

Subspace techniques encode the notion of the state as 'bottleneck between past and future' using a series of geometric operations on input-output sequences. They should be contrasted to the ideas governing PEM approaches as described in Chapter 5.

Subspace algorithms make extensive use of the observability and controllability matrices and of their structure.

10.1 Deterministic Subspace Identification

The class of subspace algorithms found their root in the methods for converting known sequences of impulse response matrices (IRs), Transfer Function matrices (TFs) or covariance matrices into a corresponding state-space system. Since those state-spaces satisfy the characteristics as encoded in IRs, TFs or covariance matrices, they are referred to as 'realizations' (not to be confused with

Given: An input signal $\{\mathbf{u}_t\}_{t=1}^n \subset \mathbb{R}^m$ and output signal $\{\mathbf{y}_t\}_{t=1}^n \subset \mathbb{R}^p$, both of length n, and satisfying an (unkown) deterministic state-space of order d, or

$$\begin{cases} \mathbf{x}_{t+1} & = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t \\ \mathbf{y} & = \mathbf{C}\mathbf{x}_t + \mathbf{D}\mathbf{u}_t \end{cases} \tag{10.1}$$

where $t = 1, \ldots, n$ and $\mathbf{x}_0$ is fixed.

Problem: Recover

(a) The order d of the unknown system.

(b) The unknown system matrices $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$

Figure 10.1: The problem a deterministic Subspace Identification algorithms aims to solve.

'realizations' of random variables as described in Chapter 5). The naming convention originates from times where people sought quite intensively to physical 'realizations' of mathematically described electrical systems.

10.1.1 Deterministic Realization: IR2SS

Given a sequence of IR matrices $\{\mathbf{G}_\tau\}_{\tau\geq 0}$ with $\sum_{\tau\geq 0}\|\mathbf{G}\|_2 < \infty$, consider the problem of finding a realization $\mathcal{S} = (\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$. Since we know that $\mathbf{G}_0 = \mathbf{D}$, we make life easier by focussing on the problem of recovering the matrices $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ from given sequence of IR matrices $\{\mathbf{G}_\tau\}_{\tau\geq 1}$.

Let us start by defining the *Block-Hankel* matrix $\mathbf{H}_{d'} \in \mathbb{R}^{d'p\times d'm}$ of degree d' which will play a central role in the derivation:

$$\mathbf{H}_{d'} = \begin{bmatrix} \mathbf{G}_1 & \mathbf{G}_2 & \mathbf{G}_3 & \dots & \mathbf{G}_{d'} \\ \mathbf{G}_2 & \mathbf{G}_3 & & & \mathbf{G}_{d'+1} \\ \mathbf{G}_3 & & \ddots & & \mathbf{G}_{d'+2} \\ \vdots & & & & \vdots \\ \mathbf{G}_{d'} & & & & \mathbf{G}_{2d'-1} \end{bmatrix} \in \mathbb{R}^{d'p\times d'm}. \tag{10.2}$$

Then the input-signal $\mathbf{y}$ and its corresponding output-signal $\mathbf{u}$ are related as follows. Let $\mathbf{u}_t^- = (\mathbf{u}_t, \mathbf{u}_{t-1}, \mathbf{u}_{t-2}, \dots)^T \in \mathbb{R}^\infty$ be the reverse of the input-signal, and let $\mathbf{y}_t^+ = (\mathbf{y}_t, \mathbf{y}_{t+1}, \mathbf{y}_{t+2}, \dots)^T \in \mathbb{R}^\infty$ be the forward output signal taking values from instant t on. Then we can write

$$\mathbf{y}_t^+ = \begin{bmatrix} \mathbf{y}_t \\ \mathbf{y}_{t+1} \\ \mathbf{y}_{t+2} \\ \vdots \end{bmatrix} = \mathbf{H}_\infty \mathbf{u}_t^- = \begin{bmatrix} \mathbf{G}_1 & \mathbf{G}_2 & \mathbf{G}_3 & \dots \\ \mathbf{G}_2 & \mathbf{G}_3 & & \\ \mathbf{G}_3 & & & \\ \vdots & & & \ddots \end{bmatrix} \begin{bmatrix} \mathbf{u}_t \\ \mathbf{u}_{t-1} \\ \mathbf{u}_{t-2} \\ \vdots \end{bmatrix} \tag{10.3}$$

where the limit is taken for $d' \to \infty$ (represented by the open ended '...' in the matrix). Hence the interpretation of the matrix $\mathbf{H}_\infty$ is that 'when presented the system with an input which drops to 0 when going beyond t, then the block-Hankel matrix $\mathbf{H}_\infty$ computes the output of the system after t'. That is, it gives the system output when the system is in free mode.

Lemma 10 (Factorization) *Given a State-space system $\mathcal{S} = (\mathbf{A}, \mathbf{B}, \mathbf{C})$ (with $\mathbf{D} = 0$), then*

$$\mathbf{H}_d = \mathcal{O}_d \mathbf{T}\mathbf{T}^{-1}\mathcal{C}_d, \tag{10.4}$$

for any nonsingular transformation $\mathbf{T} \in \mathbb{R}^{d\times d}$. Moreover, we have the following rank conditions

$$rank(\mathbf{H}_d) \leq \max(rank(\mathcal{O}_d), rank(\mathcal{C}_d)) \leq d. \tag{10.5}$$

Lemma 11 (Minimal Realization) *Given a State-space system $\mathcal{S} = (\mathbf{A}, \mathbf{B}, \mathbf{C})$ (with $\mathbf{D} = 0$) with minimal state-dimension d, then we have for all $d' \geq d$ that*

$$rank(\mathbf{H}_{d'}) = d. \tag{10.6}$$

This result is directly seen by construction. However the implications are important: it says that the rank of the block-Hankel matrix is equal to the minimal dimension of the states in a state-space model which can exhibit the given behavior. That means that in using this factorization, we are not only given the system-matrices, but as well the minimal dimension. Recall that the problem of order estimation in PEM methods is often considered as a separate model selection problem, and required in a senses the use of ad-hoc tools.

The idea of the Ho-Kalman realization algorithm is then that in case the block-Hankel matrix $\mathbf{H}_{d'}$ can be formed for large enough d', the Singular Value Decomposition of this matrix let us recover the observability matrix $\mathcal{O}_{d'}$ and the controllability matrix $\mathcal{C}_{d'}$ (up to a transformation $\mathbf{T}$). These can in turn be used to extract the system matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ (up to a transformation $\mathbf{T}$). In order to perform this last step, we need the following straightforward idea:

Proposition 2 (Recovering System Matrices) *The observability matrix $\mathcal{O}_d$ satisfies the following recursive relations*

$$\mathcal{O}_d = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}_{d-1} \end{bmatrix} = \begin{bmatrix} \mathbf{C} \\ \mathcal{O}_{d-1}\mathbf{A} \end{bmatrix}. \tag{10.7}$$

The controlability matrix $\mathcal{C}_d$ satisfies the following recursive relations

$$\mathcal{C}_d = \begin{bmatrix} \mathbf{B} \\ \mathbf{AB} \\ \mathbf{A}^2\mathbf{B} \\ \vdots \\ \mathbf{A}_{d-1}\mathbf{B} \end{bmatrix} = \begin{bmatrix} \mathbf{B} \\ \mathbf{A}\mathcal{C}_{d-1} \end{bmatrix}. \tag{10.8}$$

That means that once the matrix $\mathcal{C}_d$ and $\mathcal{O}_d$ are known, the system matrices $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ can be recovered straightforwardly by selecting appropriate parts. Optimality of the SVD then ensures that we will recover a minimal state space 'realizing' the sequence of IR matrices $\{\mathbf{G}_\tau\}_{\tau>0}$. In full, the algorithm becomes as follows.

Algorithm 1 (Ho-Kalman) *Given $d' \geq d$ and the IRs $\{\mathbf{G}_\tau\}_{\tau=0}^{d'}$, find a realization $\mathcal{S} = (\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ up to within a similarity transform $\mathbf{T}$.*

1. *Set $\mathbf{D} = \mathbf{G}_0$.*

2. *Decompose*

3. *Find $\mathbf{A}$*

4. *Find $\mathbf{B}$ and $\mathbf{C}$*

10.1.2 N4SID

Now we turn to the question how one may use the ideas of the Kalman-Ho algorithm in order to find a realization directly from input-output data, rather than from given IRs. Again, this can be done by performing a sequence of projections under the assumption that the given data obeys a state-space model with unknown (but fixed) system matrices $\mathcal{S} = (\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$. This technique is quite different from the PEM approach were the estimation problem is turned into an optimization problem. To make the difference explicit, we refer to the projection-based algorithms as *Subspace Identification (SID) algorithms*. The first studied SID algorithm was (arguably) N4SID (niftily pronounced as a californian plate as '*enforce it*'). The abbreviation stands for 'Numerical algorithm For Subspace IDentification'.

The central insight is the following expression of the future output of the system in terms of (i) the unknown states at that time, and (ii) the future input signals. Formally, we define the matrices representing *'past'* as

$$\mathbf{U}_p = \begin{bmatrix} \mathbf{u}_1 & \dots & \mathbf{u}_{d'} \\ \mathbf{u}_2 & & \mathbf{u}_{d'+1} \\ \vdots & & \vdots \\ \mathbf{u}_{t+1} & & \mathbf{u}_{t+d'} \\ \vdots & & \vdots \\ \mathbf{u}_{n-2d'+1} & \dots & \mathbf{u}_{n-d'} \end{bmatrix}, \ \mathbf{Y}_p = \begin{bmatrix} \mathbf{y}_1 & \dots & \mathbf{y}_{d'} \\ \mathbf{y}_2 & & \mathbf{y}_{d'+1} \\ \vdots & & \vdots \\ \mathbf{y}_{t+1} & & \mathbf{y}_{t+d'} \\ \vdots & & \vdots \\ \mathbf{y}_{n-2d'+1} & \dots & \mathbf{y}_{n-d'} \end{bmatrix}, \tag{10.9}$$

and equivalently we define the matrices representing *'future'* as

$$\mathbf{U}_f = \begin{bmatrix} \mathbf{u}_{d'+1} & \dots & \mathbf{u}_{2d'} \\ \mathbf{u}_{d'+2} & & \mathbf{u}_{2d'+1} \\ \vdots & & \vdots \\ \mathbf{u}_{t+1} & & \mathbf{u}_{t+d'} \\ \vdots & & \vdots \\ \mathbf{u}_{n-d'+1} & \dots & \mathbf{u}_{n} \end{bmatrix}, \ \mathbf{Y}_f = \begin{bmatrix} \mathbf{y}_{d'+1} & \dots & \mathbf{y}_{2d'} \\ \mathbf{y}_{d'+2} & & \mathbf{y}_{2d'+1} \\ \vdots & & \vdots \\ \mathbf{y}_{t+1} & & \mathbf{y}_{t+d'} \\ \vdots & & \vdots \\ \mathbf{y}_{n-d'+1} & \dots & \mathbf{y}_{n} \end{bmatrix}. \tag{10.10}$$

Now we are all set to give the main factorization from which the N4SID algorithm will follow.

Lemma 12 (N4SID) *Assuming that a system* $\mathcal{S} = (\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ *underlying the signals exists, than there exist matrices* $\mathbf{F} \in \mathbb{R}^{md' \times pd'}$ *and* $\mathbf{F} \in \mathbb{R}^{md' \times pd'}$ *such that*

$$\mathbf{Y}_f = \mathbf{X}_f \mathcal{O}^{d'T} + \mathbf{U}_f \mathbf{G}^{d'T} \tag{10.11}$$

$$= \mathbf{U}_p \mathbf{F} + \mathbf{Y}_p \mathbf{F}' + \mathbf{U}_f \mathbf{G}^{d'T}. \tag{10.12}$$

This really follows from working out the terms, schematically (again empty entries denote blocks of zeros):

$$
\mathbf{Y}_f = \begin{bmatrix} \mathbf{y}_{d'+1} & \dots & \mathbf{y}_{2d'} \\ \mathbf{y}_{d'+2} & & \mathbf{y}_{2d'+1} \\ \vdots & & \vdots \\ \mathbf{y}_{t+1} & & \mathbf{y}_{t+d'} \\ \vdots & & \vdots \\ \mathbf{y}_{n-d'+1} & \dots & \mathbf{y}_n \end{bmatrix}
= \begin{bmatrix} \mathbf{x}_{d'+1} \\ \mathbf{x}_{d'+2} \\ \vdots \\ \mathbf{x}_{t+1} \\ \vdots \\ \mathbf{x}_{n-d'+1} \end{bmatrix} \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \dots \\ \mathbf{CA}^{d'-1} \end{bmatrix}^T + \begin{bmatrix} \mathbf{u}_{d'+1} & \dots & \mathbf{u}_{2d'} \\ \mathbf{u}_{d'+2} & & \mathbf{u}_{2d'+1} \\ \vdots & & \vdots \\ \mathbf{u}_{t+1} & & \mathbf{u}_{t+d'} \\ \vdots & & \vdots \\ \mathbf{u}_{n-d'+1} & \dots & \mathbf{u}_n \end{bmatrix} \begin{bmatrix} \mathbf{G}_0 & \mathbf{G}_1 & \dots & \mathbf{G}_{d'-1} \\ & \mathbf{G}_0 & & \\ & & \ddots & \vdots \\ & & & \mathbf{G}_0 \end{bmatrix}
= \mathbf{X}_f^{d'} \mathcal{O}^{d'^T} + \mathbf{U}_f \mathbf{G}^{d'^T}. \quad (10.13)
$$

We see that the sequence of states $\{\mathbf{x}\}_{t \geq d'}$ can be written as a (finite) linear combination of the matrices $\mathbf{U}_p$ and $\mathbf{Y}_f$. Hereto, we interprete a similar linear relation

$$
\mathbf{Y}_p = \begin{bmatrix} \mathbf{y}_1 & \dots & \mathbf{y}_{d'} \\ \mathbf{y}_2 & & \mathbf{y}_{d'+1} \\ \vdots & & \vdots \\ \mathbf{y}_{t+1} & & \mathbf{y}_{t+d'} \\ \vdots & & \vdots \\ \mathbf{y}_{n-2d'+1} & \dots & \mathbf{y}_{n-d'} \end{bmatrix}
= \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_{t+1} \\ \vdots \\ \mathbf{x}_{n-2d'+1} \end{bmatrix} \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \dots \\ \mathbf{CA}^{d'-1} \end{bmatrix}^T + \begin{bmatrix} \mathbf{u}_1 & \dots & \mathbf{u}_{d'} \\ \mathbf{u}_2 & & \mathbf{u}_{d'+1} \\ \vdots & & \vdots \\ \mathbf{u}_{t+1} & & \mathbf{u}_{t+d'} \\ \vdots & & \vdots \\ \mathbf{u}_{n-2d'+1} & \dots & \mathbf{u}_{n-d'} \end{bmatrix} \begin{bmatrix} \mathbf{G}_0 & \mathbf{G}_1 & \dots & \mathbf{G}_{d'-1} \\ & \mathbf{G}_0 & & \\ & & \ddots & \vdots \\ & & & \mathbf{G}_0 \end{bmatrix}
= \mathbf{X}_p^{d'} \mathcal{O}^{d'^T} + \mathbf{U}_p \mathbf{G}^{d'^T}. \quad (10.14)
$$

Note that this step needs some care to let the indices match properly, i.e. we can only use the signals between iteration $d', \dots, n - d'$. As such the factorization follows readily.

This factorization gives us an equation connecting the 'past' and 'future'. Moreover, note that eq. (10.12) looks like a model which is linear in the unknown $\mathbf{F}, \mathbf{F}'$ and $\mathbf{G}^{d'}$, and we know by now how to solve this one since we can construct the matrices $\mathbf{Y}_f$ and $\mathbf{U}_p, \mathbf{Y}_p, \mathbf{U}_f$. Indeed, using an OLS estimator - or equivalent the orthogonal projection - lets us recover the unknown matrices, that is, under appropriate rank conditions in order to guarantee that only a single solution exists. We proceed however by defining a shortcut to solving this problem in order to arrive at less expensive implementations.

What is left to us is to decompose the term $\mathbf{O} = \mathbf{X}_f^{d'} \mathcal{O}^{d'^T}$ in the state variables and the observability matrix. This however, we learned to do from Kalman-Ho's algorithm (previous Subsection). Indeed, an SVD decomposition of the matrix $\mathbf{O}$ gives us the rank d of the observability matrix, together with a reduced rank decomposition from which $\{\mathbf{x}_t\}$ and $\mathcal{O}^d$ can be recovered (up to a similarity transform $\mathbf{T}$). Now it is not too difficult to retrieve the system matrices $\mathcal{S} = (\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$. A more robust way goes as follows: given the input and output signals, as well as the recovered states, we know that those satisfy the state-space system and as such

$$\mathbf{Z}_f \triangleq \begin{bmatrix} \mathbf{x}_2 & \mathbf{y}_1 & \\ \mathbf{x}_3 & \mathbf{y}_2 & \\ \vdots & \vdots & \\ \mathbf{x}_{t+1} & & \mathbf{y}_t \\ \vdots & \vdots & \\ \mathbf{x}_n & \mathbf{y}_{n-1} & \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{u}_1 \\ \mathbf{x}_2 & \mathbf{u}_2 \\ \vdots & \vdots \\ \mathbf{x}_t & \mathbf{u}_t \\ \vdots & \vdots \\ \mathbf{x}_{n-1} & \mathbf{u}_{n-1} \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{B} & \mathbf{C} \end{bmatrix} \triangleq \mathbf{Z}_p \begin{bmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{B} & \mathbf{C} \end{bmatrix}, \tag{10.15}$$

from which one can recover directly the matrices $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$, for instance by solving a LS problem or an Orthogonal projection as $\mathbf{Z}_p^\dagger \mathbf{Z}_f$. The N4SID algorithm is summarized as follows:

Algorithm 2 (N4SID) *Given $d' \geq d$ and the signals $\mathbf{u} \in \mathbb{R}^{pn}$ and $\mathbf{y} \in \mathbb{R}^{mn}$, find a realization $\mathcal{S} = (\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ up to within a similarity transform $\mathbf{T}$.*

1. *Find $\mathbf{O}_d$.*

2. *Find the state sequence $\mathbf{X}$*

3. *Find $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$*

10.1.3 Variations on the Theme

Since its inception, almost decades of research yielded many important improvements of the algorithm. Here we will review some of those, starting with a description of a family of related methods going under the name of MOESP subspace identification algorithms.

MOESP

Intersection

Projection

10.2 Stochastic Subspace Techniques

10.2.1 Stochastic Realization: Cov2SS

Given a sequence of covariance matrices $\{\Lambda_\tau = \mathbb{E}[Y_t Y_{t+\tau}]\}_{\tau \geq 0}$ with $\sum_{\tau \geq 0} \|\lambda_\tau\|_2 < \infty$, can we find a stochastic MIMO system $\mathcal{S} = (\mathbf{A}, \mathbf{C})$ realizing those covariance matrices? More specifically, if the realized system $\mathcal{S} = (\mathbf{A}, \mathbf{C})$ were driven by zero-mean colored Gaussian noise $\{W_t\}_t$ and $\{V_t\}_t$,

Given: An input signal $\{\mathbf{y}_t\}_{t=1}^n \subset \mathbb{R}^p$ of length n, and satisfying an (unkown) deterministic state-space of order d, or

$$\begin{cases} \mathbf{x}_{t+1} & = \mathbf{A}\mathbf{x}_t + \mathbf{v}_t \\ \mathbf{y} & = \mathbf{C}\mathbf{x}_t + \mathbf{w}_t \end{cases} \tag{10.16}$$

where $t = 1, \dots, n$ and $\mathbf{x}_0$ is fixed.
Problem: Recover

(a) The order d of the unknown system.

(b) The unknown system matrices $(\mathbf{A}, \mathbf{C})$.

Figure 10.2: The problem a stochastic Subspace Identification algorithms aims to solve.

the covariance matrices of the outputs are to match $\{\Lambda_\tau = \mathbb{E}[Y_t Y_{t+\tau}]\}_{\tau \geq 0}$. Ideas go along the line as set out in the previous subsection.

Such series of covariance matrices might be given equivalently as a spectral density function $\Phi(z)$. Here we use the following relation generalizing the z-transform

$$\Phi(z) = \sum_{\tau=-\infty}^{\infty} \Lambda_\tau z^{-\tau}, \tag{10.17}$$

and its inverse

$$\Lambda_\tau = \int_{-\infty}^{\infty} \Phi(z) z^{-\tau} dz. \tag{10.18}$$

The key once more is to consider the stochastic processes representing past and future. Specifically, given a process $Y = (\dots, Y_{t-1}, Y_t, Y_{t+1}, \dots)^T$ taking values as an infinite vector, define Y_t^+ and Y_t^- as follows

$$\begin{cases} Y_t^+ = (Y_t, Y_{t+1}, Y_{t+2} \dots)^T \\ Y_t^- = (Y_t, Y_{t-1}, Y_{t-2}, \dots)^T, \end{cases} \tag{10.19}$$

both taking values as one-sided infinite vectors. The next idea is to build up the covariance matrices associated to those stochastic processes. The devil is in the details here! The following (infinite dimensional) matrices are block Toeplitz:

$$\mathbf{L}_+ = \mathbb{E}\left[Y_t^+ Y_t^{+T}\right] = \begin{bmatrix} \Lambda_0 & \Lambda_1^T & \Lambda_2^T & \dots \\ \Lambda_1 & \Lambda_0 & \Lambda_1^T & \\ \Lambda_2 & \Lambda_1 & & \\ \vdots & & & \ddots \end{bmatrix} \tag{10.20}$$

and

$$\mathbf{L}_- = \mathbb{E}\left[Y_t^- Y_t^{-T}\right] = \begin{bmatrix} \Lambda_0 & \Lambda_1 & \Lambda_2 & \dots \\ \Lambda_1^T & \Lambda_0 & \Lambda_1 & \\ \Lambda_2^T & \Lambda_1^T & & \\ \vdots & & & \ddots \end{bmatrix} \tag{10.21}$$

The (infinite dimensional) block Hankel matrix $\mathbf{H}$ is here defined as

$$\mathbf{H} = \mathbb{E}\left[Y_t^- Y_t^{+T}\right] = \mathbb{E}\left[Y_t^+ Y_t^{-T}\right] = \begin{bmatrix} \Lambda_1 & \Lambda_2 & \Lambda_3 & \dots \\ \Lambda_2 & \Lambda_3 & \Lambda_4 & \\ \Lambda_3 & \Lambda_4 & & \\ \vdots & & & \ddots \end{bmatrix} \tag{10.22}$$

Now again we can factorize the matrix $\mathbf{H}$ into an observability part, and a controllability part. That is define the infinite observability 'matrix' corresponding with a stochastic MIMO system $\mathcal{S} = (\mathbf{A}, \mathbf{C})$ as

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \end{bmatrix}, \tag{10.23}$$

and let the infinite controllability 'matrix' of $\mathcal{S} = (\mathbf{A}, \mathbf{C})$ be given as

$$\mathcal{C} = \begin{bmatrix} \mathbf{C}^* & \mathbf{AC}^* & \mathbf{A}^2\mathbf{C}^* & \dots \end{bmatrix}. \tag{10.24}$$

Then we have again

Lemma 13 (Factorisation) *Let $\mathbf{H}, \mathcal{C}, \mathcal{O}$ be infinite 'matrices' as before associated with a stochastic MIMO system $\mathcal{S} = (\mathbf{A}, \mathbf{B})$. If the realization were driven by white noise we have for all $\tau \geq 1$ that*

$$\Lambda_\tau = \mathbb{E}[Y_t Y_{t+\tau}] = \begin{cases} \Lambda_0 & \tau = 0 \\ \mathbf{CA}^{\tau-1}\mathbf{C}^* & \tau \geq 1. \end{cases} \tag{10.25}$$

Moreover

$$\mathbf{H} = \mathcal{O}\mathcal{C}. \tag{10.26}$$

10.3 Further Work on Subspace Identification

10.4 Implementations of Subspace Identification

Chapter 11

Design of Experiments

”How to set up a data-collection experiment so as too guarantee accurately identified models?”