FROM TIME SERIES to BALANCED REPRESENTATION Part I: Theory



Jan C. Willems (University of Leuven, Belgium) Ivan Markovsky (University of Leuven, Belgium) Paolo Rapisarda (University of Maastricht, NL) Bart L.M. De Moor (University of Leuven, Belgium)

ERNSI meeting, Noorwijkerhout, NL

October 7, 2003

The problem

Given an observed vector time-series

$$egin{bmatrix} ilde{u}(1) \ ilde{y}(1) \end{bmatrix}, egin{bmatrix} ilde{u}(2) \ ilde{y}(2) \end{bmatrix}, \dots, egin{bmatrix} ilde{u}(T) \ ilde{y}(T) \end{bmatrix}$$

find a model for the system that produced these data.

Given an observed vector time-series

$$egin{bmatrix} ilde{u}(1) \ ilde{y}(1) \end{bmatrix}, egin{bmatrix} ilde{u}(2) \ ilde{y}(2) \end{bmatrix}, \dots, egin{bmatrix} ilde{u}(T) \ ilde{y}(T) \end{bmatrix}$$

find a model for the system that produced these data.

In particular, a deterministic linear time-invariant model

in balanced form.

 \exists algorithms from

$$\begin{bmatrix} \tilde{u}(1) \\ \tilde{y}(1) \end{bmatrix}, \dots, \begin{bmatrix} \tilde{u}(t_1) \\ \tilde{y}(t_1) \end{bmatrix}, \dots, \begin{bmatrix} \tilde{u}(t_2) \\ \tilde{y}(t_2) \end{bmatrix}, \dots, \begin{bmatrix} \tilde{u}(T) \\ \tilde{y}(T) \end{bmatrix}$$
$$\downarrow \downarrow \quad \text{to} \quad \downarrow \downarrow$$
$$\tilde{x}(t_1), \quad \dots, \quad \tilde{x}(t_2)$$

 \exists algorithms from

$$\begin{bmatrix} \tilde{u}(1) \\ \tilde{y}(1) \end{bmatrix}, \dots, \begin{bmatrix} \tilde{u}(t_1) \\ \tilde{y}(t_1) \end{bmatrix}, \dots, \begin{bmatrix} \tilde{u}(t_2) \\ \tilde{y}(t_2) \end{bmatrix}, \dots, \begin{bmatrix} \tilde{u}(T) \\ \tilde{y}(T) \end{bmatrix}$$
$$\downarrow \downarrow \quad \text{to } \downarrow \downarrow \\ \tilde{x}(t_1), \dots, \tilde{x}(t_2)$$
$$Solve (LS)$$
$$\begin{bmatrix} \tilde{x}(t_1+1) \cdots \tilde{x}(t_2) \\ y(t_1) \cdots y(t_2-1) \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \tilde{x}(t_1) \cdots \tilde{x}(t_2-1) \\ u(t_1) \cdots u(t_2-1) \end{bmatrix}$$

This yields a state representation.

1)

1)

Solve (LS)

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

This yields a state representation. \rightsquigarrow Reduce the state dimension, by reducing the row dimension of

$$\left[egin{array}{cccc} ilde{x}(t_1) & \ldots & ilde{x}(t_2) \end{array}
ight] .$$

Solve (LS)

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

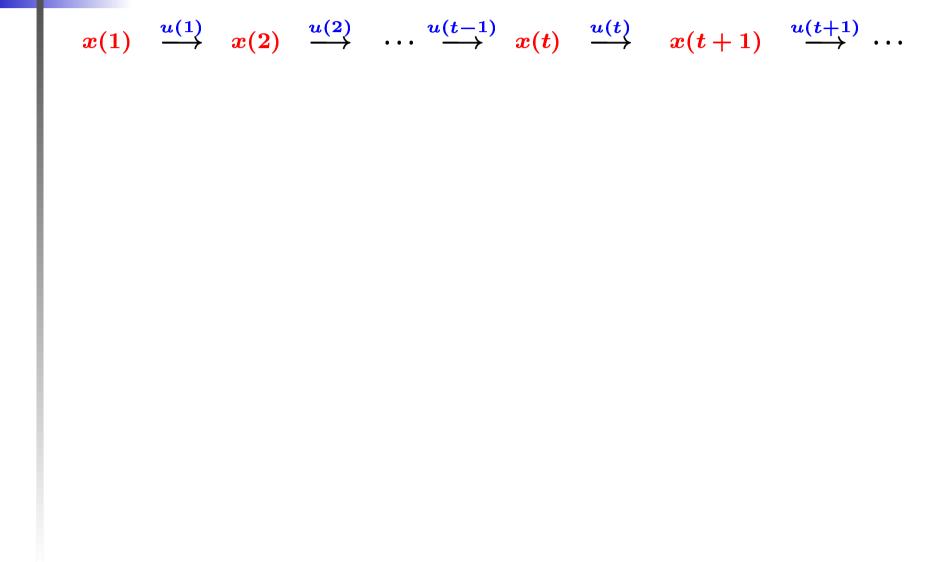
This yields a state representation. \rightsquigarrow Reduce the state dimension, by reducing the row dimension of

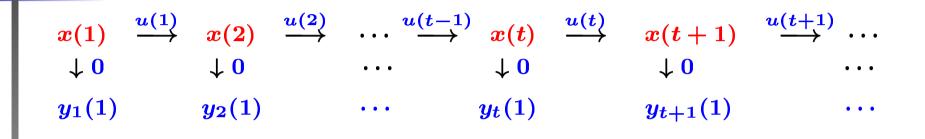
$$ilde{x}(t_1) \quad \ldots \quad ilde{x}(t_2) \ \Big] \cdot$$

This leads to the problem:

Construct $ilde{x}(t_1), \dots, ilde{x}(t_2)$ in a balanced basis.

'Subspace methods' do this.

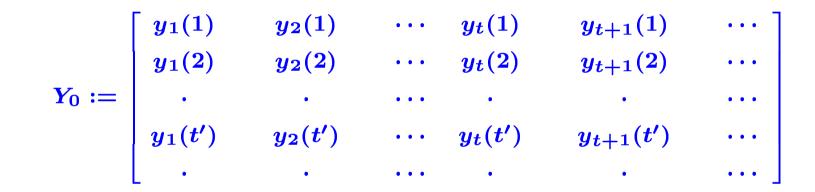




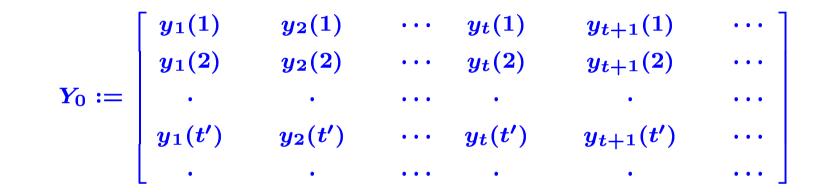
x(1)	$\stackrel{u(1)}{\longrightarrow} x(2)$	$\xrightarrow{u(2)}$	$\stackrel{u(t-1)}{\longrightarrow} x(t)$	$\stackrel{\boldsymbol{u(t)}}{\longrightarrow} x(t+1)$	$\stackrel{u(t+1)}{\rightarrow}$
↓ 0	↓ <mark>0</mark>		↓ <mark>0</mark>	↓ 0	•••
$y_1(1)$	$y_2(1)$	•••	$y_t(1)$	$y_{t+1}(1)$	
↓ <mark>0</mark>	↓ <mark>0</mark>	•••	↓ <mark>0</mark>	↓ <mark>0</mark>	•••
$y_1(2)$	$m{y_2(2)}$	•••	$y_t(2)$	$y_{t+1}(2)$	•••

$x(1) \xrightarrow{u(1)}$	$x(2) \xrightarrow{u(2)}$	$\dots u(t-$	$\stackrel{(1)}{\rightarrow} x(t) \stackrel{u(t)}{\longrightarrow}$	x(t+1)	$\stackrel{u(t+1)}{\longrightarrow} \cdots$
↓ 0	↓ 0	•••	↓ 0	↓ 0	• • •
$y_1(1)$	$y_{2}(1)$	•••	$y_t(1)$	$y_{t+1}(1)$	•••
↓ 0	↓ 0	•••	↓ 0	↓ 0	• • •
$y_1(2)$	$y_2(2)$	•••	$y_t(2)$	$y_{t+1}(2)$	•••
↓ 0	↓ <mark>0</mark>	•••	↓ 0	↓ 0	•••
:	÷	÷		:	÷
↓ <mark>0</mark>	↓ <mark>0</mark>		↓ <mark>0</mark>	↓ <mark>0</mark>	•••
$y_1(t')$	$y_2(t')$	•••	$y_t(t')$	$y_{t+1}(t')$	•••
↓ 0	↓ <mark>0</mark>	•••	↓ 0	↓ <mark>0</mark>	•••
$y_1(t'+1)$	$y_2(t'+1)$	•••	$y_t(t'+1)$	$y_{t+1}(t'+1)$)
↓ 0	↓ <mark>0</mark>	• • •	↓ 0	↓ <mark>0</mark>	•••
÷	÷	÷	÷	÷	÷

Organized into the matrix



Organized into the matrix



Note

$$\begin{aligned} \boldsymbol{x}(t+1) &= A\boldsymbol{x}(t) + B\boldsymbol{u}(t); & \text{for some } \boldsymbol{u}(\cdot) \\ \\ \boldsymbol{Y}_0 &= \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{t'-1} \\ \vdots \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(1) & \boldsymbol{x}(2) & \cdots & \boldsymbol{x}(t) & \cdots \end{bmatrix} \end{aligned}$$

How does deterministic subspace identification work?

There are basically five steps. Use the data

$$egin{bmatrix} ilde{u}(1) \ ilde{y}(1) \end{bmatrix}, egin{bmatrix} ilde{u}(2) \ ilde{y}(2) \end{bmatrix}, \dots, egin{bmatrix} ilde{u}(T) \ ilde{y}(T) \end{bmatrix}$$

to compute (an estimate of)

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

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

 \rightarrow the Hankel matrix

- 2. the impulse response matrix \rightarrow 5
- 3. an SVD of this Hankel matrix $\mathfrak{H} = \mathbf{U} \Sigma \mathbf{V}^{\top}$

- 1. a sequential zero input response series \sim **Y**
- 2. the impulse response matrix \rightarrow 5
- 3. an SVD of this Hankel matrix $\mathfrak{H} = U\Sigma V^{\dagger}$
- 4. the balanced state trajectory

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

- 1. a sequential zero input response series \rightarrow **Y**
- 2. the impulse response matrix \rightarrow 5
- 3. an SVD of this Hankel matrix $\mathfrak{H} = \mathbf{U} \Sigma \mathbf{V}^{\top}$
- 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 model class

Time axis = \mathbb{N} (discrete-time systems) $\sigma = backward shift' \quad \rightsquigarrow \quad (\sigma f)(t) := f(t+1)$ Model class: $\sigma x = Ax + Bu$ y = Cx + Du $w = \begin{bmatrix} u \\ u \end{bmatrix}$ <u>Notation</u>: $\begin{bmatrix} A & B \\ \hline C & D \end{bmatrix}$; impulse response matrix $H: \mathbb{Z}_+ \to \mathbb{R}^{p \times m}; \quad H(0) = D, H(t) = CA^{t-1}B.$

The model class

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

 $R_0w(t)+R_1w(t+1)+\cdots+R_Lw(t+\ell)=0$ $w=egin{bmatrix}u\y\y\end{bmatrix}$

is often to be preferred.

The model class

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

$$R_0w(t)+R_1w(t+1)+\cdots+R_Lw(t+\ell)=0$$
 $w=egin{pmatrix}u\\y\end{pmatrix}$

is often to be preferred. With the polynomial matrix

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

these equations can be written as

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

Call

$$\mathfrak{B} \;=\; \{ oldsymbol{w} : \mathbb{N} o \mathbb{R}^{ imes} \mid oldsymbol{R}(\sigma) oldsymbol{w} = 0 \} \ = \; \ker(oldsymbol{R}(\sigma))$$

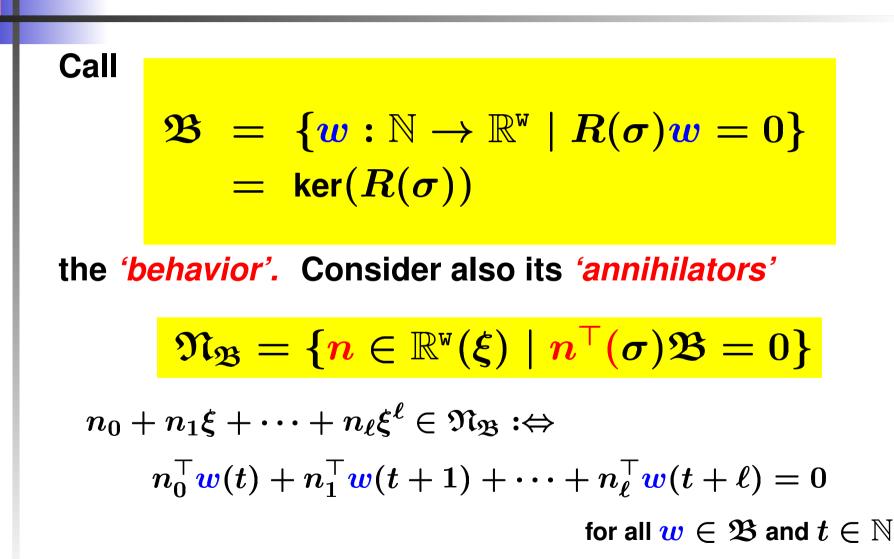
the 'behavior'.

Call

$$\mathfrak{B} = \{ oldsymbol{w} : \mathbb{N}
ightarrow \mathbb{R}^{{\scriptscriptstyle \mathbb{W}}} \mid R(\sigma) oldsymbol{w} = 0 \} \ = \ \mathsf{ker}(R(\sigma))$$

the 'behavior'.

Any subset of $(\mathbb{R}^w)^N$ which is linear, shift-invariant, and closed allows such a representation.



Call

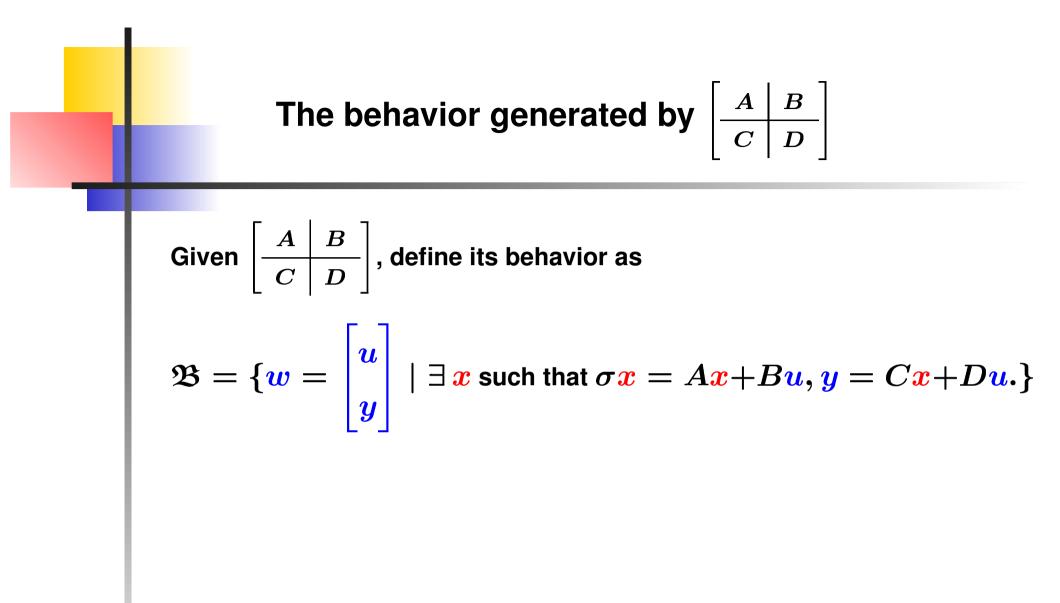
$$\mathfrak{B} = \{ oldsymbol{w} : \mathbb{N}
ightarrow \mathbb{R}^{ imes} \mid oldsymbol{R}(\sigma) oldsymbol{w} = 0 \} \ = \ \mathsf{ker}(oldsymbol{R}(\sigma))$$

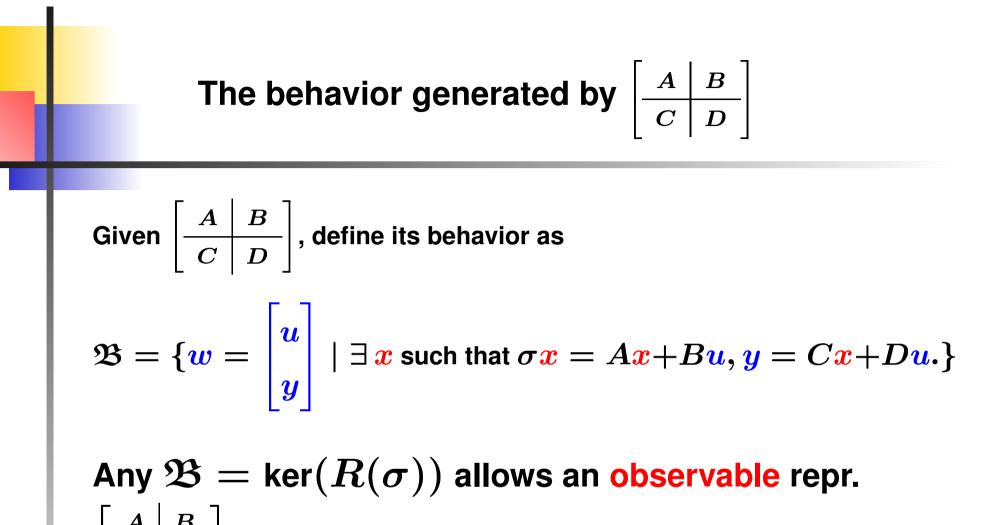
the 'behavior'. Consider also its 'annihilators'

$$\mathfrak{N}_{\mathfrak{B}} = \{ \boldsymbol{n} \in \mathbb{R}^{\scriptscriptstyle \mathrm{W}}(\xi) \mid \boldsymbol{n}^{\top}(\sigma)\mathfrak{B} = 0 \}$$

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.





 $\begin{vmatrix} A & B \\ \hline C & D \end{vmatrix}$. Assumed henceforth.

In behavioral theory

observability \Leftrightarrow **minimality** of the state repr.

Each notion has a version for each representation, $\begin{bmatrix} A & B \\ \hline C & D \end{bmatrix}$, \mathfrak{B} , and $\mathfrak{N}_{\mathfrak{B}}$. We give the most convenient one.

Controllability

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

 $\ell(\mathfrak{B})$:= 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:

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

with = in the single output case.

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

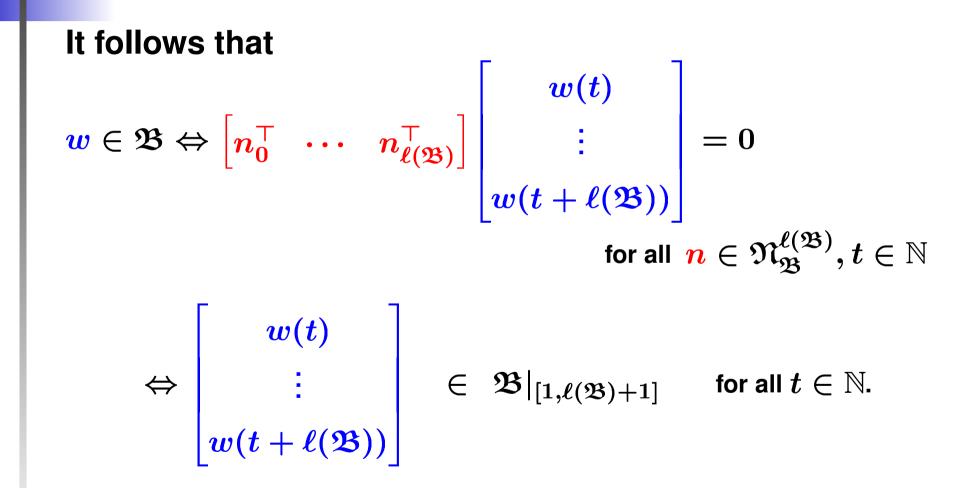
 $\mathfrak{B}|_{[1,\Delta]} \coloneqq \mathsf{the \ behavior \ restr. \ to \ the \ interval} \begin{bmatrix} 1,\Delta \end{bmatrix} = \mathsf{the \ 'legal' \ prefixes \ of \ length \ \Delta}$

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

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

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

 $n_0^{ op} w(t) + n_1^{ op} w(t+1) + \dots + n_d^{ op} w(t+d) = 0$ for all $w \in \mathfrak{B}$ and $t \in \mathbb{N}$.



$$\begin{array}{ll} \mathsf{Hence, if} \quad \Delta > \ell(\mathfrak{B}), \\ w \in \mathfrak{B} \Leftrightarrow \begin{bmatrix} n_0^\top & \cdots & n_{\Delta-1}^\top \end{bmatrix} \begin{bmatrix} w(t) \\ \vdots \\ w(t + \Delta - 1) \end{bmatrix} = 0 \\ \mathsf{for all} \quad n \in \mathfrak{N}_{\mathfrak{B}}^{\Delta - 1}, t \in \mathbb{N} \\ \Leftrightarrow \begin{bmatrix} w(t) \\ \vdots \\ w(t + \Delta - 1) \end{bmatrix} \in \mathfrak{B}|_{[1,\Delta]} \quad \text{for all} \quad t \in \mathbb{N}. \end{array}$$

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

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

Another consequence. Consider

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

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

Another consequence. Assume suffix' = prefix''.

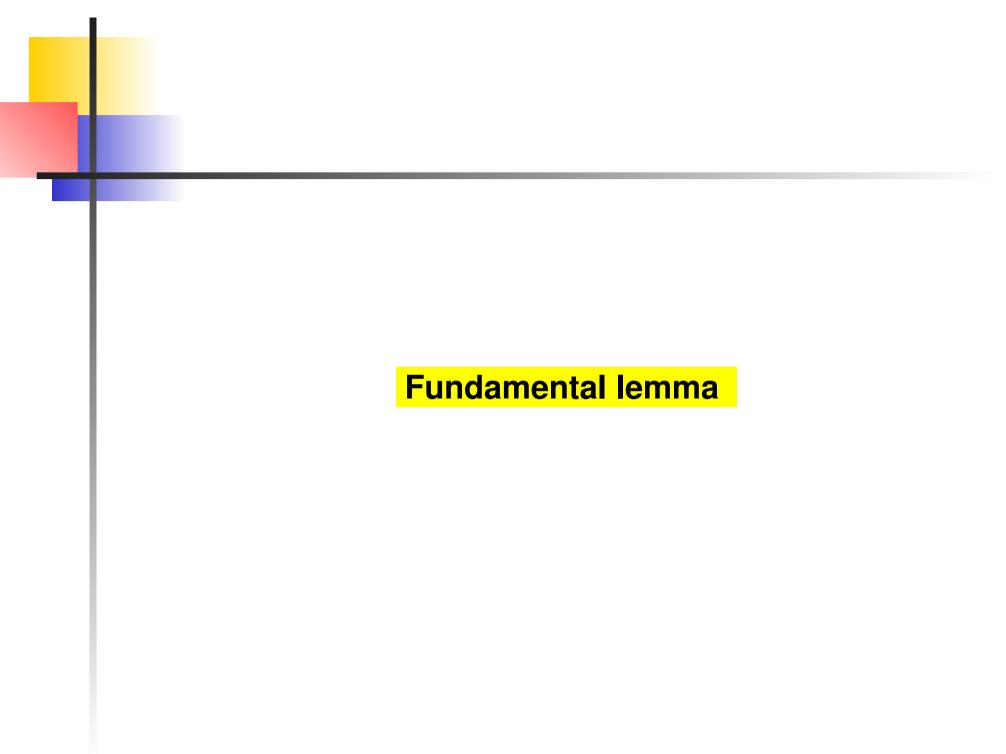
$$\left(\begin{bmatrix} \tilde{u}'(1)\\ \tilde{y}'(1) \end{bmatrix}, \dots, \begin{bmatrix} \tilde{u}'(t-\Delta)\\ \tilde{y}'(t-\Delta) \end{bmatrix}, \begin{bmatrix} \tilde{u}'(t-\Delta+1)\\ \tilde{y}'(t-\Delta+1) \end{bmatrix}, \dots, \begin{bmatrix} \tilde{u}'(t)\\ \tilde{y}'(t) \end{bmatrix} \right)$$

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

Another consequence. Assume suffix' = prefix''. Then their linking

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

belongs to $\mathfrak{B}|_{[1,2t-\Delta]}$, if $\Delta \geq \ell(\mathfrak{B})$, hence if $\Delta \geq n(\mathfrak{B})$.



Key question

Assume that the vector time-series

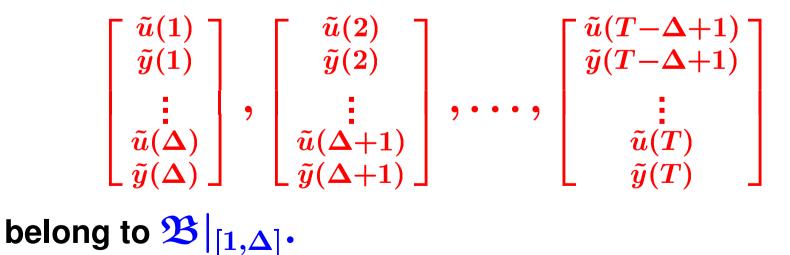
$$\begin{bmatrix} \tilde{u}(1) \\ \tilde{y}(1) \end{bmatrix}, \begin{bmatrix} \tilde{u}(2) \\ \tilde{y}(2) \end{bmatrix}, \dots, \begin{bmatrix} \tilde{u}(T) \\ \tilde{y}(T) \end{bmatrix}$$
has been produced by \mathfrak{B} .

Key question

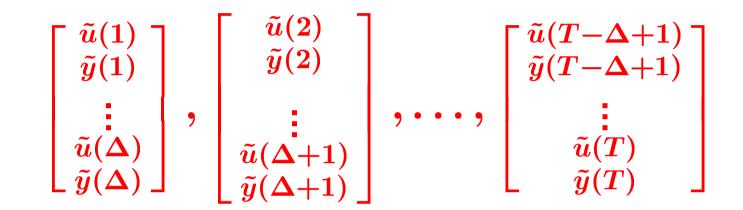
Assume that the vector time-series

$$\begin{bmatrix} \tilde{u}(1) \\ \tilde{y}(1) \end{bmatrix}, \begin{bmatrix} \tilde{u}(2) \\ \tilde{y}(2) \end{bmatrix}, \dots, \begin{bmatrix} \tilde{u}(T) \\ \tilde{y}(T) \end{bmatrix}$$
has been produced by \mathfrak{B} .

Then, of course, the vectors



Key question



Under what conditions on

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

 $ilde{u}(1), ilde{u}(2), \dots ilde{u}(T)$

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

$\left[ilde{u}(1) ight.$	$ ilde{m{u}}(2)$	$ ilde{m{u}}(3)$	•••	$\left\ ilde{u}(T-L+1) ight\ $
$ ilde{u}(2)$	$ ilde{u}(3)$	$ ilde{oldsymbol{u}}(4)$	•••	$ ilde{u}(T-L+2)$
$ ilde{u}(3)$	$ ilde{m{u}}(4)$	$ ilde{u}(5)$	•••	$ ilde{u}(T-L+3)$
1	÷	:	$\gamma_{\rm c}$:
$ ilde{u}(L)$	$ ilde{u}(L+1)$	$ ilde{u}(L+2)$	•••	$ ilde{oldsymbol{u}}(T)$

is of <u>full row rank</u>. Pers. of exc. \Leftrightarrow no linear relations.

Assume that the observed vector time-series

$$egin{bmatrix} ilde{m{u}}(1) \ ilde{m{y}}(1) \end{bmatrix}, egin{bmatrix} ilde{m{u}}(2) \ ilde{m{y}}(2) \end{bmatrix}, \dots, egin{bmatrix} ilde{m{u}}(T) \ ilde{m{y}}(T) \end{bmatrix}$$

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

Assume that the observed vector time-series

$$egin{bmatrix} ilde{m{u}}(1) \ ilde{m{y}}(1) \end{bmatrix}, egin{bmatrix} ilde{m{u}}(2) \ ilde{m{y}}(2) \end{bmatrix}, \dots, egin{bmatrix} ilde{m{u}}(T) \ ilde{m{y}}(T) \end{bmatrix}$$

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

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

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

Assume that the observed vector time-series

 $\Delta + n$

$$egin{bmatrix} ilde{m{u}}(1) \ ilde{m{y}}(1) \end{bmatrix}, egin{bmatrix} ilde{m{u}}(2) \ ilde{m{y}}(2) \end{bmatrix}, \dots, egin{bmatrix} ilde{m{u}}(T) \ ilde{m{y}}(T) \end{bmatrix}$$

has been generated by a **controllable** finite dimensional linear

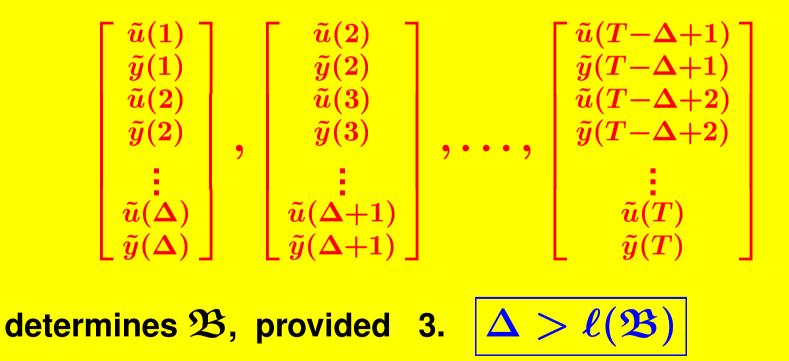
time-invariant system \rightsquigarrow behavior \mathfrak{B} . Then the vectors

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

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

Hence, under the assumptions of

1. controllability and 2. persistency of excitation, the span (& hence left annihilators) of the data vectors



Under reasonable conditions

(contr., Δ suff. large, persistency of excitation), the data matrix

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

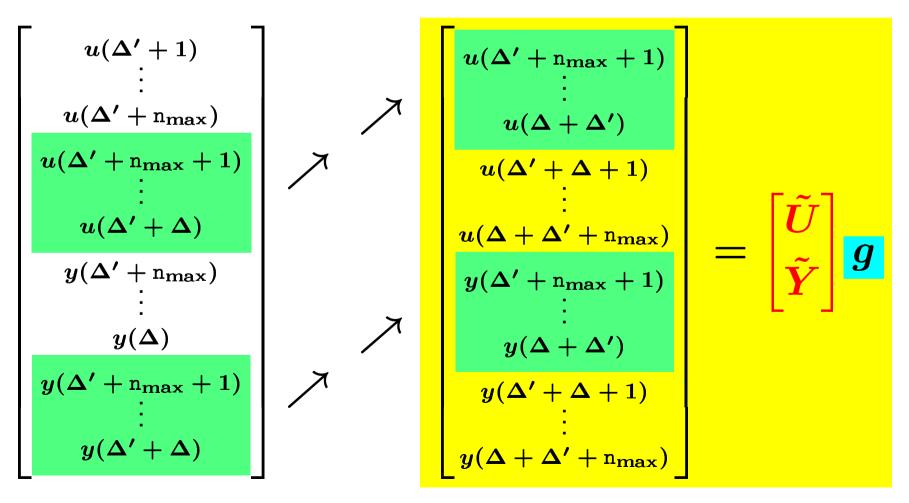
Under reasonable conditions the data matrix has the 'correct' span and the 'correct' left kernel.

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

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

and linking and solving, with $n_{\max} \geq \ell(\mathfrak{B})$ or $n(\mathfrak{B})$, u(1) $u(\Delta - n_{\max} + 1)$ $u(\Delta - n_{\max})$ $u(\Delta)$ $u(\Delta - n_{\max} + 1)$ $u(\Delta + 1)$ $u(\Delta)$ $u(2\Delta - n_{\max})$ \boldsymbol{g} y(1) $y(\Delta - n_{\max} + 1)$ $y(\Delta - n_{\max})$ $y(\Delta)$ $y(\Delta - n_{\max} + 1)$ $y(\Delta + 1)$ $y(\Delta)$ $y(2\Delta - n_{\max})$

and proceeding recursively



This way, an arbitrary long sequence

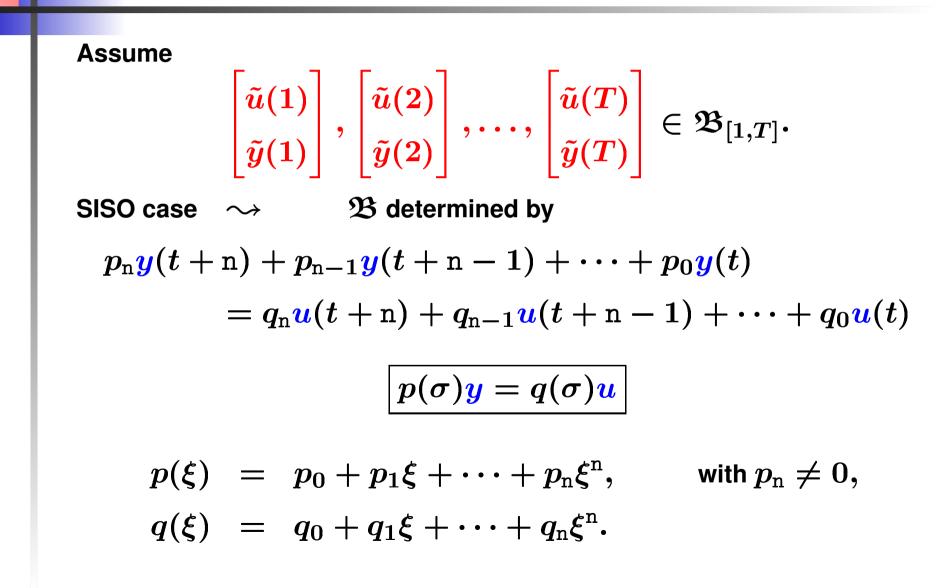
$$(egin{bmatrix} u(1) \\ y(1) \end{bmatrix}, egin{bmatrix} u(2) \\ y(2) \end{bmatrix}, \dots, egin{bmatrix} u(t) \\ y(t) \end{bmatrix}) \in \mathfrak{B}|_{[1,t]}$$

can be obtained.

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

Assume

$$egin{bmatrix} ilde{u}(1) \ ilde{y}(1) \end{bmatrix}, egin{bmatrix} ilde{u}(2) \ ilde{y}(2) \end{bmatrix}, \dots, egin{bmatrix} ilde{u}(T) \ ilde{y}(T) \end{bmatrix} \in \mathfrak{B}_{[1,T]}.$$



Assume

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

$$egin{aligned} p(\xi) &= p_0 + p_1 \xi + \dots + p_n \xi^n, & ext{with } p_n
eq 0, \ q(\xi) &= q_0 + q_1 \xi + \dots + q_n \xi^n. \end{aligned}$$
 $\mathfrak{N}_{\mathfrak{B}} = ext{span} \left\{ egin{bmatrix} -q(\xi) \\ p(\xi) \end{bmatrix}, egin{bmatrix} -\xi q(\xi) \\ \xi p(\xi) \end{bmatrix}, \dots, egin{bmatrix} -\xi^k q(\xi) \\ \xi^k p(\xi) \end{bmatrix}, \dots
ight\}$

Data matrix:

$ ilde{u}(1)$	$ ilde{u}(2)$	•••	$ ilde{u}(T-\Delta+1)$
$ ilde{m{u}}(m{2})$	$ ilde{m{u}}(3)$	•••	$ ilde{u}(T-\Delta+2)$
÷	:	÷	
$ ilde{u}(\Delta)$	$ ilde{u}(\Delta+1)$	•••	$ ilde{m{u}}(T)$
$ ilde{y}(1)$	$ ilde{y}(2)$	•••	$ ilde{y}(T-\Delta+1)$
$ ilde{y}(2)$	$ ilde{m{y}}(m{3})$	•••	$ ilde{y}(T-\Delta+2)$
÷	÷	÷	
$ ilde{m{y}}(m{\Delta})$	$ ilde{y}(\Delta+1)$	•••	$ ilde{oldsymbol{y}}(T)$ _

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

 $\begin{bmatrix} -q_0 & -q_1 & \cdots & -q_n & p_0 & p_1 & \cdots & p_n \end{bmatrix}$

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

$$\begin{bmatrix} -q_0 & \cdots & -q_n & 0 & \cdots & 0 & p_0 & \cdots & p_n & 0 & \cdots & 0 \\ 0 & -q_0 & \cdots & -q_n & \cdots & 0 & 0 & p_0 & \cdots & p_n & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & -q_0 & \cdots & -q_n & 0 & \cdots & 0 & p_0 & \cdots & p_n \end{bmatrix} \xleftarrow{\leftarrow} \operatorname{row} \Delta - \operatorname{n}$$

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

 $\begin{bmatrix} -q_0 & \cdots & -q_n & 0 & \cdots & 0 & p_0 & \cdots & p_n & 0 & \cdots & 0 \\ 0 & -q_0 & \cdots & -q_n & \cdots & 0 & 0 & p_0 & \cdots & p_n & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & -q_0 & \cdots & -q_n & 0 & \cdots & 0 & p_0 & \cdots & p_n \end{bmatrix} \xleftarrow{\leftarrow} \operatorname{row} \Delta - \operatorname{row} \Delta -$

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

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

Extend the data matrix to a larger window:

$\tilde{u}(1)$	$ ilde{m{u}}({m{2}})$	•••	$ ilde{u}(T-\Delta'+1)$
$ ilde{m{u}}(2)$	$ ilde{m{u}}(m{3})$	•••	$ ilde{u}(T-\Delta'+2)$
÷	÷	÷	÷
$ ilde{u}(\Delta')$	$ ilde{u}(\Delta'+1)$	•••	$ ilde{u}(T)$
$ ilde{y}(1)$	$ ilde{y}(2)$	•••	$ ilde{y}(T-\Delta'+1)$
$ ilde{y}(2)$	$ ilde{y}(3)$	•••	$ ilde{y}(T-\Delta'+2)$
÷		÷	
$ ilde{y}(\Delta')$			$ ilde{y}(T)$ _

Then the left kernel contains the rows of

$$\begin{bmatrix} -q_0 & \cdots & -q_n & 0 & \cdots & 0 & p_0 & \cdots & p_n & 0 & \cdots & 0 \\ 0 & -q_0 & \cdots & -q_n & \cdots & 0 & 0 & p_0 & \cdots & p_n & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & -q_0 & \cdots & -q_n & 0 & \cdots & 0 & p_0 & \cdots & p_n \end{bmatrix} \xleftarrow{\leftarrow} \operatorname{row} \Delta' - \operatorname{n}$$

Then the left kernel contains the rows of

Then the left kernel contains the rows of

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 $\Delta' = \Delta + n$.

Then the left kernel contains the rows of

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

$$f(\xi) \left[r(\xi) \mid s(\xi)
ight] = h(\xi) \left[-q(\xi) \mid p(\xi)
ight]$$

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

$f(\xi) \left[r(\xi) \mid s(\xi) ight] = h(\xi) \left[-q(\xi) \mid p(\xi) ight]$

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

$f(\xi) \left[r(\xi) \mid s(\xi) ight] = h(\xi) \left[-q(\xi) \mid p(\xi) ight]$

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

$f(\xi) \left[r(\xi) \mid s(\xi) ight] = h(\xi) \left[-q(\xi) \mid p(\xi) ight]$

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

$$egin{bmatrix} r(\xi) &\mid s(\xi) \end{bmatrix} = h(\xi) egin{bmatrix} -q(\xi) &\mid p(\xi) \end{bmatrix}$$

An idea of the proof

and hence

$$ig| r_0 \ \cdots \ \cdot \ \cdots \ r_{\Delta-1} \ s_0 \ \cdots \ \cdot \ \cdots \ s_{\Delta-1}$$

is in the span of the rows of

$$\begin{bmatrix} -q_0 & \cdots & -q_n & 0 & \cdots & 0 & p_0 & \cdots & p_n & 0 & \cdots & 0 \\ 0 & -q_0 & \cdots & -q_n & \cdots & 0 & 0 & p_0 & \cdots & p_n & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & -q_0 & \cdots & -q_n & 0 & \cdots & 0 & p_0 & \cdots & p_n \end{bmatrix} \xleftarrow{\leftarrow} \operatorname{row} \Delta - n$$

An idea of the proof

and hence $\begin{bmatrix} r_0 & \cdots & \cdots & r_{\Delta-1} & s_0 & \cdots & \cdots & s_{\Delta-1} \end{bmatrix}$ is in the span of the rows of $\begin{bmatrix} -q_0 & \cdots & -q_n & 0 & \cdots & 0 & p_0 & \cdots & p_n & 0 & \cdots & 0 \\ 0 & -q_0 & \cdots & -q_n & \cdots & 0 & 0 & p_0 & \cdots & p_n & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & -q_0 & \cdots & -q_n & 0 & \cdots & 0 & p_0 & \cdots & p_n \end{bmatrix} \xleftarrow{\leftarrow} \operatorname{row} \Delta - n$

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

Van Overschee - De Moor from this perspective

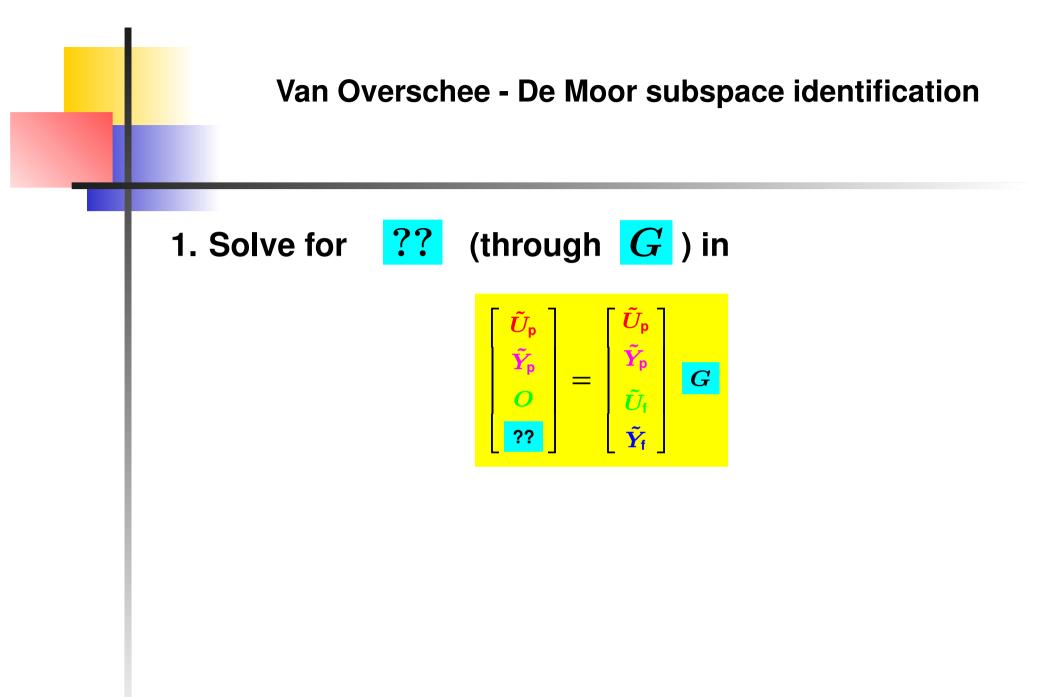


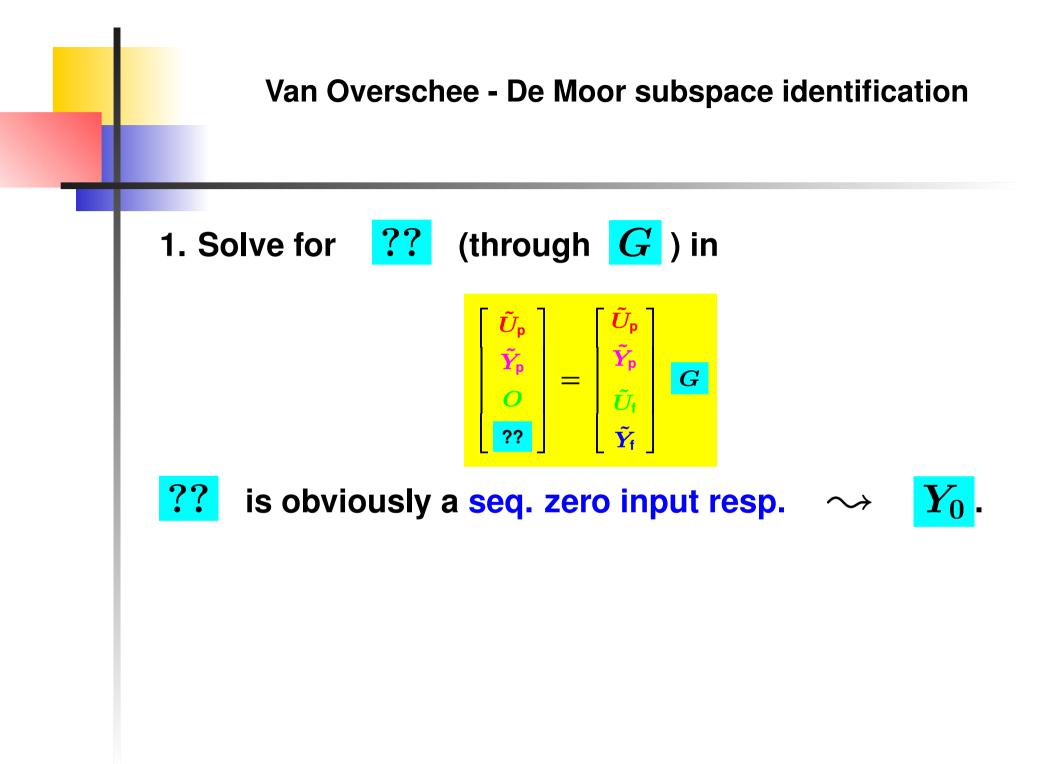
1. How can we compute a sequential zero input response series?

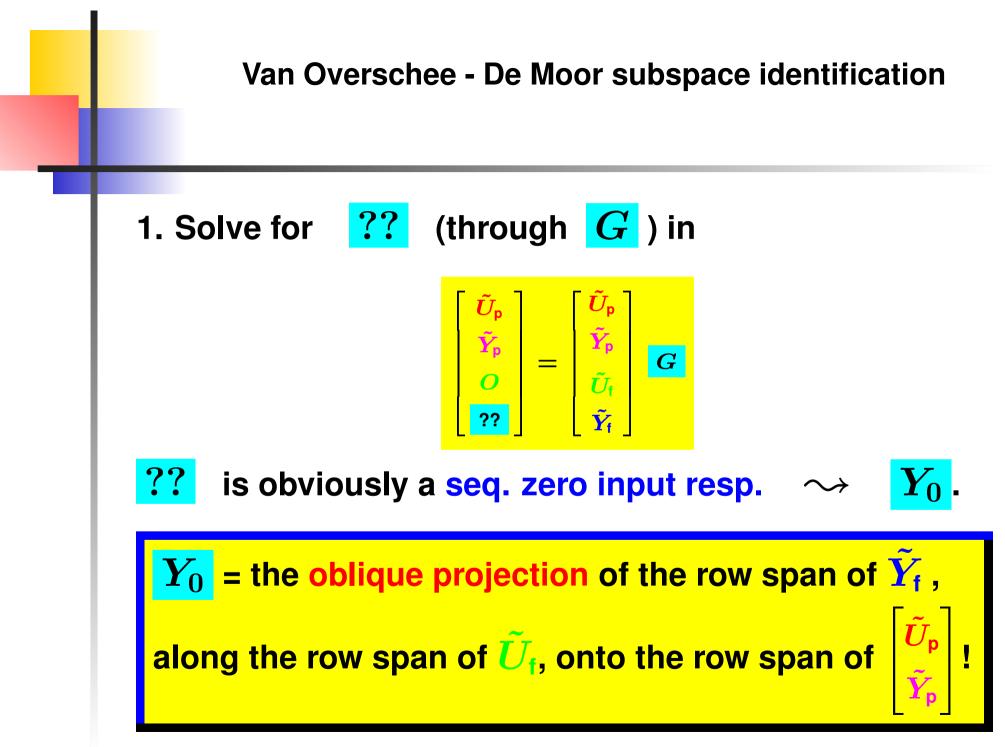
Define the 'past' and 'future' input and output data matrices by

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

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





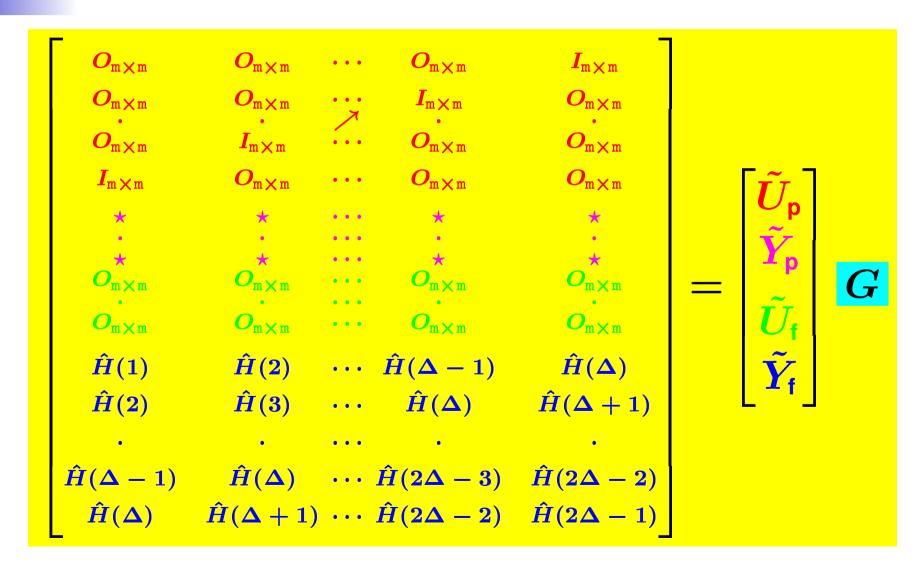


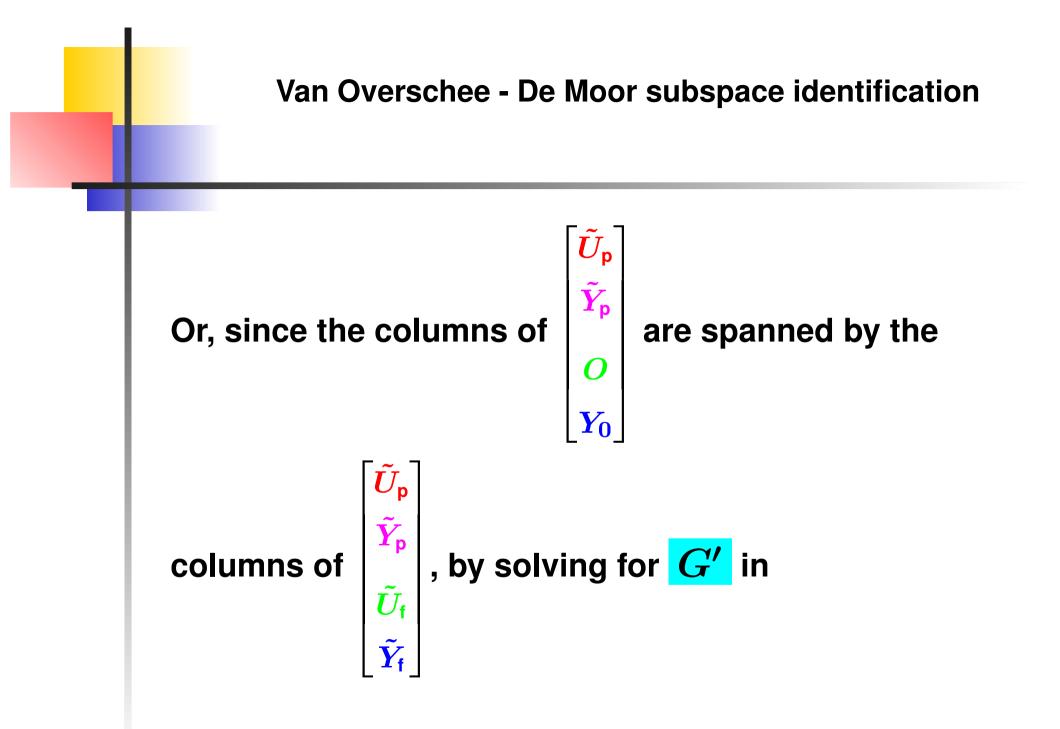


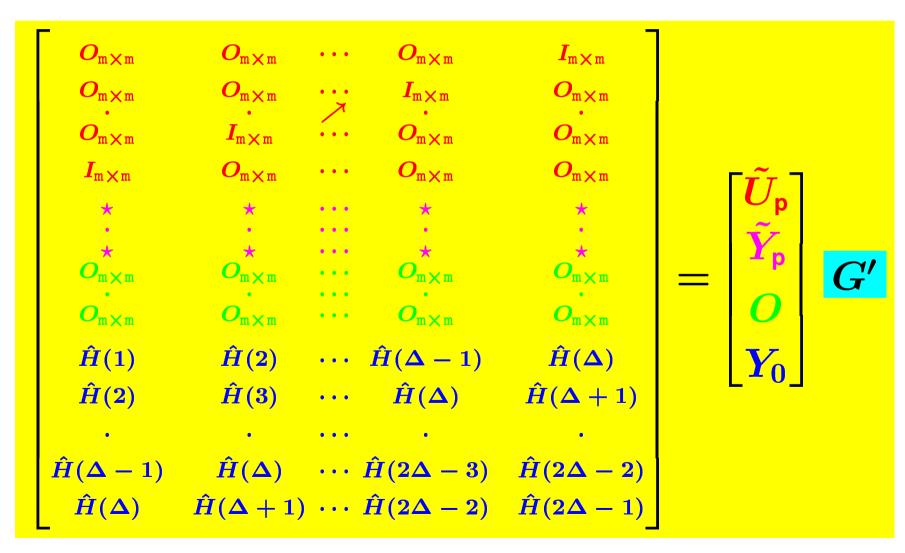
2. How can we compute (an estimate of) the Hankel matrix?

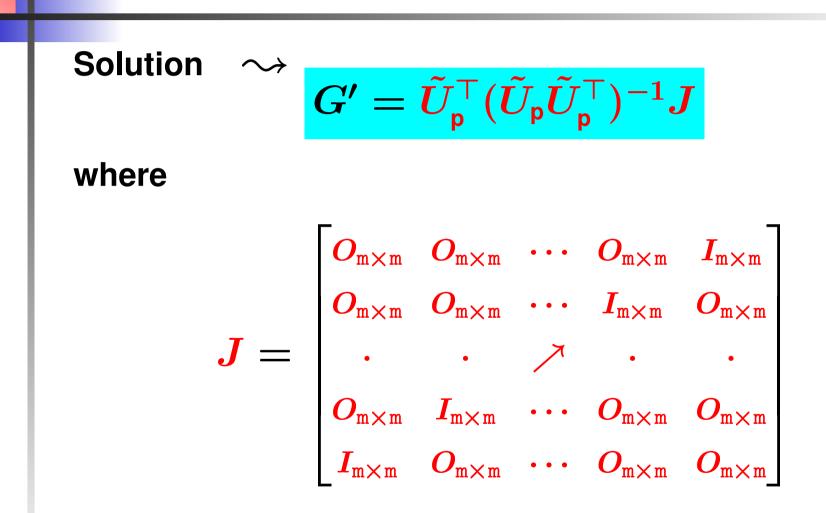
2. How can we compute (an estimate of) the Hankel matrix?

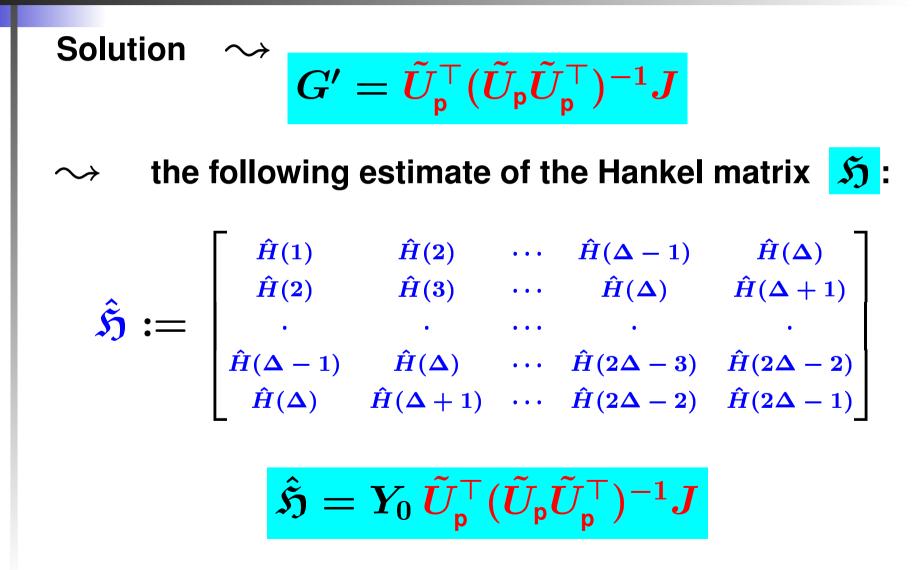
By solving for G in:

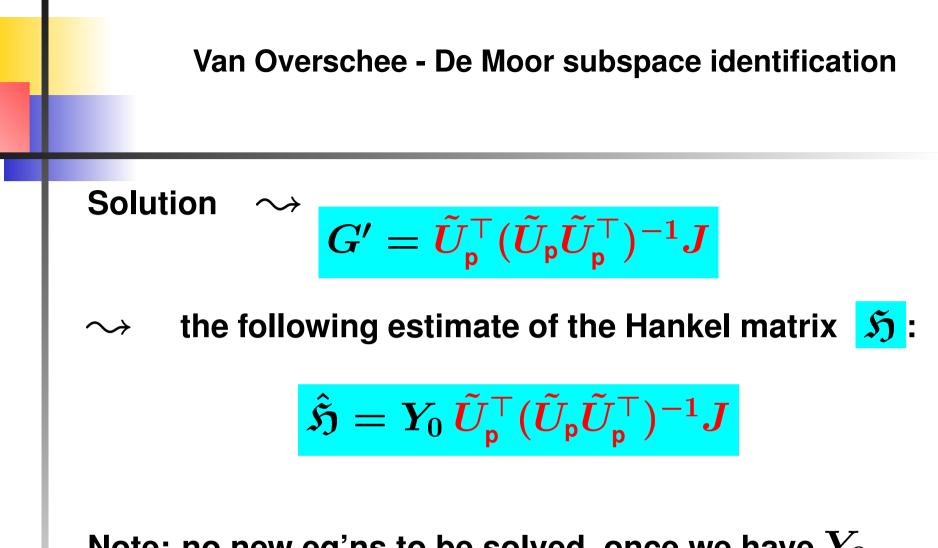




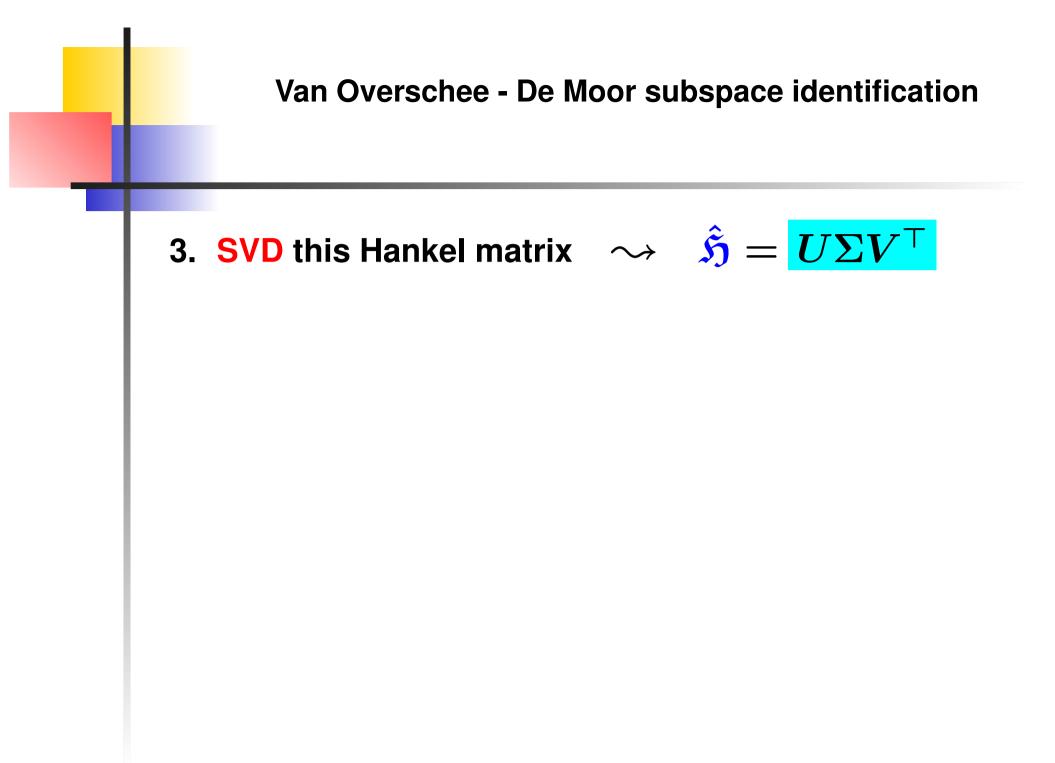


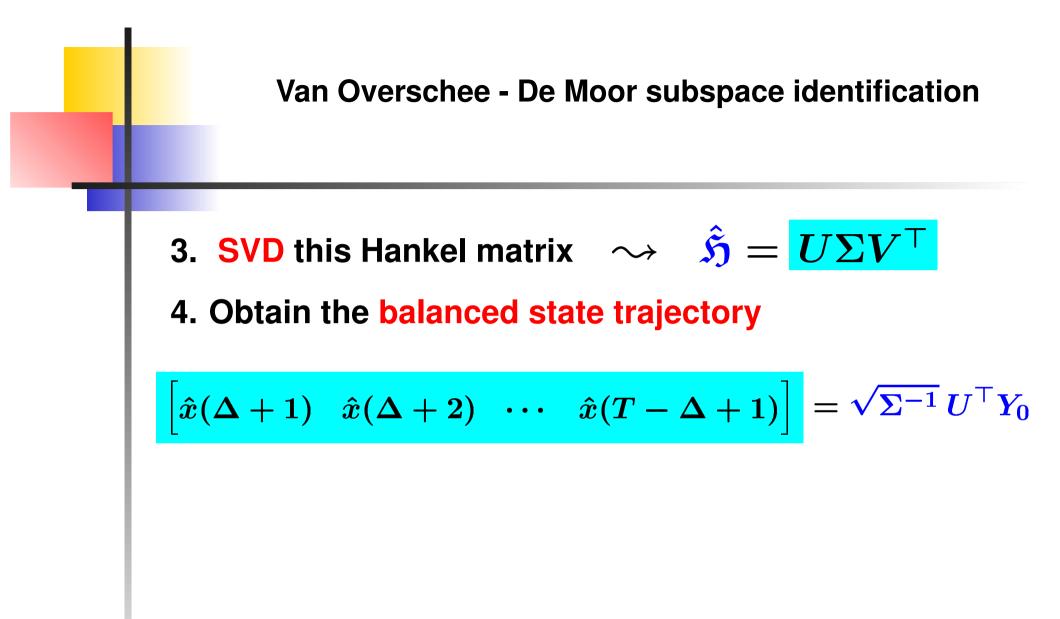






Note: no new eq'ns to be solved, once we have Y_0 .





- 3. SVD this Hankel matrix $\rightsquigarrow \hat{\mathfrak{H}} = \boldsymbol{U} \boldsymbol{\Sigma} \boldsymbol{V}^{\top}$
- 4. Obtain the balanced state trajectory

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

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

		$ ilde{u}(1)$.	$ ilde{u}(2)$.	•••	$ ilde{u}(T-2\Delta+1)$.
~ ¬		$ ilde{u}(\Delta)$	$ ilde{u}(\Delta+1)$	•••	$ ilde{u}(T-\Delta)$
$ ilde{U}_{p}$		$ ilde{m{y}}(1)$	$ ilde{m{y}}(2)$	•••	$ ilde{y}(T-2\Delta+1)$
$ ilde{Y}_{p}$		$egin{array}{c} \cdot \ ilde{y}(\Delta) \end{array}$	$. \ ilde{y}(\Delta+1)$	••••	$\cdot \ ilde{y}(T-\Delta)$
\hat{X}	—	$\hat{x}(\Delta+1)$	$\hat{x}(\Delta+2)$	•••	$\hat{x}(T-\Delta+1)$
$ ilde{U}_{f}$		$ ilde{u}(\Delta+1)$	$ ilde{u}(\Delta+2)$	••••	$ ilde{u}(T-\Delta+1)$
		•	•	•••	•
$ ilde{Y}_{f}$.		$ ilde{m{u}}(2\Delta)$	$ ilde{u}(2\Delta+1)$	•••	$ ilde{oldsymbol{u}}(oldsymbol{T})$
. 7		$ ilde{y}(\Delta+1)$	$ ilde{y}(\Delta+2)$	•••	$ ilde{y}(T-\Delta+1)$
		•		•••	•
		$ ilde{y}(2\Delta)$	$ ilde{y}(2\Delta+1)$	•••	$ ilde{m{y}}(m{T})$

- 3. SVD this Hankel matrix $\rightsquigarrow \hat{\mathfrak{H}} = \boldsymbol{U} \boldsymbol{\Sigma} \boldsymbol{V}^{\top}$
- 4. Obtain the balanced state trajectory

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

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, in Ivan's talk

 ■ From data to balanced state representation: sequential zero input response series

 → Hankel matrix
 → SVD
 → balanced state trajectory
 → est. of syst. parameters.



From data to balanced state representation: sequential zero input response series \rightarrow Hankel matrix \rightarrow SVD \rightarrow balanced state trajectory \rightarrow est. of syst. parameters. Algorithms that pass from \tilde{u}, \tilde{y} directly to a state $\begin{array}{c|c} A & B \\ \hline C & D \end{array}$: resp. $oldsymbol{ ilde{x}}$ and, from there, to (an est. of) known for some time. Difficulty:

arrive *directly* at a *balanced* model.

 ■ From data to balanced state representation: sequential zero input response series

 → Hankel matrix
 → SVD
 → balanced state trajectory
 → 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.

 ■ From data to balanced state representation: sequential zero input response series

 → Hankel matrix
 → SVD
 → balanced state trajectory
 → 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.

 ■ From data to balanced state representation: sequential zero input response series

 → Hankel matrix
 → SVD
 → balanced state trajectory
 → 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.

 From data to balanced state representation: sequential zero input response series

 Hankel matrix
 SVD
 balanced state trajectory
 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.

The combined stochastic/deterministic case from this vantage point is our next target.

