EXACT and APPROXIMATE

SYSTEM IDENTIFICATION

Jan C. Willems

K.U. Leuven, Belgium

University of Southern California **December 3, 2003**

Joint work with Ivan Markovsky (Un. of Leuven)

General Introduction

System identification

Observed data $\mapsto$ **System model**

System identification

Observed data $\mapsto$ System model

Case on interest:

Data = a finite vector time-series record

$$\tilde{w}(1), \tilde{w}(2), \ldots, \tilde{w}(T) \qquad w(t) \in \mathbb{R}^{\mathtt{w}}$$

Model:

a dynamical system that 'explains' this time-series

System identification

Difficulties:

- **'blackbox' data**
- **unmeasured inputs 'latency'**
- **any element of the model class will fit the data only approximately 'misfit'**
- **measurement 'errors'**
- **danger of 'overfitting'**

System identification

Usual approach: Data = input/output record

$$\begin{bmatrix} \tilde{u}(1) \\ \tilde{y}(1) \end{bmatrix}, \begin{bmatrix} \tilde{u}(2) \\ \tilde{y}(2) \end{bmatrix}, \ldots, \begin{bmatrix} \tilde{u}(T) \\ \tilde{y}(T) \end{bmatrix}$$

System model = an ARMAX model

$$P(\sigma)y = Q(\sigma)u + N(\sigma)\varepsilon$$

$\varepsilon =$ **'noise'**

Quality of algorithm

= asymptotic convergence $(T \to \infty)$

(consistency, efficiency, etc.)

System identification

System model = an ARMAX model

$$P(\sigma)y = Q(\sigma)u + N(\sigma)\varepsilon$$

$\varepsilon =$ 'noise'

Quality of algorithm

= asymptotic convergence $(T \to \infty)$

(consistency, efficiency, etc.)

In a sense this copes with these difficulties,

but puts stochasticity very central

Central paradigm

Algorithms should perform well with simulated data

Central paradigm

Algorithms should perform well with simulated data

- **What does 'perform well' mean?**
- **What 'simulated data' should one test the algorithm for?**

Central paradigm

Algorithms should perform well with simulated data

Methodology:

Exact ID

$\rightsquigarrow$ **Approximate ID (balancing, etc.)**

$\rightsquigarrow$ **Stochastic ID**

$\rightsquigarrow$ **Approximate stochastic ID**

Central paradigm

Algorithms should perform well with simulated data

Methodology:

Exact ID

$\rightsquigarrow$ **Approximate ID (balancing, etc.)**

$\rightsquigarrow$ **Stochastic ID**

$\rightsquigarrow$ **Approximate stochastic ID**

Approximation (data produced by high order, nonlinear, time-varying system) seems much more the core problem than protection against unmeasured stochastic inputs or measurement ‘errors’.

Deterministic system identification

Data = a vector time series

$$\tilde{w}(1), \tilde{w}(2), \ldots, \tilde{w}(T) \qquad w(t) \in \mathbb{R}^{\mathtt{w}}$$

System model = a linear time-invariant system

$$R(\sigma)w = 0$$

Quality of algorithm

= how does the algorithm behave with exact data?

= does it extract a good (optimal) approximation?

The MPUM

The MPUM

Ideas for exact modeling:

Assume $\exists$ a phenomenon that we wish to model,
produces outcomes, in the universum $\mathfrak{U}$

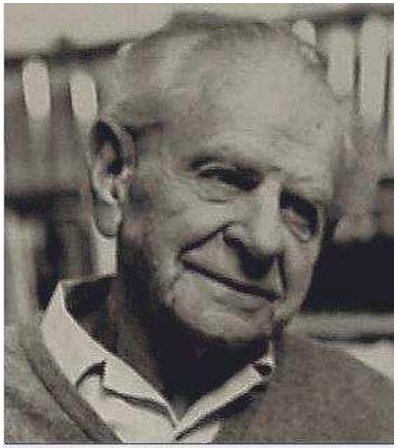
A model for the phenomenon = a subset $\mathfrak{B} \subseteq \mathfrak{U}$

The MPUM

A **model** for the phenomenon = a subset $\mathfrak{B} \subseteq \mathfrak{U}$

$\mathfrak{B}_1$ is **more powerful** than $\mathfrak{B}_2$ $:\Leftrightarrow$ $\mathfrak{B}_1 \subseteq \mathfrak{B}_2$

'the more a model forbids,
the better it is'



Sir Karl Popper (1902-1994)

The MPUM

A **model** for the phenomenon = a subset $\mathfrak{B} \subseteq \mathfrak{U}$

A **model class** := a family $\mathfrak{M}$ of subsets of $\mathfrak{U}$

The MPUM

A **model** for the phenomenon = a subset $\mathfrak{B} \subseteq \mathfrak{U}$

Data = a subset $\mathfrak{D} \subseteq \mathfrak{U}$, 'measured outcomes'.

$\mathfrak{B}$ is **unfalsified** by $\mathfrak{D}$ $:\Leftrightarrow$ $\mathfrak{D} \subseteq \mathfrak{B}$.

The MPUM

A model for the phenomenon = a subset $\mathfrak{B} \subseteq \mathfrak{U}$

$\mathfrak{B}^*$ is the Most Powerful Unfalsified Model MPUM in $\mathfrak{M}$ for $\mathfrak{D}$ $:\Leftrightarrow$

1. $\mathfrak{D} \subseteq \mathfrak{B}^* \in \mathfrak{M}$
2. $\mathfrak{D} \subseteq \mathfrak{B} \in \mathfrak{M} \implies \mathfrak{B}^* \subseteq \mathfrak{B}$

Does $\mathfrak{B}^*$ exist?

$\mathfrak{B}^*$ exists if

(i) $\mathfrak{U} \in \mathfrak{M}$

(ii) **$\mathfrak{M}$ is closed under intersection**

In this case, $\mathfrak{B}^* = \cap$ of the unfalsified models

Does $\mathfrak{B}^*$ exist?

$\mathfrak{B}^*$ **exists if**

(i) $\mathfrak{U} \in \mathfrak{M}$

(ii) $\mathfrak{M}$ is closed under intersection

In this case, $\mathfrak{B}^* = \cap$ of the unfalsified models

Examples:

$$\mathfrak{M} = 2^{\mathfrak{U}}; \;\; \mathfrak{B}^* = \mathfrak{D}$$

Does $\mathfrak{B}^*$ exist?

$\mathfrak{B}^*$ exists if

(i) $\mathfrak{U} \in \mathfrak{M}$

(ii) $\mathfrak{M}$ is closed under intersection

In this case, $\mathfrak{B}^* = \cap$ of the unfalsified models

Examples:

$\mathfrak{U} = \mathbb{R}^{\mathrm{n}}$, $\mathfrak{M} =$ all linear subspaces ,
$\mathfrak{B}^* = \mathrm{span}\{d \in \mathbb{R}^{\mathrm{n}} \mid d \in \mathfrak{D}\}$

Does $\mathfrak{B}^*$ exist?

$\mathfrak{B}^*$ **exists if**

(i) $\mathfrak{U} \in \mathfrak{M}$

(ii) $\mathfrak{M}$ **is closed under intersection**

In this case, $\mathfrak{B}^* = \cap$ **of the unfalsified models**

Examples:

$\mathfrak{D}$ **= a time-series**
$\mathfrak{M}$ **= the linear time-invariant systems**

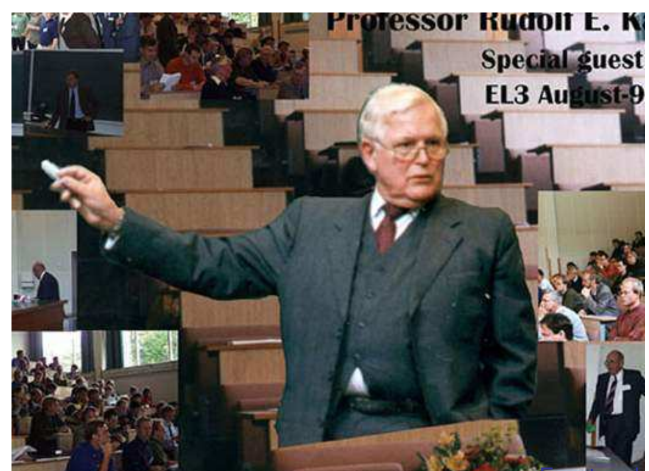
The model class

The model class

Time axis = $\mathbb{N}$ **(discrete-time systems)**

σ = **'backward shift'** $\rightsquigarrow$ $(\sigma f)(t) := f(t+1)$

$$
\begin{aligned}
\sigma x &= Ax + Bu \\
y &= Cx + Du \\
w &= \begin{bmatrix} u \\ y \end{bmatrix}
\end{aligned}
$$



The model class

$$\begin{aligned} \sigma x &= Ax + Bu \\ y &= Cx + Du \\ w &= \begin{bmatrix} u \\ y \end{bmatrix} \end{aligned}$$

Notation: $\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$; **impulse response matrix**

$$H : \mathbb{Z}_+ \to \mathbb{R}^{\mathtt{p}\times\mathtt{m}}; \quad H(0) = D, H(t) = CA^{t-1}B.$$

The model class

But, for good reasons, the (equivalent) representation as a system of linear difference equations

$$R_0 w(t) + R_1 w(t+1) + \cdots + R_L w(t+\ell) = 0 \qquad w = \begin{bmatrix} u \\ y \end{bmatrix}$$

is often to be preferred.

The model class

But, for good reasons, the (equivalent) representation as a system of linear difference equations

$$R_0 w(t) + R_1 w(t+1) + \cdots + R_L w(t+\ell) = 0 \qquad w = \begin{bmatrix} u \\ y \end{bmatrix}$$

is often to be preferred. With the polynomial matrix

$$R(\xi) = R_0 + R_1 \xi + \cdots + R_\ell \xi^\ell$$

these equations can be written as

$$\boxed{R(\sigma)w = 0}$$

The behavior of $R(\sigma)w = 0$

Call

$$\begin{aligned} \mathfrak{B} &= \{ w : \mathbb{N} \to \mathbb{R}^{\mathtt{w}} \mid R(\sigma)w = 0 \} \\ &= \ker(R(\sigma)) \end{aligned}$$

the ***'behavior'.***

The behavior of $R(\sigma)w = 0$

Call

$$\mathfrak{B} = \{w : \mathbb{N} \to \mathbb{R}^{\mathtt{w}} \mid R(\sigma)w = 0\} = \ker(R(\sigma))$$

the ***'behavior'.***

Any subset of $(\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$ which is

linear, shift-invariant, and closed

allows such a representation.

The behavior of $R(\sigma)w = 0$

Call

$$\begin{aligned}\mathfrak{B} &= \{w : \mathbb{N} \to \mathbb{R}^{\mathtt{w}} \mid R(\sigma)w = 0\} \\ &= \ker(R(\sigma))\end{aligned}$$

the ***'behavior'.***

Notation: $\mathfrak{L}^{\mathtt{w}}$, $\mathfrak{L}^{\bullet}$.

$\mathfrak{L}^{\bullet}$ **has very nice properties w.r.t. +, $\cap$, projection, action of linear difference operators, ...**

The behavior of $R(\sigma)w = 0$

Call

$$\begin{aligned} \mathfrak{B} &= \{ w : \mathbb{N} \to \mathbb{R}^{\mathtt{w}} \mid R(\sigma)w = 0 \} \\ &= \ker(R(\sigma)) \end{aligned}$$

the ***'behavior'***.

Notation: $\mathfrak{L}^{\mathtt{w}}$, $\mathfrak{L}^{\bullet}$.

Consequence: $\mathfrak{L}^{\mathtt{w}}$ **has intersection property,**

$\Longrightarrow$ **MPUM exists!**

The behavior generated by $\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$

Given $\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$, define its behavior as

$$\mathfrak{B} = \{ w = \begin{bmatrix} u \\ y \end{bmatrix} \mid \exists\, x \text{ such that } \sigma x = Ax + Bu,\ y = Cx + Du. \}$$

The behavior generated by $\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$

Given $\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$, define its behavior as

$$\mathfrak{B} = \{ w = \begin{bmatrix} u \\ y \end{bmatrix} \mid \exists\, x \text{ such that } \sigma x = Ax + Bu,\, y = Cx + Du. \}$$

Any $\mathfrak{B} = \ker(R(\sigma))$ allows an observable repr.
$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$. **Assumed henceforth.**

In behavioral theory

observability $\Leftrightarrow$ minimality of the state repr.

Computation of the MPUM

Computation of the MPUM

Given an observed vector time-series

$$\tilde{w}(-t), \ldots, \tilde{w}(0), \tilde{w}(1), \ldots, \tilde{w}(t), \ldots \qquad w(t) \in \mathbb{R}^{\mathtt{w}}$$

find a representation of the MPUM in $\mathfrak{L}^{\mathtt{w}}$.

'Exact, deterministic' system identification.

$\exists$ **algorithms (intersection of 'past' and 'future') that pass directly from**

$$\tilde{w}(-t), \ldots, \tilde{w}(0), \tilde{w}(1), \ldots, \tilde{w}(t), \ldots$$

$\downarrow\downarrow$ **to** $\downarrow\downarrow$

$$\tilde{x}(-t), \ldots, \tilde{x}(0), \tilde{x}(1), \ldots, \tilde{x}(t), \ldots$$

$\exists$ **algorithms (intersection of 'past' and 'future') that pass directly from**

$$\tilde{w}(-t), \ldots, \tilde{w}(0), \tilde{w}(1), \ldots, \tilde{w}(t), \ldots$$

$\downarrow\downarrow$ **to** $\downarrow\downarrow$

$$\tilde{x}(-t), \ldots, \tilde{x}(0), \tilde{x}(1), \ldots, \tilde{x}(t), \ldots$$

Solve (LS)

$$\begin{bmatrix} \tilde{x}(t_1+1) & \cdots & \tilde{x}(t_2) \\ \tilde{y}(t_1) & \cdots & \tilde{y}(t_2-1) \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \tilde{x}(t_1) & \cdots & \tilde{x}(t_2-1) \\ \tilde{u}(t_1) & \cdots & \tilde{u}(t_2-1) \end{bmatrix}$$

This yields a state representation of the MPUM.

Solve (LS)

$$\begin{bmatrix} \tilde{x}(t_1+1) & \cdots & \tilde{x}(t_2) \\ \tilde{y}(t_1) & \cdots & \tilde{y}(t_2-1) \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \tilde{x}(t_1) & \cdots & \tilde{x}(t_2-1) \\ \tilde{u}(t_1) & \cdots & \tilde{u}(t_2-1) \end{bmatrix}$$

This yields a state representation of the MPUM. $\rightsquigarrow$
Reduce the state dimension, and solve by LS using reduced

$$\begin{bmatrix} \tilde{x}(t_1) & \ldots & \tilde{x}(t_2) \end{bmatrix}.$$

Solve (LS)

$$\begin{bmatrix} \tilde{x}(t_1+1) & \cdots & \tilde{x}(t_2) \\ \tilde{y}(t_1) & \cdots & \tilde{y}(t_2-1) \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \tilde{x}(t_1) & \cdots & \tilde{x}(t_2-1) \\ \tilde{u}(t_1) & \cdots & \tilde{u}(t_2-1) \end{bmatrix}$$

This yields a state representation of the MPUM. $\rightsquigarrow$
Reduce the state dimension, and solve by LS using reduced

$$\begin{bmatrix} \tilde{x}(t_1) & \ldots & \tilde{x}(t_2) \end{bmatrix}.$$

This leads to the problem:

Construct $\tilde{x}(t_1), \ldots, \tilde{x}(t_2)$ **in a balanced basis.**

'Subspace methods' do this.

A ‘sequential’ zero input response series

A 'sequential' zero input response series

$$x(1) \xrightarrow{u(1)} x(2) \xrightarrow{u(2)} \cdots \xrightarrow{u(t-1)} x(t) \xrightarrow{u(t)} x(t+1) \xrightarrow{u(t+1)} \cdots$$

A ‘sequential’ zero input response series

$$\begin{array}{ccccccccccc}
x(1) & \xrightarrow{u(1)} & x(2) & \xrightarrow{u(2)} & \cdots & \xrightarrow{u(t-1)} & x(t) & \xrightarrow{u(t)} & x(t+1) & \xrightarrow{u(t+1)} & \cdots \\
\downarrow 0 & & \downarrow 0 & & \cdots & & \downarrow 0 & & \downarrow 0 & & \cdots \\
y_1(1) & & y_2(1) & & \cdots & & y_t(1) & & y_{t+1}(1) & & \cdots
\end{array}$$

A ‘sequential’ zero input response series

$$\begin{array}{cccccccccc}
x(1) & \xrightarrow{u(1)} & x(2) & \xrightarrow{u(2)} & \cdots & \xrightarrow{u(t-1)} & x(t) & \xrightarrow{u(t)} & x(t+1) & \xrightarrow{u(t+1)} \cdots \\
\downarrow 0 & & \downarrow 0 & & \cdots & & \downarrow 0 & & \downarrow 0 & \cdots \\
y_1(1) & & y_2(1) & & \cdots & & y_t(1) & & y_{t+1}(1) & \cdots \\
\downarrow 0 & & \downarrow 0 & & \cdots & & \downarrow 0 & & \downarrow 0 & \cdots \\
y_1(2) & & y_2(2) & & \cdots & & y_t(2) & & y_{t+1}(2) & \cdots
\end{array}$$

A 'sequential' zero input response series

$$
\begin{array}{ccccccccccc}
x(1) & \xrightarrow{u(1)} & x(2) & \xrightarrow{u(2)} & \cdots & \xrightarrow{u(t-1)} & x(t) & \xrightarrow{u(t)} & x(t+1) & \xrightarrow{u(t+1)} & \cdots \\
\downarrow 0 & & \downarrow 0 & & \cdots & & \downarrow 0 & & \downarrow 0 & & \cdots \\
y_1(1) & & y_2(1) & & \cdots & & y_t(1) & & y_{t+1}(1) & & \cdots \\
\downarrow 0 & & \downarrow 0 & & \cdots & & \downarrow 0 & & \downarrow 0 & & \cdots \\
y_1(2) & & y_2(2) & & \cdots & & y_t(2) & & y_{t+1}(2) & & \cdots \\
\downarrow 0 & & \downarrow 0 & & \cdots & & \downarrow 0 & & \downarrow 0 & & \cdots \\
\vdots & & \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\
\downarrow 0 & & \downarrow 0 & & \cdots & & \downarrow 0 & & \downarrow 0 & & \cdots \\
y_1(t') & & y_2(t') & & \cdots & & y_t(t') & & y_{t+1}(t') & & \cdots \\
\downarrow 0 & & \downarrow 0 & & \cdots & & \downarrow 0 & & \downarrow 0 & & \cdots \\
y_1(t'+1) & & y_2(t'+1) & & \cdots & & y_t(t'+1) & & y_{t+1}(t'+1) & & \cdots \\
\downarrow 0 & & \downarrow 0 & & \cdots & & \downarrow 0 & & \downarrow 0 & & \cdots \\
\vdots & & \vdots & & \vdots & & \vdots & & \vdots & & \vdots
\end{array}
$$

A 'sequential' zero input response series

Organized into the matrix

$$Y_0 := \begin{bmatrix} y_1(1) & y_2(1) & \cdots & y_t(1) & y_{t+1}(1) & \cdots \\ y_1(2) & y_2(2) & \cdots & y_t(2) & y_{t+1}(2) & \cdots \\ \cdot & \cdot & \cdots & \cdot & \cdot & \cdots \\ y_1(t') & y_2(t') & \cdots & y_t(t') & y_{t+1}(t') & \cdots \\ \cdot & \cdot & \cdots & \cdot & \cdot & \cdots \end{bmatrix}$$

A 'sequential' zero input response series

Organized into the matrix

$$Y_0 := \begin{bmatrix} y_1(1) & y_2(1) & \cdots & y_t(1) & y_{t+1}(1) & \cdots \\ y_1(2) & y_2(2) & \cdots & y_t(2) & y_{t+1}(2) & \cdots \\ . & . & \cdots & . & . & \cdots \\ y_1(t') & y_2(t') & \cdots & y_t(t') & y_{t+1}(t') & \cdots \\ . & . & \cdots & . & . & \cdots \end{bmatrix}$$

Note

$$x(t+1) = Ax(t) + Bu(t); \qquad \text{for some } u(\cdot)$$

$$Y_0 = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{t'-1} \\ \vdots \end{bmatrix} \begin{bmatrix} x(1) & x(2) & \cdots & x(t) & \cdots \end{bmatrix}$$

How does deterministic subspace identification work ?

Deterministic subspace identification

There are basically five steps. Use the data

$$\begin{bmatrix} \tilde{u}(1) \\ \tilde{y}(1) \end{bmatrix}, \begin{bmatrix} \tilde{u}(2) \\ \tilde{y}(2) \end{bmatrix}, \ldots, \begin{bmatrix} \tilde{u}(T) \\ \tilde{y}(T) \end{bmatrix}$$

to compute (an estimate of)

Deterministic subspace identification

1. a sequential zero input response series matrix of the system that produced the data $\rightsquigarrow$ Y_0

Deterministic subspace identification

1. a **sequential zero input response series** $\rightsquigarrow$ Y_0
2. the **impulse response matrix** $H : \mathbb{Z}_+ \rightarrow \mathbb{R}^{\mathtt{p} \times \mathtt{m}}$ of the system that produced the data

   $\rightsquigarrow$ **the Hankel matrix** $\mathfrak{H}$

Deterministic subspace identification

1. a **sequential zero input response series** $\rightsquigarrow$ Y_0
2. the **impulse response matrix** $\rightsquigarrow$ $\mathfrak{H}$
3. an **SVD** of this Hankel matrix $\mathfrak{H} = U\Sigma V^\top$

Deterministic subspace identification

1. a **sequential zero input response series** $\rightsquigarrow$ Y_0
2. the **impulse response matrix** $\rightsquigarrow$ $\mathfrak{H}$
3. an **SVD** of this Hankel matrix $\mathfrak{H} = U\Sigma V^\top$
4. the **balanced state trajectory**

$$\begin{bmatrix} x(1) & x(2) & \cdots & x(t) & \cdots \end{bmatrix} = \sqrt{\Sigma^{-1}}\, U^\top Y_0$$

Deterministic subspace identification

1. a **sequential zero input response series** $\rightsquigarrow$ Y_0
2. the **impulse response matrix** $\rightsquigarrow$ $\mathfrak{H}$
3. an **SVD** of this Hankel matrix $\mathfrak{H} = U\Sigma V^\top$
4. the **balanced state trajectory**

$$\begin{bmatrix} x(1) & x(2) & \cdots & x(t) & \cdots \end{bmatrix} = \sqrt{\Sigma^{-1}}\, U^\top Y_0$$

5. (LS) solve, with u, y, x a (data ind.) system traj.

$$\begin{bmatrix} x(2) & x(3) & \cdots & x(t+1) & \cdots \\ y(1) & y(2) & \cdots & y(t) & \cdots \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x(1) & x(2) & \cdots & x(t) & \cdots \\ u(1) & u(2) & \cdots & u(t) & \cdots \end{bmatrix}$$

This yields a desired balanced state representation.

The question is

How do we compute all these responses, starting from the data ?

The behavior of $R(\sigma)w = 0$

Call

$$\begin{aligned} \mathfrak{B} &= \{ w : \mathbb{N} \to \mathbb{R}^{\mathtt{w}} \mid R(\sigma)w = 0 \} \\ &= \ker(R(\sigma)) \end{aligned}$$

the ***'behavior'.***

The behavior of $R(\sigma)w = 0$

Call

$$\begin{aligned} \mathfrak{B} &= \{w : \mathbb{N} \to \mathbb{R}^{\mathtt{w}} \mid R(\sigma)w = 0\} \\ &= \ker(R(\sigma)) \end{aligned}$$

the ***'behavior'.***

Any subset of $(\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$ which is

linear, shift-invariant, and closed

allows such a representation.

The behavior of $R(\sigma)w = 0$

Call

$$\begin{aligned} \mathfrak{B} &= \{w : \mathbb{N} \to \mathbb{R}^{\mathtt{w}} \mid R(\sigma)w = 0\} \\ &= \ker(R(\sigma)) \end{aligned}$$

the ***'behavior'.*** Consider also its ***'annihilators'***

$$\mathfrak{N}_{\mathfrak{B}} = \{n \in \mathbb{R}^{\mathtt{w}}(\xi) \mid n^{\top}(\sigma)\mathfrak{B} = 0\}$$

$$n_0 + n_1\xi + \cdots + n_\ell\xi^\ell \in \mathfrak{N}_{\mathfrak{B}} :\Leftrightarrow$$

$$n_0^\top w(t) + n_1^\top w(t+1) + \cdots + n_\ell^\top w(t+\ell) = 0$$

for all $w \in \mathfrak{B}$ and $t \in \mathbb{N}$

The behavior of $R(\sigma)w = 0$

Call

$$\begin{aligned} \mathfrak{B} &= \{w : \mathbb{N} \to \mathbb{R}^{\mathtt{w}} \mid R(\sigma)w = 0\} \\ &= \ker(R(\sigma)) \end{aligned}$$

the *'behavior'*. Consider also its *'annihilators'*

$$\mathfrak{N}_{\mathfrak{B}} = \{n \in \mathbb{R}^{\mathtt{w}}(\xi) \mid n^{\top}(\sigma)\mathfrak{B} = 0\}$$

Note: (the transpose of) each row of R belongs to $\mathfrak{N}_{\mathfrak{B}}$.

$\mathfrak{N}_{\mathfrak{B}}$ = the module generated by the transposes of the rows of R.

Properties and invariants of $\mathfrak{B}$

Each notion has a version for each representation,

$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$, $\mathfrak{B}$, and $\mathfrak{N}_{\mathfrak{B}}$.

We give the most convenient one.

Properties and invariants of $\mathfrak{B}$

Controllability

$\mathtt{m}(\mathfrak{B}), \mathtt{p}(\mathfrak{B}), \mathtt{n}(\mathfrak{B})$ **:= input, output, state dimension**

Properties and invariants of $\mathfrak{B}$

$\ell(\mathfrak{B})$:= the *lag* in $\mathfrak{B}$

= the degree of R in a *'shortest lag'* repr.

$$R(\sigma)w = 0$$

= the observability index

= the narrowest window through which 'legality' of $w \in \mathfrak{B}$ can be determined.

There holds:

$$\ell(\mathfrak{B}) \leq \mathtt{n}(\mathfrak{B})$$

with = in the single output case.

Properties and invariants of $\mathfrak{B}$

$\ell(\mathfrak{B})$:= the *lag* in $\mathfrak{B}$

$\mathfrak{B}|_{[1,\Delta]}$:= the behavior restr. to the interval $[1, \Delta]$
= the ‘legal’ prefixes of length Δ

Properties and invariants of $\mathfrak{B}$

$\ell(\mathfrak{B})$:= the *lag* in $\mathfrak{B}$

$\mathfrak{B}|_{[1,\Delta]}$:= the behavior restr. to the interval $[1, \Delta]$
= the 'legal' prefixes of length Δ

$\mathfrak{N}^{\mathrm{d}}_{\mathfrak{B}}$:= the annihilators of degree $\leq \mathtt{d}$.

$$n_0^\top w(t) + n_1^\top w(t+1) + \cdots + n_{\mathtt{d}}^\top w(t+\mathtt{d}) = 0$$

for all $w \in \mathfrak{B}$ and $t \in \mathbb{N}$.

Properties and invariants of $\mathfrak{B}$

It follows that

$$w \in \mathfrak{B} \Leftrightarrow \begin{bmatrix} n_0^\top & \cdots & n_{\ell(\mathfrak{B})}^\top \end{bmatrix} \begin{bmatrix} w(t) \\ \vdots \\ w(t+\ell(\mathfrak{B})) \end{bmatrix} = 0$$

for all $n \in \mathfrak{N}_{\mathfrak{B}}^{\ell(\mathfrak{B})}, t \in \mathbb{N}$

$$\Leftrightarrow \begin{bmatrix} w(t) \\ \vdots \\ w(t+\ell(\mathfrak{B})) \end{bmatrix} \in \mathfrak{B}|_{[1,\ell(\mathfrak{B})+1]}$$ for all $t \in \mathbb{N}$.

Properties and invariants of $\mathfrak{B}$

Hence, if $\Delta > \ell(\mathfrak{B})$,

$$w \in \mathfrak{B} \Leftrightarrow \begin{bmatrix} n_0^\top & \cdots & n_{\Delta-1}^\top \end{bmatrix} \begin{bmatrix} w(t) \\ \vdots \\ w(t+\Delta-1) \end{bmatrix} = 0$$

for all $n \in \mathfrak{N}_{\mathfrak{B}}^{\Delta-1}, t \in \mathbb{N}$

$$\Leftrightarrow \begin{bmatrix} w(t) \\ \vdots \\ w(t+\Delta-1) \end{bmatrix} \in \mathfrak{B}|_{[1,\Delta]} \quad \textbf{for all} \quad t \in \mathbb{N}.$$

Properties and invariants of $\mathfrak{B}$

Hence, if $\Delta > \ell(\mathfrak{B})$, $\mathfrak{B}$ is uniquely determined by its **'short'** sequences and **'short'** annihilators

$$\mathfrak{B}|_{[1,\Delta]} \quad \text{and} \quad \mathfrak{N}_{\mathfrak{B}}^{\Delta-1}.$$

Properties and invariants of $\mathfrak{B}$

Another consequence. Consider

$$\left(\begin{bmatrix}\tilde{u}'(1)\\ \tilde{y}'(1)\end{bmatrix},\ldots,\begin{bmatrix}\tilde{u}'(t-\Delta)\\ \tilde{y}'(t-\Delta)\end{bmatrix},\begin{bmatrix}\tilde{u}'(t-\Delta+1)\\ \tilde{y}'(t-\Delta+1)\end{bmatrix},\ldots,\begin{bmatrix}\tilde{u}'(t)\\ \tilde{y}'(t)\end{bmatrix}\right)\in\mathfrak{B}|_{[1,t]}$$

$$\left(\begin{bmatrix}\tilde{u}''(1)\\ \tilde{y}''(1)\end{bmatrix},\ldots,\begin{bmatrix}\tilde{u}''(\Delta)\\ \tilde{y}''(\Delta)\end{bmatrix},\begin{bmatrix}\tilde{u}''(\Delta+1)\\ \tilde{y}''(\Delta+1)\end{bmatrix},\quad\ldots\quad,\begin{bmatrix}\tilde{u}''(t)\\ \tilde{y}''(t)\end{bmatrix}\right)\in\mathfrak{B}|_{[1,t]}$$

Properties and invariants of $\mathfrak{B}$

Another consequence. Assume suffix' = prefix".

$$\left(\begin{bmatrix}\tilde u'(1)\\ \tilde y'(1)\end{bmatrix},\dots,\begin{bmatrix}\tilde u'(t-\Delta)\\ \tilde y'(t-\Delta)\end{bmatrix},\begin{bmatrix}\tilde u'(t-\Delta+1)\\ \tilde y'(t-\Delta+1)\end{bmatrix},\dots,\begin{bmatrix}\tilde u'(t)\\ \tilde y'(t)\end{bmatrix}\right)$$

$$//$$

$$\left(\begin{bmatrix}\tilde u''(1)\\ \tilde y''(1)\end{bmatrix},\dots,\begin{bmatrix}\tilde u''(\Delta)\\ \tilde y''(\Delta)\end{bmatrix},\begin{bmatrix}\tilde u''(\Delta+1)\\ \tilde y''(\Delta+1)\end{bmatrix},\dots,\begin{bmatrix}\tilde u''(t)\\ \tilde y''(t)\end{bmatrix}\right)$$

Properties and invariants of $\mathfrak{B}$

Another consequence. Assume **suffix' = prefix''.**

Then their linking

$$\left(\begin{bmatrix}\tilde{u}'(1)\\ \tilde{y}'(1)\end{bmatrix},\ldots,\begin{bmatrix}\tilde{u}'(t-\Delta)\\ \tilde{y}'(t-\Delta)\end{bmatrix},\begin{bmatrix}\tilde{u}'(t-\Delta+1)\\ \tilde{y}'(t-\Delta+1)\end{bmatrix},\ldots,\begin{bmatrix}\tilde{u}'(t)\\ \tilde{y}'(t)\end{bmatrix},\begin{bmatrix}\tilde{u}''(\Delta+1)\\ \tilde{y}''(\Delta+1)\end{bmatrix},\ldots,\begin{bmatrix}\tilde{u}''(t)\\ \tilde{y}''(t)\end{bmatrix}\right)$$

belongs to $\mathfrak{B}|_{[1,2t-\Delta]}$,

if $\Delta \geq \ell(\mathfrak{B})$, **hence if** $\Delta \geq \mathtt{n}(\mathfrak{B})$.

Fundamental lemma

Key question

Assume that the vector time-series

$$\begin{bmatrix} \tilde{u}(1) \\ \tilde{y}(1) \end{bmatrix}, \begin{bmatrix} \tilde{u}(2) \\ \tilde{y}(2) \end{bmatrix}, \ldots, \begin{bmatrix} \tilde{u}(T) \\ \tilde{y}(T) \end{bmatrix}$$

has been produced by $\mathfrak{B}$.

Key question

Assume that the vector time-series

$$\begin{bmatrix} \tilde{u}(1) \\ \tilde{y}(1) \end{bmatrix}, \begin{bmatrix} \tilde{u}(2) \\ \tilde{y}(2) \end{bmatrix}, \ldots, \begin{bmatrix} \tilde{u}(T) \\ \tilde{y}(T) \end{bmatrix}$$

has been produced by $\mathfrak{B}$.

Then, of course, the vectors

$$\begin{bmatrix} \tilde{u}(1) \\ \tilde{y}(1) \\ \vdots \\ \tilde{u}(\Delta) \\ \tilde{y}(\Delta) \end{bmatrix}, \begin{bmatrix} \tilde{u}(2) \\ \tilde{y}(2) \\ \vdots \\ \tilde{u}(\Delta+1) \\ \tilde{y}(\Delta+1) \end{bmatrix}, \ldots, \begin{bmatrix} \tilde{u}(T-\Delta+1) \\ \tilde{y}(T-\Delta+1) \\ \vdots \\ \tilde{u}(T) \\ \tilde{y}(T) \end{bmatrix}$$

belong to $\mathfrak{B}|_{[1,\Delta]}$.

Key question

$$\begin{bmatrix} \tilde{u}(1) \\ \tilde{y}(1) \\ \vdots \\ \tilde{u}(\Delta) \\ \tilde{y}(\Delta) \end{bmatrix}, \begin{bmatrix} \tilde{u}(2) \\ \tilde{y}(2) \\ \vdots \\ \tilde{u}(\Delta+1) \\ \tilde{y}(\Delta+1) \end{bmatrix}, \ldots, \begin{bmatrix} \tilde{u}(T-\Delta+1) \\ \tilde{y}(T-\Delta+1) \\ \vdots \\ \tilde{u}(T) \\ \tilde{y}(T) \end{bmatrix}$$

Under what conditions on

$$\begin{bmatrix} \tilde{u}(1) \\ \tilde{y}(1) \end{bmatrix}, \begin{bmatrix} \tilde{u}(2) \\ \tilde{y}(2) \end{bmatrix}, \ldots, \begin{bmatrix} \tilde{u}(T) \\ \tilde{y}(T) \end{bmatrix} \text{ and } \mathfrak{B}$$

do they span $\mathfrak{B}|_{[1,\Delta]}$ ***and hence, if*** $\Delta > \ell(\mathfrak{B})$, ***determine the generating behavior*** $\mathfrak{B}$ ***?***

Persistency of excitation

The vector time-series

$$\tilde{u}(1), \tilde{u}(2), \ldots \tilde{u}(T)$$

is said to be ***persistently exciting of order L*** if the Hankel matrix

$$\begin{bmatrix} \tilde{u}(1) & \tilde{u}(2) & \tilde{u}(3) & \cdots & \tilde{u}(T-L+1) \\ \tilde{u}(2) & \tilde{u}(3) & \tilde{u}(4) & \cdots & \tilde{u}(T-L+2) \\ \tilde{u}(3) & \tilde{u}(4) & \tilde{u}(5) & \cdots & \tilde{u}(T-L+3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \tilde{u}(L) & \tilde{u}(L+1) & \tilde{u}(L+2) & \cdots & \tilde{u}(T) \end{bmatrix}$$

is of full row rank. Pers. of exc. $\Leftrightarrow$ no linear relations.

Fundamental lemma

Assume that the observed vector time-series

$$\begin{bmatrix} \tilde{u}(1) \\ \tilde{y}(1) \end{bmatrix}, \begin{bmatrix} \tilde{u}(2) \\ \tilde{y}(2) \end{bmatrix}, \ldots, \begin{bmatrix} \tilde{u}(T) \\ \tilde{y}(T) \end{bmatrix}$$

has been generated by a controllable finite dimensional linear time-invariant system $\rightsquigarrow$ behavior $\mathfrak{B}$.

Fundamental lemma

Assume that the observed vector time-series

$$\begin{bmatrix} \tilde{u}(1) \\ \tilde{y}(1) \end{bmatrix}, \begin{bmatrix} \tilde{u}(2) \\ \tilde{y}(2) \end{bmatrix}, \ldots, \begin{bmatrix} \tilde{u}(T) \\ \tilde{y}(T) \end{bmatrix}$$

has been generated by a **controllable** finite dimensional linear time-invariant system $\rightsquigarrow$ behavior $\mathfrak{B}$. Then the vectors

$$\begin{bmatrix} \tilde{u}(1) \\ \tilde{y}(1) \\ \vdots \\ \tilde{u}(\Delta) \\ \tilde{y}(\Delta) \end{bmatrix}, \begin{bmatrix} \tilde{u}(2) \\ \tilde{y}(2) \\ \vdots \\ \tilde{u}(\Delta+1) \\ \tilde{y}(\Delta+1) \end{bmatrix}, \ldots, \begin{bmatrix} \tilde{u}(T-\Delta+1) \\ \tilde{y}(T-\Delta+1) \\ \vdots \\ \tilde{u}(T) \\ \tilde{y}(T) \end{bmatrix}$$

span $\mathfrak{B}|_{[1,\Delta]}$ if $\tilde{u}(1), \ldots, \tilde{u}(T)$ is persistently exc. of order

???

Fundamental lemma

Assume that the observed vector time-series

$$\begin{bmatrix} \tilde{u}(1) \\ \tilde{y}(1) \end{bmatrix}, \begin{bmatrix} \tilde{u}(2) \\ \tilde{y}(2) \end{bmatrix}, \ldots, \begin{bmatrix} \tilde{u}(T) \\ \tilde{y}(T) \end{bmatrix}$$

has been generated by a **controllable** finite dimensional linear time-invariant system $\rightsquigarrow$ behavior $\mathfrak{B}$. Then the vectors

$$\begin{bmatrix} \tilde{u}(1) \\ \tilde{y}(1) \\ \vdots \\ \tilde{u}(\Delta) \\ \tilde{y}(\Delta) \end{bmatrix}, \begin{bmatrix} \tilde{u}(2) \\ \tilde{y}(2) \\ \vdots \\ \tilde{u}(\Delta+1) \\ \tilde{y}(\Delta+1) \end{bmatrix}, \ldots, \begin{bmatrix} \tilde{u}(T-\Delta+1) \\ \tilde{y}(T-\Delta+1) \\ \vdots \\ \tilde{u}(T) \\ \tilde{y}(T) \end{bmatrix}$$

span $\mathfrak{B}|_{[1,\Delta]}$ if $\tilde{u}(1), \ldots, \tilde{u}(T)$ is persistently exc. of order

$$\boxed{\Delta + \mathrm{n}(\mathfrak{B})}$$

Fundamental lemma

Hence, under the assumptions of

1. controllability and 2. persistency of excitation,

the span (& hence left annihilators) of the data vectors

$$\begin{bmatrix} \tilde{u}(1) \\ \tilde{y}(1) \\ \tilde{u}(2) \\ \tilde{y}(2) \\ \vdots \\ \tilde{u}(\Delta) \\ \tilde{y}(\Delta) \end{bmatrix}, \begin{bmatrix} \tilde{u}(2) \\ \tilde{y}(2) \\ \tilde{u}(3) \\ \tilde{y}(3) \\ \vdots \\ \tilde{u}(\Delta+1) \\ \tilde{y}(\Delta+1) \end{bmatrix}, \ldots, \begin{bmatrix} \tilde{u}(T-\Delta+1) \\ \tilde{y}(T-\Delta+1) \\ \tilde{u}(T-\Delta+2) \\ \tilde{y}(T-\Delta+2) \\ \vdots \\ \tilde{u}(T) \\ \tilde{y}(T) \end{bmatrix}$$

determines $\mathfrak{B}$, provided 3. $\boxed{\Delta > \ell(\mathfrak{B})}$

Conclusion

Under reasonable conditions

(contr., Δ suff. large, persistency of excitation),

the data matrix

$$\begin{bmatrix} \tilde{U} \\ \tilde{Y} \end{bmatrix} = \begin{bmatrix} \tilde{u}(1) & \tilde{u}(2) & \cdots & \tilde{u}(T-\Delta+1) \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{u}(\Delta) & \tilde{u}(\Delta+1) & \cdots & \tilde{u}(T) \\ \tilde{y}(1) & \tilde{y}(2) & \cdots & \tilde{y}(T-\Delta+1) \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{y}(\Delta) & \tilde{y}(\Delta+1) & \cdots & \tilde{y}(T) \end{bmatrix}$$

has the 'correct' span and the 'correct' left kernel.

Conclusion

Under reasonable conditions the data matrix has the 'correct' span and the 'correct' left kernel.

$\Rightarrow$ any response, in particular, seq. zero input resp., impulse resp., etc., can be obtained by solving

$$\begin{bmatrix} u(1) \\ \vdots \\ u(\Delta) \\ y(1) \\ \vdots \\ y(\Delta) \end{bmatrix} = \begin{bmatrix} \tilde{U} \\ \tilde{Y} \end{bmatrix} g$$

Conclusion

and linking and solving, with $\mathtt{n}_{\max} \geq \ell(\mathfrak{B})$ or $\mathtt{n}(\mathfrak{B})$,

$$
\begin{bmatrix}
u(1) \\ \vdots \\ u(\Delta - \mathtt{n}_{\max}) \\ u(\Delta - \mathtt{n}_{\max} + 1) \\ \vdots \\ u(\Delta) \\ y(1) \\ \vdots \\ y(\Delta - \mathtt{n}_{\max}) \\ y(\Delta - \mathtt{n}_{\max} + 1) \\ \vdots \\ y(\Delta)
\end{bmatrix}
\quad \nearrow \nearrow \quad
\begin{bmatrix}
u(\Delta - \mathtt{n}_{\max} + 1) \\ \vdots \\ u(\Delta) \\ u(\Delta + 1) \\ \vdots \\ u(2\Delta - \mathtt{n}_{\max}) \\ y(\Delta - \mathtt{n}_{\max} + 1) \\ \vdots \\ y(\Delta) \\ y(\Delta + 1) \\ \vdots \\ y(2\Delta - \mathtt{n}_{\max})
\end{bmatrix}
=
\begin{bmatrix}
\tilde{U} \\ \tilde{Y}
\end{bmatrix}
g
$$

Conclusion

and proceeding recursively

$$
\begin{bmatrix}
u(\Delta'+1) \\ \vdots \\ u(\Delta'+\mathrm{n_{max}}) \\
u(\Delta'+\mathrm{n_{max}}+1) \\ \vdots \\ u(\Delta'+\Delta) \\
y(\Delta'+\mathrm{n_{max}}) \\ \vdots \\ y(\Delta) \\
y(\Delta'+\mathrm{n_{max}}+1) \\ \vdots \\ y(\Delta'+\Delta)
\end{bmatrix}
\begin{matrix} \nearrow \quad \nearrow \\ \\ \nearrow \quad \nearrow \end{matrix}
\begin{bmatrix}
u(\Delta'+\mathrm{n_{max}}+1) \\ \vdots \\ u(\Delta+\Delta') \\
u(\Delta'+\Delta+1) \\ \vdots \\ u(\Delta+\Delta'+\mathrm{n_{max}}) \\
y(\Delta'+\mathrm{n_{max}}+1) \\ \vdots \\ y(\Delta+\Delta') \\
y(\Delta'+\Delta+1) \\ \vdots \\ y(\Delta+\Delta'+\mathrm{n_{max}})
\end{bmatrix}
=
\begin{bmatrix} \tilde{U} \\ \tilde{Y} \end{bmatrix} g
$$

Conclusion

This way, an arbitrary long sequence

$$\left(\begin{bmatrix} u(1) \\ y(1) \end{bmatrix}, \begin{bmatrix} u(2) \\ y(2) \end{bmatrix}, \ldots, \begin{bmatrix} u(t) \\ y(t) \end{bmatrix}\right) \in \mathfrak{B}|_{[1,t]}$$

can be obtained.

Note: These algorithms allow nicely for (LS) approximate computations.

An idea of the proof

An idea of the proof

Assume

$$\begin{bmatrix} \tilde{u}(1) \\ \tilde{y}(1) \end{bmatrix}, \begin{bmatrix} \tilde{u}(2) \\ \tilde{y}(2) \end{bmatrix}, \ldots, \begin{bmatrix} \tilde{u}(T) \\ \tilde{y}(T) \end{bmatrix} \in \mathfrak{B}_{[1,T]}.$$

An idea of the proof

Assume

$$\begin{bmatrix} \tilde{u}(1) \\ \tilde{y}(1) \end{bmatrix}, \begin{bmatrix} \tilde{u}(2) \\ \tilde{y}(2) \end{bmatrix}, \ldots, \begin{bmatrix} \tilde{u}(T) \\ \tilde{y}(T) \end{bmatrix} \in \mathfrak{B}_{[1,T]}.$$

SISO case $\rightsquigarrow$ $\mathfrak{B}$ **determined by**

$$p_n y(t+n) + p_{n-1} y(t+n-1) + \cdots + p_0 y(t)$$
$$= q_n u(t+n) + q_{n-1} u(t+n-1) + \cdots + q_0 u(t)$$

$$\boxed{p(\sigma) y = q(\sigma) u}$$

$$\begin{aligned} p(\xi) &= p_0 + p_1 \xi + \cdots + p_n \xi^n, \qquad \text{with } p_n \neq 0, \\ q(\xi) &= q_0 + q_1 \xi + \cdots + q_n \xi^n. \end{aligned}$$

An idea of the proof

Assume

$$\begin{bmatrix} \tilde{u}(1) \\ \tilde{y}(1) \end{bmatrix}, \begin{bmatrix} \tilde{u}(2) \\ \tilde{y}(2) \end{bmatrix}, \ldots, \begin{bmatrix} \tilde{u}(T) \\ \tilde{y}(T) \end{bmatrix} \in \mathfrak{B}_{[1,T]}.$$

$$\boxed{p(\sigma)y = q(\sigma)u}$$

$$\begin{aligned} p(\xi) &= p_0 + p_1\xi + \cdots + p_n\xi^n, \qquad \text{with } p_n \neq 0, \\ q(\xi) &= q_0 + q_1\xi + \cdots + q_n\xi^n. \end{aligned}$$

$$\mathfrak{N}_{\mathfrak{B}} = \text{span}\left\{ \begin{bmatrix} -q(\xi) \\ p(\xi) \end{bmatrix}, \begin{bmatrix} -\xi q(\xi) \\ \xi p(\xi) \end{bmatrix}, \ldots, \begin{bmatrix} -\xi^k q(\xi) \\ \xi^k p(\xi) \end{bmatrix}, \ldots \right\}$$

An idea of the proof

Data matrix:

$$\begin{bmatrix} \tilde{u}(1) & \tilde{u}(2) & \cdots & \tilde{u}(T-\Delta+1) \\ \tilde{u}(2) & \tilde{u}(3) & \cdots & \tilde{u}(T-\Delta+2) \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{u}(\Delta) & \tilde{u}(\Delta+1) & \cdots & \tilde{u}(T) \\ \tilde{y}(1) & \tilde{y}(2) & \cdots & \tilde{y}(T-\Delta+1) \\ \tilde{y}(2) & \tilde{y}(3) & \cdots & \tilde{y}(T-\Delta+2) \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{y}(\Delta) & \tilde{y}(\Delta+1) & \cdots & \tilde{y}(T) \end{bmatrix}$$

An idea of the proof

For $\Delta = \mathtt{n} + 1$, the left kernel contains

$$\begin{bmatrix} -q_0 & -q_1 & \cdots & -q_{\mathtt{n}} & p_0 & p_1 & \cdots & p_{\mathtt{n}} \end{bmatrix}.$$

An idea of the proof

For $\Delta > \mathrm{n}+1,$ the left kernel contains the rows of

$$\begin{bmatrix} -q_0 & \cdots & -q_{\mathrm{n}} & 0 & \cdots & 0 & p_0 & \cdots & p_{\mathrm{n}} & 0 & \cdots & 0 \\ 0 & -q_0 & \cdots & -q_{\mathrm{n}} & \cdots & 0 & 0 & p_0 & \cdots & p_{\mathrm{n}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & -q_0 & \cdots & -q_{\mathrm{n}} & 0 & \cdots & 0 & p_0 & \cdots & p_{\mathrm{n}} \end{bmatrix} \begin{matrix} \leftarrow \text{row } 1 \\ \leftarrow \text{row } 2 \\ \\ \leftarrow \text{row } \Delta - \mathrm{n} \end{matrix}$$

An idea of the proof

For $\Delta > \mathrm{n}+1,$ the left kernel contains the rows of

$$\begin{bmatrix} -q_0 & \cdots & -q_{\mathrm{n}} & 0 & \cdots & 0 & p_0 & \cdots & p_{\mathrm{n}} & 0 & \cdots & 0 \\ 0 & -q_0 & \cdots & -q_{\mathrm{n}} & \cdots & 0 & 0 & p_0 & \cdots & p_{\mathrm{n}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & -q_0 & \cdots & -q_{\mathrm{n}} & 0 & \cdots & 0 & p_0 & \cdots & p_{\mathrm{n}} \end{bmatrix} \begin{matrix} \leftarrow \text{row } 1 \\ \leftarrow \text{row } 2 \\ \\ \leftarrow \text{row } \Delta-\mathrm{n} \end{matrix}$$

Assume that the kernel contains another vector, not in their span

$$\begin{bmatrix} r_0 & \cdots & \cdot & \cdot & \cdots & r_{\Delta-1} & s_0 & \cdots & \cdot & \cdot & \cdots & s_{\Delta-1} \end{bmatrix}$$

An idea of the proof

Extend the data matrix to a larger window:

$$\begin{bmatrix} \tilde{u}(1) & \tilde{u}(2) & \cdots & \tilde{u}(T-\Delta'+1) \\ \tilde{u}(2) & \tilde{u}(3) & \cdots & \tilde{u}(T-\Delta'+2) \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{u}(\Delta') & \tilde{u}(\Delta'+1) & \cdots & \tilde{u}(T) \\ \tilde{y}(1) & \tilde{y}(2) & \cdots & \tilde{y}(T-\Delta'+1) \\ \tilde{y}(2) & \tilde{y}(3) & \cdots & \tilde{y}(T-\Delta'+2) \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{y}(\Delta') & \tilde{y}(\Delta'+1) & \cdots & \tilde{y}(T) \end{bmatrix}$$

An idea of the proof

Then the left kernel contains the rows of

$$\begin{bmatrix} -q_0 & \cdots & -q_n & 0 & \cdots & 0 & p_0 & \cdots & p_n & 0 & \cdots & 0 \\ 0 & -q_0 & \cdots & -q_n & \cdots & 0 & 0 & p_0 & \cdots & p_n & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & -q_0 & \cdots & -q_n & 0 & \cdots & 0 & p_0 & \cdots & p_n \end{bmatrix} \begin{matrix} \leftarrow \text{row } 1 \\ \leftarrow \text{row } 2 \\ \\ \leftarrow \text{row } \Delta' - \mathrm{n} \end{matrix}$$

An idea of the proof

Then the left kernel contains the rows of

$$
\begin{bmatrix}
-q_0 & \cdots & -q_n & 0 & \cdots & 0 & p_0 & \cdots & p_n & 0 & \cdots & 0 \\
0 & -q_0 & \cdots & -q_n & \cdots & 0 & 0 & p_0 & \cdots & p_n & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
0 & \cdots & 0 & -q_0 & \cdots & -q_n & 0 & \cdots & 0 & p_0 & \cdots & p_n
\end{bmatrix}
\begin{matrix}
\leftarrow \text{row } 1 \\
\leftarrow \text{row } 2 \\
\\
\leftarrow \text{row } \Delta' - n
\end{matrix}
$$

$$
\begin{bmatrix}
r_0 & \cdot & r_{\Delta-1} & 0 & \cdot & 0 & s_0 & \cdot & s_{\Delta-1} & 0 & \cdot & 0 \\
0 & r_0 & \cdot & r_{\Delta-1} & \cdot & 0 & 0 & s_0 & \cdot & s_{\Delta-1} & \cdot & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & \cdot & 0 & r_0 & \cdot & r_{\Delta-1} & 0 & \cdot & 0 & s_0 & \cdot & s_{\Delta-1}
\end{bmatrix}
\begin{matrix}
\leftarrow \text{row } 1 \\
\leftarrow \text{row } 2 \\
\\
\leftarrow \text{row } \Delta' - \Delta + 1
\end{matrix}
$$

An idea of the proof

Then the left kernel contains the rows of

$$\begin{bmatrix} -q_0 & \cdots & -q_n & 0 & \cdots & 0 & p_0 & \cdots & p_n & 0 & \cdots & 0 \\ 0 & -q_0 & \cdots & -q_n & \cdots & 0 & 0 & p_0 & \cdots & p_n & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & -q_0 & \cdots & -q_n & 0 & \cdots & 0 & p_0 & \cdots & p_n \end{bmatrix} \begin{matrix} \leftarrow \text{row } 1 \\ \leftarrow \text{row } 2 \\ \\ \leftarrow \text{row } \Delta' - n \end{matrix}$$

$$\begin{bmatrix} r_0 & \cdot & r_{\Delta-1} & 0 & \cdot & 0 & s_0 & \cdot & s_{\Delta-1} & 0 & \cdot & 0 \\ 0 & r_0 & \cdot & r_{\Delta-1} & \cdot & 0 & 0 & s_0 & \cdot & s_{\Delta-1} & \cdot & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdot & 0 & r_0 & \cdot & r_{\Delta-1} & 0 & \cdot & 0 & s_0 & \cdot & s_{\Delta-1} \end{bmatrix} \begin{matrix} \leftarrow \text{row } 1 \\ \leftarrow \text{row } 2 \\ \\ \leftarrow \text{row } \Delta'-\Delta+1 \end{matrix}$$

If all rows were linearly independent, then at each extension step, the rank of the data matrix remains constant. But, persistency of excitation $\Rightarrow$ the rank increases by 1. $\rightsquigarrow$ **conflict,** when $\Delta' = \Delta + n$.

An idea of the proof

Then the left kernel contains the rows of

$$\begin{bmatrix} -q_0 & \cdots & -q_n & 0 & \cdots & 0 & p_0 & \cdots & p_n & 0 & \cdots & 0 \\ 0 & -q_0 & \cdots & -q_n & \cdots & 0 & 0 & p_0 & \cdots & p_n & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & -q_0 & \cdots & -q_n & 0 & \cdots & 0 & p_0 & \cdots & p_n \end{bmatrix} \begin{matrix} \leftarrow \text{row } 1 \\ \leftarrow \text{row } 2 \\ \\ \leftarrow \text{row } \Delta' - n \end{matrix}$$

$$\begin{bmatrix} r_0 & \cdot & r_{\Delta-1} & 0 & \cdot & 0 & s_0 & \cdot & s_{\Delta-1} & 0 & \cdot & 0 \\ 0 & r_0 & \cdot & r_{\Delta-1} & \cdot & 0 & 0 & s_0 & \cdot & s_{\Delta-1} & \cdot & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdot & 0 & r_0 & \cdot & r_{\Delta-1} & 0 & \cdot & 0 & s_0 & \cdot & s_{\Delta-1} \end{bmatrix} \begin{matrix} \leftarrow \text{row } 1 \\ \leftarrow \text{row } 2 \\ \\ \leftarrow \text{row } \Delta' - \Delta + 1 \end{matrix}$$

Therefore one of the rows of the second matrix must be linearly dependent on the rows preceding it and the rows of the first matrix.

Written in polynomial notation, this yields

An idea of the proof

$$f(\xi)\left[r(\xi) \quad | \quad s(\xi)\right] = h(\xi)\left[-q(\xi) \quad | \quad p(\xi)\right]$$

with, without loss of generality, f and h co-prime.

An idea of the proof

$$f(\xi)\begin{bmatrix} r(\xi) & | & s(\xi) \end{bmatrix} = h(\xi)\begin{bmatrix} -q(\xi) & | & p(\xi) \end{bmatrix}$$

with, without loss of generality, f and h co-prime.
This means that f must be a factor of both p and q.

An idea of the proof

$$f(\xi)\begin{bmatrix} r(\xi) & | & s(\xi) \end{bmatrix} = h(\xi)\begin{bmatrix} -q(\xi) & | & p(\xi) \end{bmatrix}$$

with, without loss of generality, f and h co-prime.
This means that f must be a factor of both p and q.

If degree$(f) > 0$,

this contradicts the fact that $\mathfrak{B}$ is controllable.

An idea of the proof

$$f(\xi)\left[r(\xi) \quad | \quad s(\xi)\right] = h(\xi)\left[-q(\xi) \quad | \quad p(\xi)\right]$$

with, without loss of generality, f and h co-prime.
This means that f must be a factor of both p and q.

If degree$(f) > 0$,

this contradicts the fact that $\mathfrak{B}$ is controllable.

Whence, $f = 1$, but then

$$\left[r(\xi) \quad | \quad s(\xi)\right] = h(\xi)\left[-q(\xi) \quad | \quad p(\xi)\right]$$

An idea of the proof

and hence

$$\begin{bmatrix} r_0 & \cdots & \cdot & \cdot & \cdots & r_{\Delta-1} & s_0 & \cdots & \cdot & \cdot & \cdots & s_{\Delta-1} \end{bmatrix}$$

is in the span of the rows of

$$\begin{bmatrix} -q_0 & \cdots & -q_n & 0 & \cdots & 0 & p_0 & \cdots & p_n & 0 & \cdots & 0 \\ 0 & -q_0 & \cdots & -q_n & \cdots & 0 & 0 & p_0 & \cdots & p_n & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & -q_0 & \cdots & -q_n & 0 & \cdots & 0 & p_0 & \cdots & p_n \end{bmatrix} \begin{matrix} \leftarrow \text{row } 1 \\ \leftarrow \text{row } 2 \\ \\ \leftarrow \text{row } \Delta - n \end{matrix}$$

An idea of the proof

and hence

$$\begin{bmatrix} r_0 & \cdots & \cdot & \cdot & \cdots & r_{\Delta-1} & s_0 & \cdots & \cdot & \cdot & \cdots & s_{\Delta-1} \end{bmatrix}$$

is in the span of the rows of

$$\begin{bmatrix} -q_0 & \cdots & -q_n & 0 & \cdots & 0 & p_0 & \cdots & p_n & 0 & \cdots & 0 \\ 0 & -q_0 & \cdots & -q_n & \cdots & 0 & 0 & p_0 & \cdots & p_n & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & -q_0 & \cdots & -q_n & 0 & \cdots & 0 & p_0 & \cdots & p_n \end{bmatrix} \begin{matrix} \leftarrow \text{row } 1 \\ \leftarrow \text{row } 2 \\ \\ \leftarrow \text{row } \Delta - n \end{matrix}$$

Therefore, the data matrix had the **'correct'** kernel to begin with. **QED**

From time-series to balanced reduction

1. How can we compute a sequential zero input response series?

Define the 'past' and 'future' input and output data matrices by

$$\begin{bmatrix} \tilde{U}_{\mathrm{p}} \\ \tilde{Y}_{\mathrm{p}} \\ \tilde{U}_{\mathrm{f}} \\ \tilde{Y}_{\mathrm{f}} \end{bmatrix} = \begin{bmatrix} \tilde{u}(1) & \tilde{u}(2) & \cdots & \tilde{u}(T-2\Delta+1) \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{u}(\Delta) & \tilde{u}(\Delta+1) & \cdots & \tilde{u}(T-\Delta) \\ \tilde{y}(1) & \tilde{y}(2) & \cdots & \tilde{y}(T-2\Delta+1) \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{y}(\Delta) & \tilde{y}(\Delta+1) & \cdots & \tilde{y}(T-\Delta) \\ \tilde{u}(\Delta+1) & \tilde{u}(\Delta+2) & \cdots & \tilde{u}(T-\Delta+1) \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{u}(2\Delta) & \tilde{u}(2\Delta+1) & \cdots & \tilde{u}(T) \\ \tilde{y}(\Delta+1) & \tilde{y}(\Delta+2) & \cdots & \tilde{y}(T-\Delta+1) \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{y}(2\Delta) & \tilde{y}(2\Delta+1) & \cdots & \tilde{y}(T) \end{bmatrix}$$

Assume $\mathtt{n}(\mathfrak{B}) \ll \Delta \ll T$ & pers. of excitation, as needed.

1. Solve for ?? (through G) in

$$\begin{bmatrix} \tilde{U}_p \\ \tilde{Y}_p \\ O \\ ?? \end{bmatrix} = \begin{bmatrix} \tilde{U}_p \\ \tilde{Y}_p \\ \tilde{U}_f \\ \tilde{Y}_f \end{bmatrix} G$$

1. Solve for $??$ (through G) in

$$\begin{bmatrix} \tilde{U}_{\mathrm{p}} \\ \tilde{Y}_{\mathrm{p}} \\ O \\ ?? \end{bmatrix} = \begin{bmatrix} \tilde{U}_{\mathrm{p}} \\ \tilde{Y}_{\mathrm{p}} \\ \tilde{U}_{\mathrm{f}} \\ \tilde{Y}_{\mathrm{f}} \end{bmatrix} G$$

$??$ is obviously a seq. zero input resp. $\rightsquigarrow$ Y_0.

1. Solve for $??$ (through G) in

$$\begin{bmatrix} \tilde{U}_{\text{p}} \\ \tilde{Y}_{\text{p}} \\ O \\ ?? \end{bmatrix} = \begin{bmatrix} \tilde{U}_{\text{p}} \\ \tilde{Y}_{\text{p}} \\ \tilde{U}_{\text{f}} \\ \tilde{Y}_{\text{f}} \end{bmatrix} G$$

$??$ is obviously a seq. zero input resp. $\rightsquigarrow$ Y_0.

Y_0 = the oblique projection of the row span of $\tilde{Y}_{\text{f}}$,

along the row span of $\tilde{U}_{\text{f}}$, onto the row span of $\begin{bmatrix} \tilde{U}_{\text{p}} \\ \tilde{Y}_{\text{p}} \end{bmatrix}$!

2. How can we compute (an estimate of) the Hankel matrix?

2. How can we compute (an estimate of) the Hankel matrix?

By solving for G in:

$$\begin{bmatrix} O_{m\times m} & O_{m\times m} & \cdots & O_{m\times m} & I_{m\times m} \\ O_{m\times m} & O_{m\times m} & \cdots & I_{m\times m} & O_{m\times m} \\ \vdots & \vdots & \nearrow & \vdots & \vdots \\ O_{m\times m} & I_{m\times m} & \cdots & O_{m\times m} & O_{m\times m} \\ I_{m\times m} & O_{m\times m} & \cdots & O_{m\times m} & O_{m\times m} \\ \star & \star & \cdots & \star & \star \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \star & \star & \cdots & \star & \star \\ O_{m\times m} & O_{m\times m} & \cdots & O_{m\times m} & O_{m\times m} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ O_{m\times m} & O_{m\times m} & \cdots & O_{m\times m} & O_{m\times m} \\ \hat{H}(1) & \hat{H}(2) & \cdots & \hat{H}(\Delta-1) & \hat{H}(\Delta) \\ \hat{H}(2) & \hat{H}(3) & \cdots & \hat{H}(\Delta) & \hat{H}(\Delta+1) \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \hat{H}(\Delta-1) & \hat{H}(\Delta) & \cdots & \hat{H}(2\Delta-3) & \hat{H}(2\Delta-2) \\ \hat{H}(\Delta) & \hat{H}(\Delta+1) & \cdots & \hat{H}(2\Delta-2) & \hat{H}(2\Delta-1) \end{bmatrix} = \begin{bmatrix} \tilde{U}_{\mathrm{p}} \\ \tilde{Y}_{\mathrm{p}} \\ \tilde{U}_{\mathrm{f}} \\ \tilde{Y}_{\mathrm{f}} \end{bmatrix} G$$

Or, since the columns of $\begin{bmatrix} \tilde{U}_{\text{p}} \\ \tilde{Y}_{\text{p}} \\ O \\ Y_0 \end{bmatrix}$ are spanned by the columns of $\begin{bmatrix} \tilde{U}_{\text{p}} \\ \tilde{Y}_{\text{p}} \\ \tilde{U}_{\text{f}} \\ \tilde{Y}_{\text{f}} \end{bmatrix}$, by solving for G' in

$$\begin{bmatrix} O_{m\times m} & O_{m\times m} & \cdots & O_{m\times m} & I_{m\times m} \\ O_{m\times m} & O_{m\times m} & \cdots & I_{m\times m} & O_{m\times m} \\ \vdots & \vdots & \nearrow & \vdots & \vdots \\ O_{m\times m} & I_{m\times m} & \cdots & O_{m\times m} & O_{m\times m} \\ I_{m\times m} & O_{m\times m} & \cdots & O_{m\times m} & O_{m\times m} \\ \star & \star & \cdots & \star & \star \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \star & \star & \cdots & \star & \star \\ O_{m\times m} & O_{m\times m} & \cdots & O_{m\times m} & O_{m\times m} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ O_{m\times m} & O_{m\times m} & \cdots & O_{m\times m} & O_{m\times m} \\ \hat{H}(1) & \hat{H}(2) & \cdots & \hat{H}(\Delta-1) & \hat{H}(\Delta) \\ \hat{H}(2) & \hat{H}(3) & \cdots & \hat{H}(\Delta) & \hat{H}(\Delta+1) \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \hat{H}(\Delta-1) & \hat{H}(\Delta) & \cdots & \hat{H}(2\Delta-3) & \hat{H}(2\Delta-2) \\ \hat{H}(\Delta) & \hat{H}(\Delta+1) & \cdots & \hat{H}(2\Delta-2) & \hat{H}(2\Delta-1) \end{bmatrix} = \begin{bmatrix} \tilde{U}_p \\ \tilde{Y}_p \\ O \\ Y_0 \end{bmatrix} G'$$

Solution $\rightsquigarrow$ $$G' = \tilde{U}_\mathrm{p}^\top (\tilde{U}_\mathrm{p}\tilde{U}_\mathrm{p}^\top)^{-1} J$$

where

$$J = \begin{bmatrix} O_{\mathtt{m}\times\mathtt{m}} & O_{\mathtt{m}\times\mathtt{m}} & \cdots & O_{\mathtt{m}\times\mathtt{m}} & I_{\mathtt{m}\times\mathtt{m}} \\ O_{\mathtt{m}\times\mathtt{m}} & O_{\mathtt{m}\times\mathtt{m}} & \cdots & I_{\mathtt{m}\times\mathtt{m}} & O_{\mathtt{m}\times\mathtt{m}} \\ \cdot & \cdot & \nearrow & \cdot & \cdot \\ O_{\mathtt{m}\times\mathtt{m}} & I_{\mathtt{m}\times\mathtt{m}} & \cdots & O_{\mathtt{m}\times\mathtt{m}} & O_{\mathtt{m}\times\mathtt{m}} \\ I_{\mathtt{m}\times\mathtt{m}} & O_{\mathtt{m}\times\mathtt{m}} & \cdots & O_{\mathtt{m}\times\mathtt{m}} & O_{\mathtt{m}\times\mathtt{m}} \end{bmatrix}$$

Solution $\rightsquigarrow$ $$G' = \tilde{U}_{\mathsf{p}}^\top (\tilde{U}_{\mathsf{p}} \tilde{U}_{\mathsf{p}}^\top)^{-1} J$$

$\rightsquigarrow$ the following estimate of the Hankel matrix $\mathfrak{H}$:

$$\hat{\mathfrak{H}} := \begin{bmatrix} \hat{H}(1) & \hat{H}(2) & \cdots & \hat{H}(\Delta-1) & \hat{H}(\Delta) \\ \hat{H}(2) & \hat{H}(3) & \cdots & \hat{H}(\Delta) & \hat{H}(\Delta+1) \\ \cdot & \cdot & \cdots & \cdot & \cdot \\ \hat{H}(\Delta-1) & \hat{H}(\Delta) & \cdots & \hat{H}(2\Delta-3) & \hat{H}(2\Delta-2) \\ \hat{H}(\Delta) & \hat{H}(\Delta+1) & \cdots & \hat{H}(2\Delta-2) & \hat{H}(2\Delta-1) \end{bmatrix}$$

$$\hat{\mathfrak{H}} = Y_0 \, \tilde{U}_{\mathsf{p}}^\top (\tilde{U}_{\mathsf{p}} \tilde{U}_{\mathsf{p}}^\top)^{-1} J$$

Solution $\rightsquigarrow$

$$G' = \tilde{U}_{\mathsf{p}}^{\top} (\tilde{U}_{\mathsf{p}} \tilde{U}_{\mathsf{p}}^{\top})^{-1} J$$

$\rightsquigarrow$ the following estimate of the Hankel matrix $\mathfrak{H}$:

$$\hat{\mathfrak{H}} = Y_0 \, \tilde{U}_{\mathsf{p}}^{\top} (\tilde{U}_{\mathsf{p}} \tilde{U}_{\mathsf{p}}^{\top})^{-1} J$$

Note: no new eq'ns to be solved, once we have Y_0.

3. **SVD this Hankel matrix** $\rightsquigarrow$ $\hat{\mathfrak{H}} = U\Sigma V^\top$

3. **SVD this Hankel matrix** $\rightsquigarrow$ $\hat{\mathfrak{H}} = U\Sigma V^\top$

4. **Obtain the balanced state trajectory**

$$\begin{bmatrix} \hat{x}(\Delta+1) & \hat{x}(\Delta+2) & \cdots & \hat{x}(T-\Delta+1) \end{bmatrix} = \sqrt{\Sigma^{-1}}\, U^\top Y_0$$

3. SVD this Hankel matrix $\rightsquigarrow$ $\hat{\mathfrak{H}} = U\Sigma V^\top$

4. Obtain the balanced state trajectory

$$\begin{bmatrix}\hat{x}(\Delta+1) & \hat{x}(\Delta+2) & \cdots & \hat{x}(T-\Delta+1)\end{bmatrix} = \sqrt{\Sigma^{-1}}\,U^\top Y_0$$

$\hat{x}(\Delta+1), \hat{x}(\Delta+2), \cdots, \hat{x}(T-\Delta+1)$ are estimates of a balanced state traj. separating the 'past' and 'future'.

$$
\begin{bmatrix} \tilde{U}_{\mathrm{p}} \\ \tilde{Y}_{\mathrm{p}} \\ \hat{X} \\ \tilde{U}_{\mathrm{f}} \\ \tilde{Y}_{\mathrm{f}} \end{bmatrix}
=
\begin{bmatrix}
\tilde{u}(1) & \tilde{u}(2) & \cdots & \tilde{u}(T-2\Delta+1) \\
\vdots & \vdots & \cdots & \vdots \\
\tilde{u}(\Delta) & \tilde{u}(\Delta+1) & \cdots & \tilde{u}(T-\Delta) \\
\tilde{y}(1) & \tilde{y}(2) & \cdots & \tilde{y}(T-2\Delta+1) \\
\vdots & \vdots & \cdots & \vdots \\
\tilde{y}(\Delta) & \tilde{y}(\Delta+1) & \cdots & \tilde{y}(T-\Delta) \\
\hat{x}(\Delta+1) & \hat{x}(\Delta+2) & \cdots & \hat{x}(T-\Delta+1) \\
\tilde{u}(\Delta+1) & \tilde{u}(\Delta+2) & \cdots & \tilde{u}(T-\Delta+1) \\
\vdots & \vdots & \cdots & \vdots \\
\tilde{u}(2\Delta) & \tilde{u}(2\Delta+1) & \cdots & \tilde{u}(T) \\
\tilde{y}(\Delta+1) & \tilde{y}(\Delta+2) & \cdots & \tilde{y}(T-\Delta+1) \\
\vdots & \vdots & \cdots & \vdots \\
\tilde{y}(2\Delta) & \tilde{y}(2\Delta+1) & \cdots & \tilde{y}(T)
\end{bmatrix}
$$

3. **SVD** this Hankel matrix $\rightsquigarrow \quad \hat{\mathfrak{H}} = U\Sigma V^\top$

4. Obtain the **balanced state trajectory**

$$\begin{bmatrix} \hat{x}(\Delta+1) & \hat{x}(\Delta+2) & \cdots & \hat{x}(T-\Delta+1) \end{bmatrix} = \sqrt{\Sigma^{-1}}\, U^\top Y_0$$

5. Compute the (LS) sol'n of the **linear equations**

$$\begin{bmatrix} \hat{x}(\Delta+2) & \cdots & \hat{x}(T-\Delta+1) \\ \tilde{y}(\Delta+1) & \cdots & \tilde{y}(T-\Delta) \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \hat{x}(\Delta+1) & \cdots & \hat{x}(T-\Delta) \\ \tilde{u}(\Delta+1) & \cdots & \tilde{u}(T-\Delta) \end{bmatrix}$$

This solution yields the desired **balanced** system.

More on this and other algorithms, soon on my website

Simulations

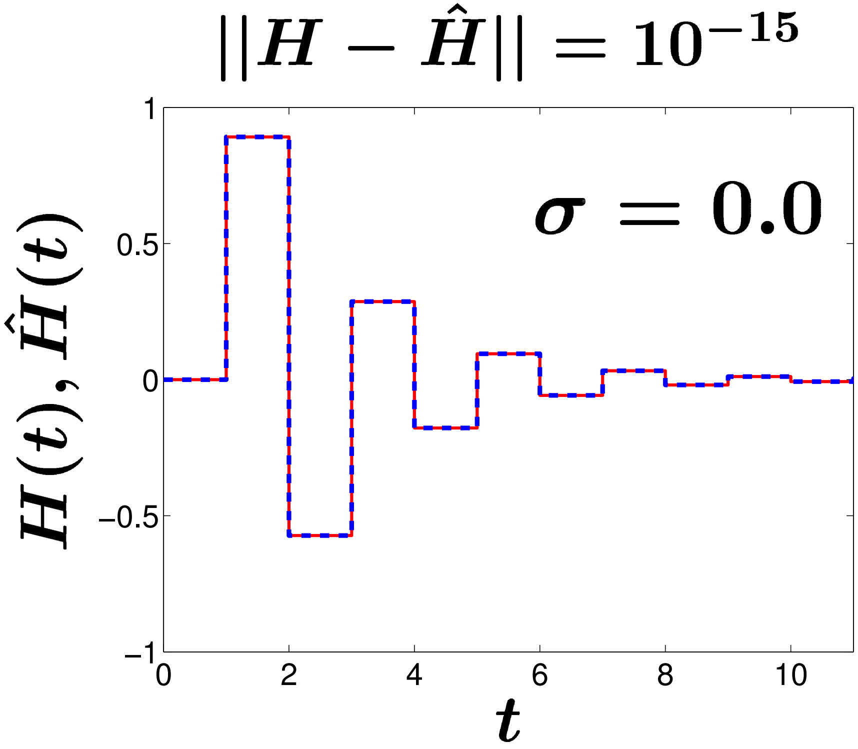
Simulations

In all simulations the system has a transfer function

$$C(Iz - A)^{-1}B + D = \frac{0.89172(z - 0.5193)(z + 0.5595)}{(z - 0.4314)(z + 0.4987)(z + 0.6154)}.$$

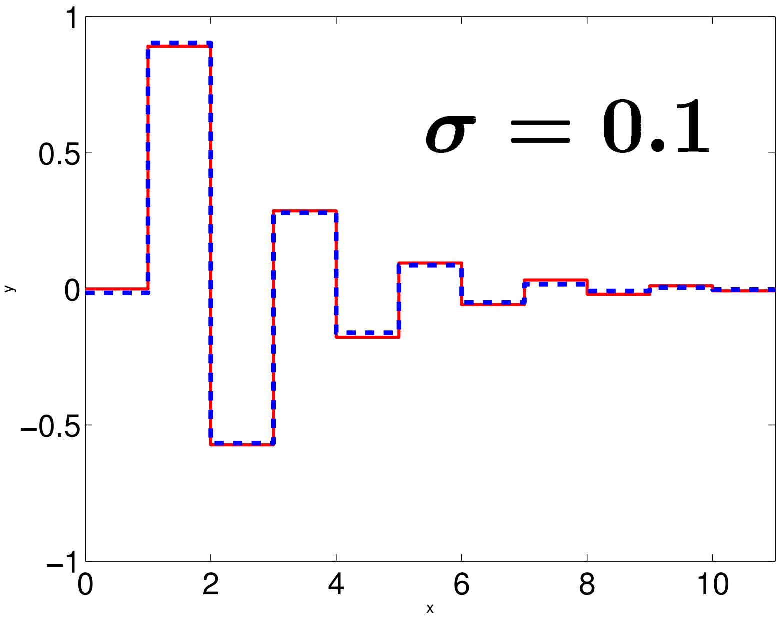
The input is a unit variance white noise and the data available for identification is the corresponding trajectory $w = (u, y)$, corrupted by white noise with standard deviation σ.

Simulations



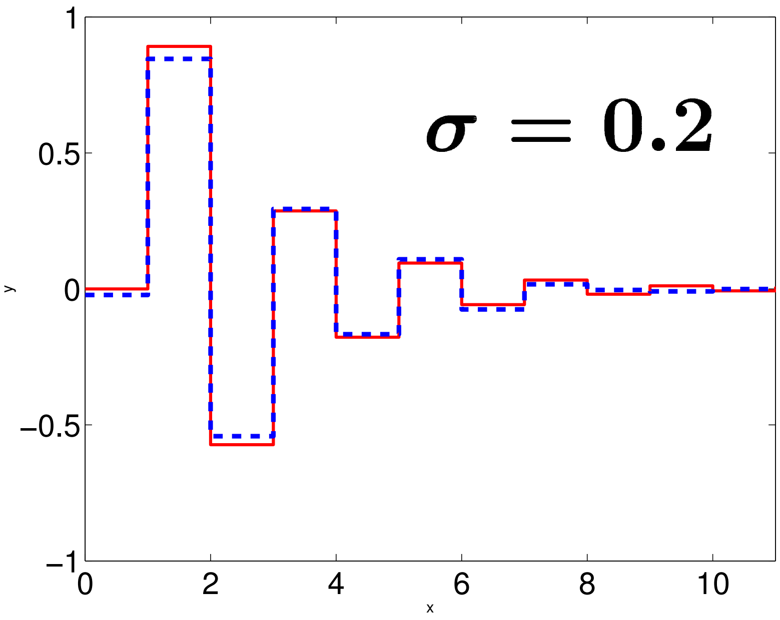
Simulations

$$||H - \hat{H}|| = 0.02$$



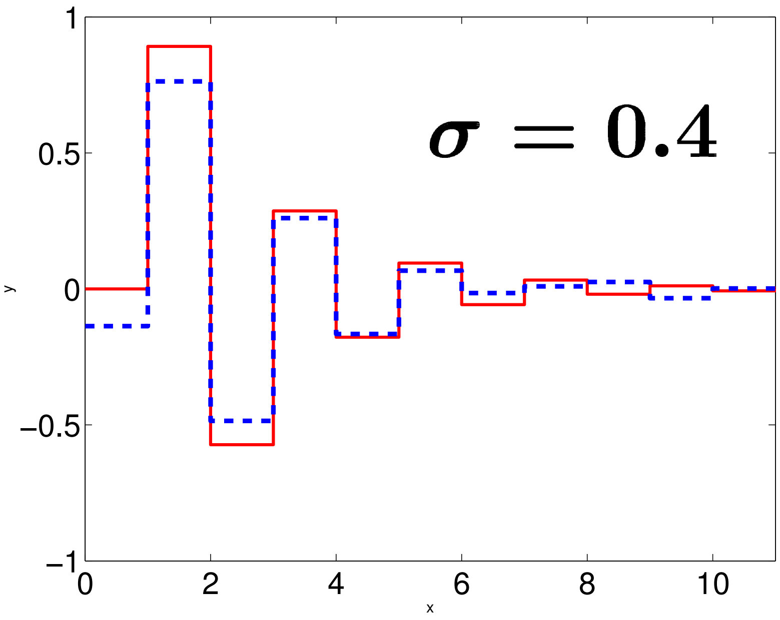
Simulations

$$||H - \hat{H}|| = 0.05$$



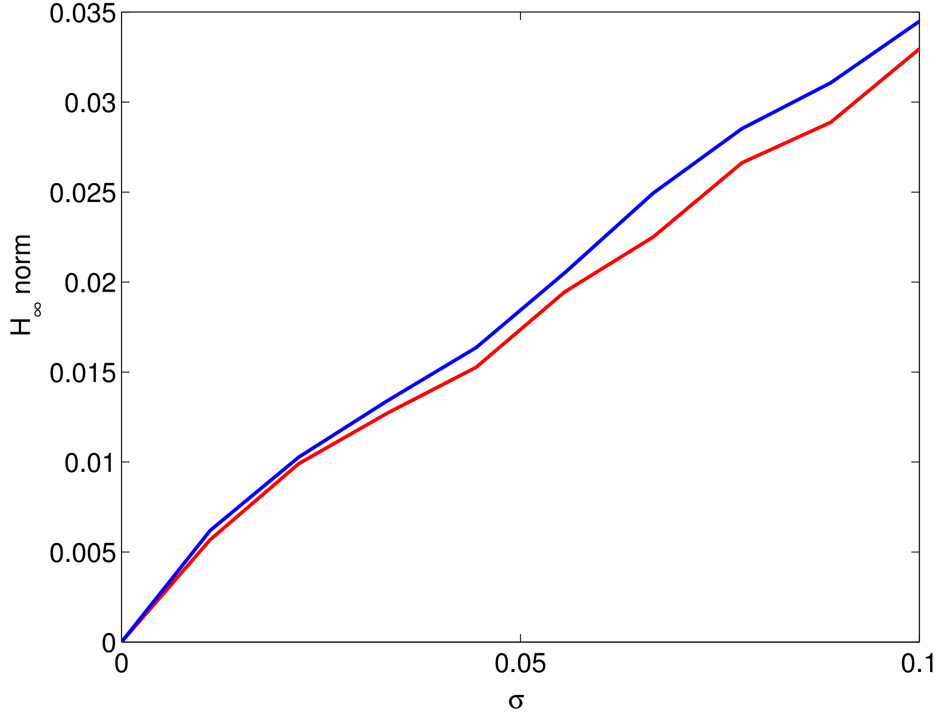
Simulations

$$||H - \hat{H}|| = 0.21$$

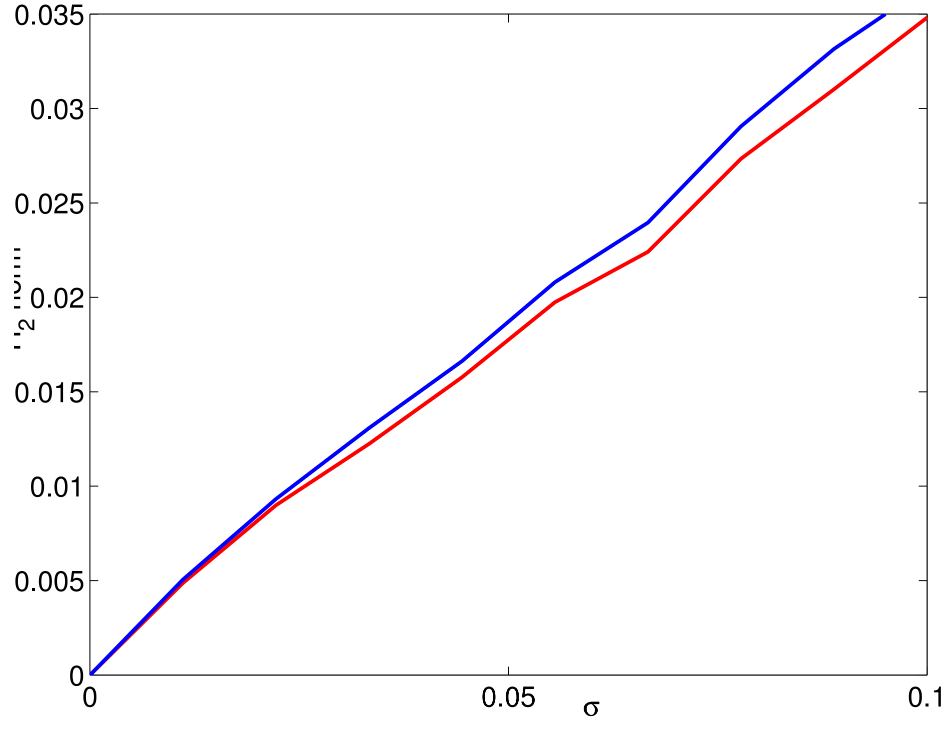


Simulations

Improvement over balancing from $\hat{H}$ directly

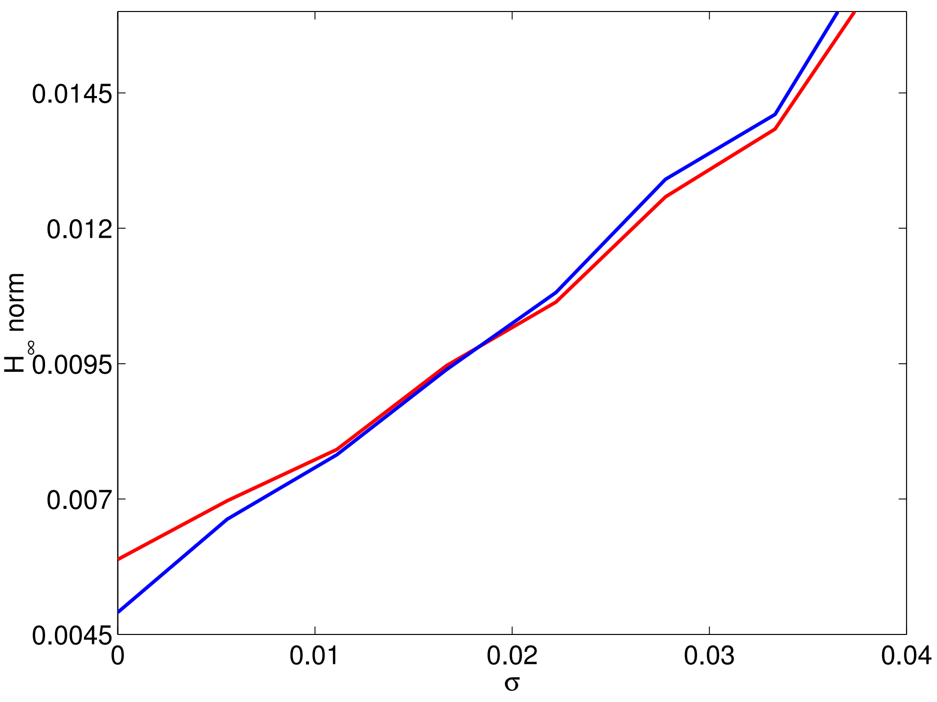


Simulations



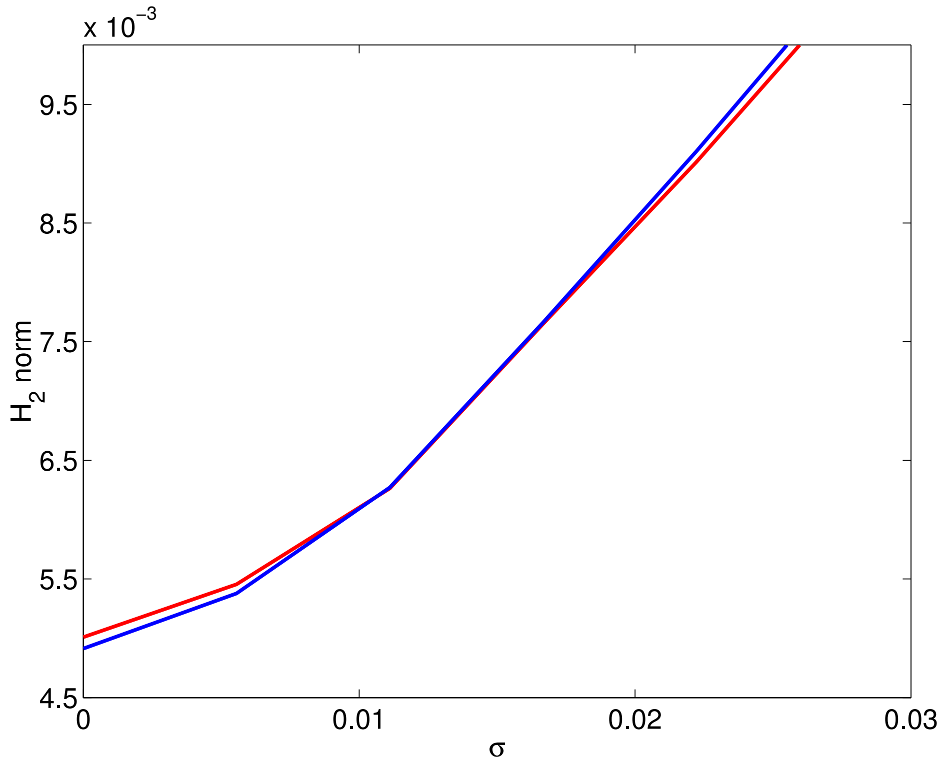
Simulations

reduction to order 2



Simulations

reduction to order 2



Summary

Summary

- From data to balanced state representation:
  sequential zero input response series
  $\longrightarrow$ Hankel matrix
  $\longrightarrow$ SVD
  $\longrightarrow$ balanced state trajectory
  $\longrightarrow$ est. of syst. parameters.

Summary

- **From data to balanced state representation:** sequential zero input response series
  $\longrightarrow$ Hankel matrix
  $\longrightarrow$ SVD
  $\longrightarrow$ balanced state trajectory
  $\longrightarrow$ est. of syst. parameters.

- **Algorithms that pass from $\tilde{u}, \tilde{y}$ *directly* to a state** resp. $\tilde{x}$ and, from there, to (an est. of) $\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$: known for some time. Difficulty:

  arrive *directly* at a *balanced* model.

Summary

- **From data to balanced state representation:** sequential zero input response series
  $\longrightarrow$ Hankel matrix
  $\longrightarrow$ SVD
  $\longrightarrow$ balanced state trajectory
  $\longrightarrow$ est. of syst. parameters.
- **The algorithms may be viewed as part of the research question:**
  Develop algorithms that pass from a given system representation directly to a balanced state representation, or reduction.

Summary

- From data to balanced state representation: sequential zero input response series
  $\longrightarrow$ Hankel matrix
  $\longrightarrow$ SVD
  $\longrightarrow$ balanced state trajectory
  $\longrightarrow$ est. of syst. parameters.
- Under reasonable conditions, every system response can be obtained by solving a linear equation involving the Hankel matrix of the data.

Summary

- From data to balanced state representation:
  sequential zero input response series
  $\longrightarrow$ Hankel matrix
  $\longrightarrow$ SVD
  $\longrightarrow$ balanced state trajectory
  $\longrightarrow$ est. of syst. parameters.
- Under reasonable conditions, every system response can be obtained by solving a linear equation involving the Hankel matrix of the data.
- These insights will be used for setting up effective algorithms for subspace-like identification.

Thank you

Thank you

Thank you

Thank you

Thank you

Thank you

Thank you

Thank you