On the occasion of Keith's 60-th

STRUCTURAL ASPECTS OF SYSTEM IDENTIFICATION

by

Keith Glover

B.Sc. (Eng.) Imperial College of Science and Technology,

London University
1967

E.E., M.S. E.E., Massachusetts Institute of Technology
1971

SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

at the

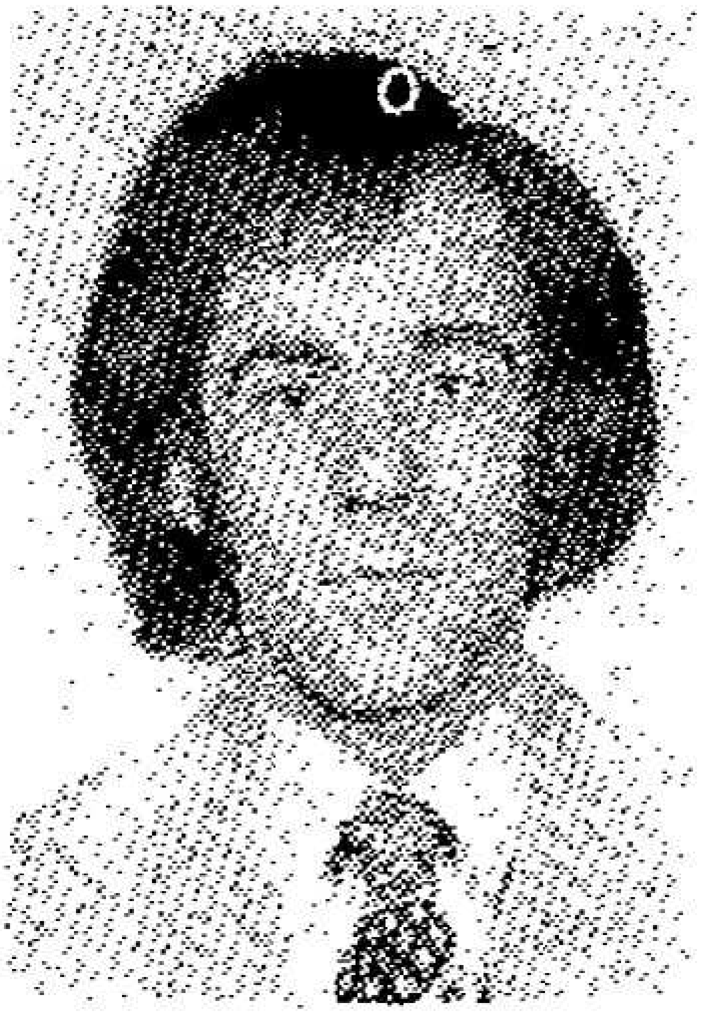
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

August 1973

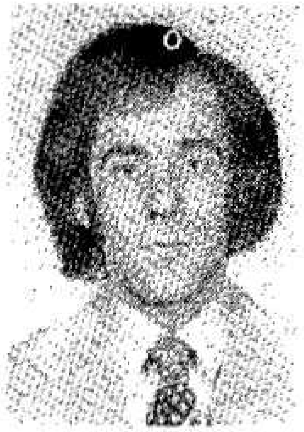
Signature of Author ______________________________
Department of Electrical Engineering

Certified by ______________________________
Thesis Supervisor

Accepted by ______________________________
Chairman, Departmental Committee on Graduate Students



Keith
Bromle
He rece
trical e
of Scie
versity,
S.M., I
enginee
of Tecl
and 197
From



Keith Glover (S'71–M'73) was born in Bromley, Kent, England, on April 23, 1946. He received the B.Sc.(Eng.) degree in electrical engineering from the Imperial College of Science and Technology, London University, London, England, in 1967, and the S.M., E.E., and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, in 1971, 1971, and 1973, respectively.

From 1967 to 1969 he worked on the development of digital communication equipment at the Marconi Company, Chelmsford, Essex, England, and he was a Kennedy Memorial Fellow at M.I.T. from 1969 to 1971. He is currently an Assistant Professor of Electrical Engineering at the University of Southern California, Los Angeles. His present research interests are in system identification and linear system theory.

Jan C. Willems (S'66–M'68) was born in Bruges, Belgium, in 1939. He graduated in electrical and mechanical engineering from the University of Ghent, Belgium, in 1963, received the M.S. degree in electrical engineering from the University of Rhode Island, Kingston, in 1965, and the Ph.D. degree in electrical engiueering from Massachusetts Institute of Technology, Cambridge, in 1968.

From June 1968 he was an Assistant Professor in the Department of Electrical Engineering at M.I.T. until in February 1973 when he was appointed

Parametrizations of Linear Dynamical Systems: Canonical Forms and Identifiability

KEITH GLOVER, MEMBER, IEEE, AND JAN C. WILLEMS, MEMBER, IEEE

Abstract—We consider the problem of what parametrizations of linear dynamical systems are appropriate for identification (i.e., so that the identification problem has a unique solution, and all systems of a particular class can be represented). Canonical forms for controllable linear systems under similarity transformation are considered and it is shown that their use in identification may cause numerical difficulties, and an alternate approach is proposed which avoids these difficulties. Then it is assumed that the system matrices

α. In the context of identifying such dynamical systems the following two properties of a parametrization are desirable.

Property 1: The parametrization should be identifiable in some sense.

Property 2: All systems in an appropriate class can be represented by the parametrization.

STATE FROM DATA

Jan C. Willems
K.U. Leuven, Belgium

Keithfest, Cambridge, UK **April 21, 2006**

Joint work with Ivan Markovsky & Bart De Moor (K.U. Leuven)

The problem

Question

Compute the left kernel of the (block) Hankel matrix

$$\begin{bmatrix} \tilde{w}(1) & \tilde{w}(2) & \cdots & \tilde{w}(t'') & \cdots \\ \tilde{w}(2) & \tilde{w}(3) & \cdots & \tilde{w}(t''+1) & \cdots \\ \tilde{w}(3) & \tilde{w}(4) & \cdots & \tilde{w}(t''+2) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \\ \tilde{w}(t') & \tilde{w}(t'+1) & \cdots & \tilde{w}(t'+t''-1) & \cdots \\ \tilde{w}(t'+1) & \tilde{w}(t'+2) & \cdots & \tilde{w}(t'+t'') & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Background

$\tilde{w} \mapsto$ MPUM

Given the observed (infinite horizon) vector time-series

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

compute the most powerful unfalsified model (MPUM) that generated it.

$\tilde{w} \mapsto$ MPUM

Given the observed (infinite horizon) vector time-series

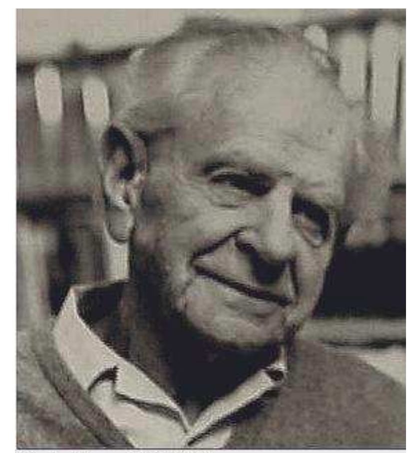
$$\tilde{w} = \tilde{w}(1), \tilde{w}(2), \ldots, \tilde{w}(t), \ldots \qquad \tilde{w}(t) \in \mathbb{R}^{\mathtt{w}}$$

compute the most powerful unfalsified model (MPUM) that generated it.

MPUM : model in a model class,

explains the observations $\tilde{w}$

+ as little else as possible.



Sir Karl Popper (1902-1994)

Karl Popper (1902-1994)

The model class

Exceedingly familiar: The model $\mathfrak{B} \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$ belongs to $\mathfrak{L}^{\mathtt{w}}$:$\Leftrightarrow$

- **$\mathfrak{B}$ is linear, shift-invariant, and closed**
- **$\mathfrak{B}$ is linear, time-invariant, and complete :$\Leftrightarrow$ 'prefix determined'**

The model class

The model $\mathfrak{B} \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$ belongs to $\mathfrak{L}^{\mathtt{w}}$:⇔

- $\mathfrak{B}$ is linear, shift-invariant, and closed
- $\mathfrak{B}$ is linear, time-invariant, and complete :⇔ 'prefix determined'
- $\exists$ matrices $R_0, R_1, \ldots, R_L$ such that $\mathfrak{B}$: all w that satisfy

$$R_0 w(t) + R_1 w(t+1) + \cdots + R_L w(t+L) = 0 \qquad \forall\, t \in \mathbb{N}$$

In the obvious polynomial matrix notation

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

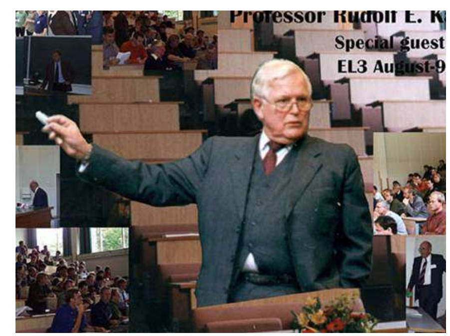
- Including input/output partition

$$P(\sigma) y = Q(\sigma) u, \quad w \cong \begin{bmatrix} u \\ y \end{bmatrix} \quad \det(P) \neq 0$$

The model class

The model $\mathfrak{B} \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$ belongs to $\mathfrak{L}^{\mathtt{w}}$ $:\Leftrightarrow$

- $\mathfrak{B}$ is linear, shift-invariant, and closed
- $\mathfrak{B}$ is linear, time-invariant, and complete $:\Leftrightarrow$ 'prefix determined'
- $R(\sigma)w = 0$
- $P(\sigma)y = Q(\sigma)u, \quad w \cong \begin{bmatrix} u \\ y \end{bmatrix}$



- $\exists$ matrices A, B, C, D such that
  $\mathfrak{B}$ consists of all $w's$ generated by

$$x(t+1) = Ax(t) + Bu(t), \; y(t) = Cx(t) + Du(t), \quad w \cong \begin{bmatrix} u \\ y \end{bmatrix}$$

The model class

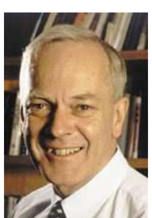
The model $\mathfrak{B} \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$ belongs to $\mathfrak{L}^{\mathtt{w}}$:$\Leftrightarrow$

- $\mathfrak{B}$ is linear, shift-invariant, and closed
- $\mathfrak{B}$ is linear, time-invariant, and complete :$\Leftrightarrow$ 'prefix determined'
- $R(\sigma)w = 0$
- $P(\sigma)y = Q(\sigma)u$, $\quad w \cong \begin{bmatrix} u \\ y \end{bmatrix}$
- $\sigma x = Ax + Bu,\ y = Cx + Du,$ $\quad w \cong \begin{bmatrix} u \\ y \end{bmatrix}$
- $\exists$ a matrix of rational functions G such that $\mathfrak{B}$ $=$ sol'ns of

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

without LOG strictly proper, with LOG proper stable rational.

e.g.










et multi alteri

The problem

Given the observed (infinite horizon) vector time-series

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

compute the MPUM in $\mathfrak{L}^{\mathtt{w}}$ that generated these data.

‘Exact’, ‘deterministic’ system ID (with an eye to approximation).

Subspace ID

$$\tilde{w} \mapsto \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$$

Once we have (an estimate of) the MPUM, the system that produced the data $\tilde{w}$, we can analyze it, make an i/o partition, an observable state representation

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t), \qquad w(t) \cong \begin{bmatrix} u(t) \\ y(t) \end{bmatrix} \end{aligned}$$

and compute the (unique) state trajectory

$$\tilde{x}(1), \tilde{x}(2), \ldots, \tilde{x}(t), \ldots$$

corresponding to

$$\tilde{w}(1), \tilde{w}(2), \ldots, \tilde{w}(t), \ldots$$

$$\tilde{w} \mapsto \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$$

Once we have (an estimate of) the MPUM, the system that produced the data $\tilde{w}$, we can analyze it, make an i/o partition, an observable state representation

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t), \qquad w(t) \cong \begin{bmatrix} u(t) \\ y(t) \end{bmatrix} \end{aligned}$$

and compute the (unique) state trajectory

$$\tilde{x}(1), \tilde{x}(2), \ldots, \tilde{x}(t), \ldots$$

Of course,

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

$$\tilde{w} \mapsto \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$$

Of course,

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

But if we could go the other way:

first compute the state trajectory $\tilde{x}$, directly from $\tilde{w}$,
then this equation provides a way of

identifying the system parameters $\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$

Classical realization is a special case: impulse response data.

$$\tilde{w} \mapsto \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$$

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

Yields an attractive SYSID procedure:

- **Truncation** at suff. large t; copes with **missing data**: cancel columns; extends to more than one observed time series, ...
- **Model reduce** using SVD c.s. by first lowering the row dim. of the matrix $\tilde{X} = \begin{bmatrix} \tilde{x}(1) & \tilde{x}(2) & \cdots & \tilde{x}(t) & \cdots \end{bmatrix}$
- Solve for $\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$ using **Least Squares**

$\rightsquigarrow$ **what has come to be known as 'subspace ID'.**

Algorithms compare favorably compared to PEM, etc.

From data to state

$$\tilde{w} \mapsto \tilde{x}$$

How does this work?

$$\tilde{w}(1), \tilde{w}(2), \ldots, \tilde{w}(t), \ldots$$

$$\Downarrow$$

$$\tilde{x}(1), \tilde{x}(2), \ldots, \tilde{x}(t), \ldots$$

This is a very nice system theoretic question.

$\tilde{w} \mapsto \tilde{x}$

Henceforth, Δ sufficiently large.

Can we somehow identify, directly from the data, the map

$$\boxed{\tilde{w}(1), \tilde{w}(2), \ldots, \tilde{w}(\Delta)} \quad \longmapsto \quad \boxed{\tilde{x}(1)}$$

$$\boxed{\tilde{w}(2), \tilde{w}(3), \ldots, \tilde{w}(\Delta+1)} \quad \longmapsto \quad \boxed{\tilde{x}(2)}$$

$$\vdots \qquad \vdots \qquad \vdots$$

or

$$\boxed{\tilde{w}(1), \tilde{w}(2), \ldots, \tilde{w}(\Delta)} \quad \longmapsto \quad \boxed{\tilde{x}(\Delta+1)}$$

$$\boxed{\tilde{w}(2), \tilde{w}(3), \ldots, \tilde{w}(\Delta+1)} \quad \longmapsto \quad \boxed{\tilde{x}(\Delta+2)}$$

$$\vdots \qquad \vdots \qquad \vdots$$

There are many algorithms. We discuss two.

$\tilde{w} \mapsto \tilde{x}$ by past/future intersection

$$\left[\frac{\mathcal{H}_-}{\mathcal{H}_+}\right] = \left[\begin{array}{ccccc} \tilde{w}(1) & \tilde{w}(2) & \cdots & \tilde{w}(t) & \cdots \\ \tilde{w}(2) & \tilde{w}(3) & \cdots & \tilde{w}(t+1) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \\ \tilde{w}(\Delta) & \tilde{w}(\Delta+1) & \cdots & \tilde{w}(t+\Delta-1) & \cdots \\ \hline \tilde{w}(\Delta+1) & \tilde{w}(\Delta+2) & \cdots & \tilde{w}(t+\Delta) & \cdots \\ \tilde{w}(\Delta+2) & \tilde{w}(\Delta+3) & \cdots & \tilde{w}(t+\Delta+1) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \\ \tilde{w}(2\Delta) & \tilde{w}(2\Delta+1) & \cdots & \tilde{w}(t+2\Delta-1) & \cdots \end{array}\right] \begin{array}{c} \uparrow \\ \uparrow \\ \uparrow \\ \text{'PAST'} \\ \hline \text{'FUTURE'} \\ \downarrow \\ \downarrow \\ \downarrow \end{array}$$

$\tilde{w} \mapsto \tilde{x}$ by past/future intersection

$$\left[\frac{\mathcal{H}_-}{\mathcal{H}_+}\right] = \left[\begin{array}{ccccc} \tilde{w}(1) & \tilde{w}(2) & \cdots & \tilde{w}(t) & \cdots \\ \tilde{w}(2) & \tilde{w}(3) & \cdots & \tilde{w}(t+1) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \\ \tilde{w}(\Delta) & \tilde{w}(\Delta+1) & \cdots & \tilde{w}(t+\Delta-1) & \cdots \\ \hline \tilde{w}(\Delta+1) & \tilde{w}(\Delta+2) & \cdots & \tilde{w}(t+\Delta) & \cdots \\ \tilde{w}(\Delta+2) & \tilde{w}(\Delta+3) & \cdots & \tilde{w}(t+\Delta+1) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \\ \tilde{w}(2\Delta) & \tilde{w}(2\Delta+1) & \cdots & \tilde{w}(t+2\Delta-1) & \cdots \end{array}\right] \begin{array}{c} \uparrow \\ \uparrow \\ \uparrow \\ \text{'PAST'} \\ \hline \text{'FUTURE'} \\ \downarrow \\ \downarrow \\ \downarrow \end{array}$$

The intersection of the span of the rows of $\mathcal{H}_-$ with the span of the rows of $\mathcal{H}_+$ = the state space. The common linear combinations

$$\left[\begin{array}{ccccc} \tilde{x}(\Delta+1) & \tilde{x}(\Delta+2) & \cdots & \tilde{x}(t+\Delta) & \cdots \end{array}\right] \leftarrow \boxed{\text{'PRESENT' STATE}}$$

State = what is common between past and future.
Existing algorithms (N4SID, MOESP,...) use past/future partition.

How do we compute this intersection?

$$\begin{bmatrix} a_1 \\ \hline a_2 \end{bmatrix}^\top \begin{bmatrix} M_1 \\ \hline M_2 \end{bmatrix} \Rightarrow a_1^\top M_1 + a_2^\top M_2 = 0 \rightsquigarrow a_1^\top M_1 = -a_2^\top M_2$$

$$0 = \begin{bmatrix} a_1 \\ \hline a_2 \end{bmatrix}^\top \left[\begin{array}{ccccc} \tilde{w}(1) & \tilde{w}(2) & \cdots & \tilde{w}(t) & \cdots \\ \tilde{w}(2) & \tilde{w}(3) & \cdots & \tilde{w}(t+1) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \\ \tilde{w}(\Delta) & \tilde{w}(\Delta+1) & \cdots & \tilde{w}(t+\Delta-1) & \cdots \\ \hline \tilde{w}(\Delta+1) & \tilde{w}(\Delta+2) & \cdots & \tilde{w}(t+\Delta) & \cdots \\ \tilde{w}(\Delta+2) & \tilde{w}(\Delta+3) & \cdots & \tilde{w}(t+\Delta+1) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \\ \tilde{w}(2\Delta) & \tilde{w}(2\Delta+1) & \cdots & \tilde{w}(t+2\Delta-1) & \cdots \end{array}\right] \begin{array}{c} \uparrow \\ \uparrow \\ \uparrow \\ \text{'PAST'} \\ \hline \text{'FUTURE'} \\ \downarrow \\ \downarrow \\ \downarrow \end{array}$$

How do we compute this intersection?

$$\begin{bmatrix} a_1 \\ \hline a_2 \end{bmatrix}^\top \begin{bmatrix} M_1 \\ \hline M_2 \end{bmatrix} \Rightarrow a_1^\top M_1 + a_2^\top M_2 = 0 \rightsquigarrow a_1^\top M_1 = -a_2^\top M_2$$

$$0 = \begin{bmatrix} a_1 \\ \hline a_2 \end{bmatrix}^\top \left[\begin{array}{ccccc} \tilde{w}(1) & \tilde{w}(2) & \cdots & \tilde{w}(t) & \cdots \\ \tilde{w}(2) & \tilde{w}(3) & \cdots & \tilde{w}(t+1) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \\ \tilde{w}(\Delta) & \tilde{w}(\Delta+1) & \cdots & \tilde{w}(t+\Delta-1) & \cdots \\ \hline \tilde{w}(\Delta+1) & \tilde{w}(\Delta+2) & \cdots & \tilde{w}(t+\Delta) & \cdots \\ \tilde{w}(\Delta+2) & \tilde{w}(\Delta+3) & \cdots & \tilde{w}(t+\Delta+1) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \\ \tilde{w}(2\Delta) & \tilde{w}(2\Delta+1) & \cdots & \tilde{w}(t+2\Delta-1) & \cdots \end{array}\right] \begin{array}{c} \uparrow \\ \uparrow \\ \uparrow \\ \text{'PAST'} \\ \hline \text{'FUTURE'} \\ \downarrow \\ \downarrow \\ \downarrow \end{array}$$

Exploiting Hankel structure $\rightsquigarrow$ **following algorithm**

$\tilde{w} \mapsto \tilde{x}$ via left annihilators

Compute 'the' left annihilators of the Hankel matrix:

$$\begin{bmatrix} N_1 & N_2 & N_3 & \cdots & N_\Delta \end{bmatrix} \begin{bmatrix} \tilde{w}(1) & \tilde{w}(2) & \cdots & \tilde{w}(t) & \cdots \\ \tilde{w}(2) & \tilde{w}(3) & \cdots & \tilde{w}(t+1) & \cdots \\ \tilde{w}(3) & \tilde{w}(4) & \cdots & \tilde{w}(t+2) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \cdots \\ \tilde{w}(\Delta) & \tilde{w}(\Delta+1) & \cdots & \tilde{w}(t+\Delta-1) & \cdots \end{bmatrix} = 0$$

$\tilde{w} \mapsto \tilde{x}$ via left annihilators

Compute ‘the’ left annihilators of the Hankel matrix:

$$\begin{bmatrix} N_1 & N_2 & N_3 & \cdots & N_\Delta \end{bmatrix} \begin{bmatrix} \tilde{w}(1) & \tilde{w}(2) & \cdots & \tilde{w}(t) & \cdots \\ \tilde{w}(2) & \tilde{w}(3) & \cdots & \tilde{w}(t+1) & \cdots \\ \tilde{w}(3) & \tilde{w}(4) & \cdots & \tilde{w}(t+2) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \cdots \\ \tilde{w}(\Delta) & \tilde{w}(\Delta+1) & \cdots & \tilde{w}(t+\Delta-1) & \cdots \end{bmatrix} = 0$$

Then

$$\begin{bmatrix} \tilde{x}(1) & \tilde{x}(2) & \cdots & \tilde{x}(t) & \cdots \end{bmatrix}$$

$$= \begin{bmatrix} N_2 & N_3 & \cdots & N_\Delta & 0 \\ N_3 & N_4 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots\vdots\vdots & \vdots & \vdots \\ N_{\Delta-1} & N_\Delta & \cdots & 0 & 0 \\ N_\Delta & 0 & \cdots & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{w}(1) & \tilde{w}(2) & \cdots & \tilde{w}(t) & \cdots \\ \tilde{w}(2) & \tilde{w}(3) & \cdots & \tilde{w}(t+1) & \\ \tilde{w}(3) & \tilde{w}(4) & \cdots & \tilde{w}(t+2) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \\ \tilde{w}(\Delta) & \tilde{w}(\Delta+1) & \cdots & \tilde{w}(t+\Delta-1) & \cdots \end{bmatrix}$$

$\uparrow \; \uparrow \; \uparrow$

‘shift-and-cut’

$\tilde{w} \mapsto \tilde{x}$ via left annihilators

Compute 'the' left annihilators of the Hankel matrix:

$$\begin{bmatrix} N_1 & N_2 & N_3 & \cdots & N_\Delta \end{bmatrix} \begin{bmatrix} \tilde{w}(1) & \tilde{w}(2) & \cdots & \tilde{w}(t) & \cdots \\ \tilde{w}(2) & \tilde{w}(3) & \cdots & \tilde{w}(t+1) & \cdots \\ \tilde{w}(3) & \tilde{w}(4) & \cdots & \tilde{w}(t+2) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \cdots \\ \tilde{w}(\Delta) & \tilde{w}(\Delta+1) & \cdots & \tilde{w}(t+\Delta-1) & \cdots \end{bmatrix} = 0$$

Then

$$\begin{bmatrix} \tilde{x}(1) & \tilde{x}(2) & \cdots & \tilde{x}(t) & \cdots \end{bmatrix}$$

$$= \begin{bmatrix} N_2 & N_3 & \cdots & N_\Delta & 0 \\ N_3 & N_4 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots\vdots\vdots & \vdots & \vdots \\ N_{\Delta-1} & N_\Delta & \cdots & 0 & 0 \\ N_\Delta & 0 & \cdots & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{w}(1) & \tilde{w}(2) & \cdots & \tilde{w}(t) & \cdots \\ \tilde{w}(2) & \tilde{w}(3) & \cdots & \tilde{w}(t+1) & \\ \tilde{w}(3) & \tilde{w}(4) & \cdots & \tilde{w}(t+2) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \\ \tilde{w}(\Delta) & \tilde{w}(\Delta+1) & \cdots & \tilde{w}(t+\Delta-1) & \cdots \end{bmatrix}$$

↑ ↑ ↑

'shift-and-cut'

a non-minimal state, thou

Back to the beginning

Our problem

Compute the left kernel of the (block) Hankel matrix

$$\begin{bmatrix} \tilde{w}(1) & \tilde{w}(2) & \cdots & \tilde{w}(t'') & \cdots \\ \tilde{w}(2) & \tilde{w}(3) & \cdots & \tilde{w}(t''+1) & \cdots \\ \tilde{w}(3) & \tilde{w}(4) & \cdots & \tilde{w}(t''+2) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \\ \tilde{w}(t') & \tilde{w}(t'+1) & \cdots & \tilde{w}(t'+t''-1) & \cdots \\ \tilde{w}(t'+1) & \tilde{w}(t'+2) & \cdots & \tilde{w}(t'+t'') & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

The module structure

Annihilators as polynomials

Each left annihilator can be identified with a vector polynomial

$$\begin{bmatrix} a_0 & a_1 \cdots a_\Delta & 0 & \cdots \end{bmatrix} \begin{bmatrix} \tilde{w}(1) & \tilde{w}(2) & \cdots & \tilde{w}(t'') & \cdots \\ \tilde{w}(2) & \tilde{w}(3) & \cdots & \tilde{w}(t''+1) & \cdots \\ \tilde{w}(3) & \tilde{w}(4) & \cdots & \tilde{w}(t''+2) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \tilde{w}(t') & \tilde{w}(t'+1) & \cdots & \tilde{w}(t'+t''-1) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} = 0$$

$$\cong \quad a(\xi) = a_0 + a_1\xi + \cdots + a_\Delta \xi^\Delta \in \mathbb{R}[\xi]^{1\times \mathtt{w}} \quad \in \textbf{left kernel}$$

Annihilators as polynomials

Closed under addition

$$\begin{bmatrix} a_0 & \cdots & a_\Delta & 0 & \cdots \end{bmatrix}$$
$$\begin{bmatrix} b_0 & \cdots & b_\Delta & 0 & \cdots \end{bmatrix}$$
$$\Downarrow$$
$$\begin{bmatrix} a_0 + b_0 \cdots a_\Delta + b_\Delta & 0 & \cdots \end{bmatrix} \begin{bmatrix} \tilde{w}(1) & \cdots & \tilde{w}(t'') & \cdots \\ \tilde{w}(2) & \cdots & \tilde{w}(t''+1) & \cdots \\ \tilde{w}(3) & \cdots & \tilde{w}(t''+2) & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{w}(t') & \cdots & \tilde{w}(t'+t''-1) & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} = 0$$

Annihilators as polynomials

and under shifting

$$\begin{array}{c}[a_0 \quad a_1 \quad \cdots \quad a_\Delta \quad 0 \quad 0 \cdots] \\ \Downarrow \\ [\,0 \quad a_0 \quad \cdots \quad a_{\Delta-1} \quad a_\Delta \quad 0 \cdots]\end{array} \begin{bmatrix} \tilde{w}(1) & \cdots & \tilde{w}(t'') & \cdots \\ \tilde{w}(2) & \cdots & \tilde{w}(t''+1) & \cdots \\ \tilde{w}(3) & \cdots & \tilde{w}(t''+2) & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{w}(t') & \cdots & \tilde{w}(t'+t''-1) & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} = 0$$

$a(\xi) = a_0 + a_1\xi + \cdots + a_\Delta \xi^\Delta$ $\in$ **left kernel**

$b(\xi) = b_0 + b_1\xi + \cdots + b_\Delta \xi^\Delta$ $\in$ **left kernel**

$\Rightarrow$ $a(\xi) + b(\xi)$ **and** $\xi a(\xi)$ $\in$ **left kernel.**

Annihilators as polynomials

$a(\xi) = a_0 + a_1\xi + \cdots + a_\Delta \xi^\Delta$ $\in$ **left kernel**

$b(\xi) = b_0 + b_1\xi + \cdots + b_\Delta \xi^\Delta$ $\in$ **left kernel**

$\Rightarrow$ $a(\xi) + b(\xi)$ **and** $\xi a(\xi)$ $\in$ **left kernel.**

$\Rightarrow$ **The left kernel hence forms a $\mathbb{R}[\xi]$-module.**

! Finitely generated: $\exists$ **annihilators** $a(\xi), b(\xi), \cdots, c(\xi)$ **that yield all under** $+$ **and shifts.**

Left kernel is in a sense always finite dimensional (dim. $\mathtt{p} \leq \mathtt{w}$).

The module in subspace ID

State construction via the generators

Generators

$$
\begin{array}{llllll}
[a_0 & a_1 & \cdots & a_{n_1}] & & \\
[b_0 & b_1 & \cdots & \cdots & b_{n_2}] & \\
 & & \vdots & & & \\
[c_0 & c_1 & \cdots & \cdots & \cdots & c_{n_p}]
\end{array}
$$

State construction via the generators

Generators

$$\begin{array}{llllll}
[a_0 & a_1 & \cdots & a_{n_1}] & & \\
[b_0 & b_1 & \cdots & \cdots & b_{n_2}] & \\
 & & \vdots & & & \\
[c_0 & c_1 & \cdots & \cdots & \cdots & c_{n_p}]
\end{array}$$

Then

$$\begin{bmatrix}
a_1 & \cdots & a_{n_1-1} & a_{n_1} & 0 & \cdots \\
a_2 & \cdots & a_{n_1} & 0 & 0 & \cdots \\
\vdots & \vdots\vdots\vdots & \vdots & \vdots & \vdots & \vdots \\
a_{n_1} & 0 & \cdots & 0 & 0 & \cdots \\
\vdots & \vdots\vdots\vdots & \vdots & \vdots & \vdots & \vdots \\
c_1 & \cdots & \cdots & \cdots & c_{n_p-1} & c_{n_p} \\
c_2 & \cdots & \cdots & \cdots & c_{n_p} & 0 \\
\vdots & \vdots\vdots\vdots & \vdots & \vdots & \vdots & \vdots \\
c_{n_p} & 0 & \cdots & \cdots & 0 & 0
\end{bmatrix}
\begin{bmatrix} \tilde{x}(1) & \tilde{x}(2) & \cdots & \tilde{x}(t) & \cdots \end{bmatrix}
=
\begin{bmatrix}
\tilde{w}(1) & \tilde{w}(2) & \cdots & \tilde{w}(t) & \cdots \\
\tilde{w}(2) & \tilde{w}(3) & \cdots & \tilde{w}(t+1) & \cdots \\
\tilde{w}(3) & \tilde{w}(4) & \cdots & \tilde{w}(t+2) & \cdots \\
\vdots & \vdots & \vdots & \vdots & \\
\tilde{w}(c_{n_p}) & \tilde{w}(c_{n_p}+1) & \cdots & \tilde{w}(t+c_{n_p}-1) & \cdots
\end{bmatrix}$$

State construction via the generators

Then

$$\begin{bmatrix} a_1 & \cdots & a_{n_1-1} & a_{n_1} & 0 & \cdots \\ a_2 & \cdots & a_{n_1} & 0 & 0 & \cdots \\ \vdots & \vdots\vdots\vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n_1} & 0 & \cdots & 0 & 0 & \cdots \\ \vdots & \vdots\vdots\vdots & \vdots & \vdots & \vdots & \vdots \\ c_1 & \cdots & \cdots & \cdots & c_{n_p-1} & c_{n_p} \\ c_2 & \cdots & \cdots & \cdots & c_{n_p} & 0 \\ \vdots & \vdots\vdots\vdots & \vdots & \vdots & \vdots & \vdots \\ c_{n_p} & 0 & \cdots & \cdots & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{x}(1) & \tilde{x}(2) & \cdots & \tilde{x}(t) & \cdots \end{bmatrix}$$

$$= \begin{bmatrix} \tilde{w}(1) & \tilde{w}(2) & \cdots & \tilde{w}(t) & \cdots \\ \tilde{w}(2) & \tilde{w}(3) & \cdots & \tilde{w}(t+1) & \cdots \\ \tilde{w}(3) & \tilde{w}(4) & \cdots & \tilde{w}(t+2) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \\ \tilde{w}(c_{n_p}) & \tilde{w}(c_{n_p}+1) & \cdots & \tilde{w}(t+c_{n_p}-1) & \cdots \end{bmatrix}$$

Suitable conditions on generators $\rightsquigarrow$ minimal state.

Computation of the generators

$\tilde{w} \mapsto$ left kernel

Suppose we found a left annihilator of

$$\begin{bmatrix} \tilde{w}(1) & \tilde{w}(2) & \cdots & \tilde{w}(t) & \cdots \\ \tilde{w}(2) & \tilde{w}(3) & \cdots & \tilde{w}(t+1) & \cdots \\ \tilde{w}(3) & \tilde{w}(4) & \cdots & \tilde{w}(t+2) & \cdots \\ \vdots & \vdots & \vdots & & \\ \tilde{w}(\Delta) & \tilde{w}(\Delta+1) & \cdots & \tilde{w}(t+\Delta-1) & \cdots \end{bmatrix}$$

$\tilde{w} \mapsto$ left kernel

Suppose we found a left annihilator of

$$\begin{bmatrix} \tilde{w}(1) & \tilde{w}(2) & \cdots & \tilde{w}(t) & \cdots \\ \tilde{w}(2) & \tilde{w}(3) & \cdots & \tilde{w}(t+1) & \cdots \\ \tilde{w}(3) & \tilde{w}(4) & \cdots & \tilde{w}(t+2) & \cdots \\ \vdots & \vdots & \vdots & & \\ \tilde{w}(\Delta) & \tilde{w}(\Delta+1) & \cdots & \tilde{w}(t+\Delta-1) & \cdots \end{bmatrix}$$

Can we use this to simplify finding the other left annihilators of

$$\begin{bmatrix} \tilde{w}(1) & \tilde{w}(2) & \cdots & \tilde{w}(t) & \cdots \\ \tilde{w}(2) & \tilde{w}(3) & \cdots & \tilde{w}(t+1) & \cdots \\ \tilde{w}(3) & \tilde{w}(4) & \cdots & \tilde{w}(t+2) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \tilde{w}(\Delta) & \tilde{w}(\Delta+1) & \cdots & \tilde{w}(t+\Delta-1) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

The completion lemma

Let $R(\xi) \in \mathbb{R}^{\mathtt{p}\times\mathtt{w}}[\xi]$ be left prime. Then $\exists\ E(\xi) \in \mathbb{R}^{(\mathtt{w}-\mathtt{p})\times\mathtt{w}}[\xi]$ such that

$$\begin{bmatrix} R(\xi) \\ E(\xi) \end{bmatrix} \quad \text{is unimodular}$$

meaning $\det =$ a non-zero constant, invertible as a pol. matrix.

The completion lemma

Let $R(\xi) \in \mathbb{R}^{\mathtt{p} \times \mathtt{w}}[\xi]$ be left prime. Then $\exists$ $E(\xi) \in \mathbb{R}^{(\mathtt{w}-\mathtt{p}) \times \mathtt{w}}[\xi]$ such that

$$\begin{bmatrix} R(\xi) \\ E(\xi) \end{bmatrix} \quad \text{is unimodular}$$

meaning $\det$ = a non-zero constant, invertible as a pol. matrix.

Ex.

$\mathtt{p} = 1, \mathtt{w} = 2, R(\xi) = [r_1(\xi) \;\; r_2(\xi)], E(\xi) = [-y(\xi) \;\; x(\xi)]$

Given $r_1(\xi), r_2(\xi) \in \mathbb{R}[\xi]$, find $x(\xi), y(\xi) \in \mathbb{R}[\xi]$ such that

$$x(\xi)r_1(\xi) + y(\xi)r_2(\xi) = 1 \qquad \textbf{Bézout}$$

Solvable iff r_1, r_2 coprime. $\exists$ algorithms, etc.

The completion lemma

Let $R(\xi) \in \mathbb{R}^{\mathtt{p}\times\mathtt{w}}[\xi]$ be left prime. Then $\exists\ E(\xi) \in \mathbb{R}^{(\mathtt{w}-\mathtt{p})\times\mathtt{w}}[\xi]$ such that

$$\begin{bmatrix} R(\xi) \\ E(\xi) \end{bmatrix} \quad \text{is unimodular}$$

Equivalent proposition:

For a given $\mathfrak{B} \in \mathfrak{L}^{\mathtt{w}}$, there exists $\mathfrak{B}' \in \mathfrak{L}^{\mathtt{w}}$, such that

$$\mathfrak{B} \oplus \mathfrak{B}' = (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$$

iff $\mathfrak{B}$ is ‘controllable’.

Application to left kernel computation

Assume

$$[a_0 \quad a_1 \quad \cdots \quad a_{\mathtt{n}_1}] \begin{bmatrix} \tilde{w}(1) & \tilde{w}(2) & \cdots & \tilde{w}(t) & \cdots \\ \tilde{w}(2) & \tilde{w}(3) & \cdots & \tilde{w}(t+1) & \cdots \\ \tilde{w}(3) & \tilde{w}(4) & \cdots & \tilde{w}(t+2) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \tilde{w}(\mathtt{n}_1+1) & \tilde{w}(\mathtt{n}_1+2) & \cdots & \tilde{w}(t+\mathtt{n}_1) & \cdots \end{bmatrix} = 0$$

Application to left kernel computation

Assume

$$\begin{bmatrix} a_0 & a_1 & \cdots & a_{\mathtt{n}_1} \end{bmatrix} \begin{bmatrix} \tilde{w}(1) & \tilde{w}(2) & \cdots & \tilde{w}(t) & \cdots \\ \tilde{w}(2) & \tilde{w}(3) & \cdots & \tilde{w}(t+1) & \cdots \\ \tilde{w}(3) & \tilde{w}(4) & \cdots & \tilde{w}(t+2) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \tilde{w}(\mathtt{n}_1+1) & \tilde{w}(\mathtt{n}_1+2) & \cdots & \tilde{w}(t+\mathtt{n}_1) & \cdots \end{bmatrix} = 0$$

Complete $a(\xi) \rightsquigarrow E_a(\xi)$ $\begin{bmatrix} a \\ E_a \end{bmatrix}$ **unimodular.**

Application to left kernel computation

Assume

$$\begin{bmatrix} a_0 & a_1 & \cdots & a_{\mathtt{n}_1} \end{bmatrix} \begin{bmatrix} \tilde{w}(1) & \tilde{w}(2) & \cdots & \tilde{w}(t) & \cdots \\ \tilde{w}(2) & \tilde{w}(3) & \cdots & \tilde{w}(t+1) & \cdots \\ \tilde{w}(3) & \tilde{w}(4) & \cdots & \tilde{w}(t+2) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \tilde{w}(\mathtt{n}_1+1) & \tilde{w}(\mathtt{n}_1+2) & \cdots & \tilde{w}(t+\mathtt{n}_1) & \cdots \end{bmatrix} = 0$$

Complete $a(\xi) \rightsquigarrow E_a(\xi)$

Compute the 'error' $\tilde{e} = E_a(\sigma)\tilde{w}$

Note that $\tilde{e}$ **is** $(\mathtt{w}-1)$**-dimensional.**

Application to left kernel computation

Assume

$$[a_0 \quad a_1 \quad \cdots \quad a_{n_1}] \begin{bmatrix} \tilde{w}(1) & \tilde{w}(2) & \cdots & \tilde{w}(t) & \cdots \\ \tilde{w}(2) & \tilde{w}(3) & \cdots & \tilde{w}(t+1) & \cdots \\ \tilde{w}(3) & \tilde{w}(4) & \cdots & \tilde{w}(t+2) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \tilde{w}(n_1+1) & \tilde{w}(n_1+2) & \cdots & \tilde{w}(t+n_1) & \cdots \end{bmatrix} = 0$$

Compute

$$[b_0 \quad b_1 \quad \cdots \quad b_{n_2}] \begin{bmatrix} \tilde{e}(1) & \tilde{e}(2) & \cdots & \tilde{e}(t) & \cdots \\ \tilde{e}(2) & \tilde{e}(3) & \cdots & \tilde{e}(t+1) & \cdots \\ \tilde{e}(3) & \tilde{e}(4) & \cdots & \tilde{e}(t+2) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \tilde{e}(n_2+1) & \tilde{e}(n_2+2) & \cdots & \tilde{e}(t+n_2) & \cdots \end{bmatrix} = 0$$

Yields annihilator $b(\xi)E_a(\xi)$ $\rightsquigarrow$ **2 generators:** $a(\xi), b(\xi)E_a(\xi)$

Complete b $\rightsquigarrow$ E_b. **Compute** $\tilde{e}' = E_b(\sigma)\tilde{e}$, **proceed recursively...**

Application to left kernel computation

Assume

$$[a_0 \quad a_1 \quad \cdots \quad a_{n_1}] \begin{bmatrix} \tilde{w}(1) & \tilde{w}(2) & \cdots & \tilde{w}(t) & \cdots \\ \tilde{w}(2) & \tilde{w}(3) & \cdots & \tilde{w}(t+1) & \cdots \\ \tilde{w}(3) & \tilde{w}(4) & \cdots & \tilde{w}(t+2) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \tilde{w}(n_1+1) & \tilde{w}(n_1+2) & \cdots & \tilde{w}(t+n_1) & \cdots \end{bmatrix} = 0$$

Compute

$$[b_0 \quad b_1 \quad \cdots \quad b_{n_2}] \begin{bmatrix} \tilde{e}(1) & \tilde{e}(2) & \cdots & \tilde{e}(t) & \cdots \\ \tilde{e}(2) & \tilde{e}(3) & \cdots & \tilde{e}(t+1) & \cdots \\ \tilde{e}(3) & \tilde{e}(4) & \cdots & \tilde{e}(t+2) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \tilde{e}(n_2+1) & \tilde{e}(n_2+2) & \cdots & \tilde{e}(t+n_2) & \cdots \end{bmatrix} = 0$$

Recursively $a(\xi), \quad b(\xi)E_a(\xi), \quad \cdots, \quad c(\xi)\cdots E_b(\xi)E_a(\xi)$

yields left kernel by computing p times a left kernel vector.

Recursion can be combined with the state computation.

Summary

Summary

Subspace ID:

$$\tilde{w}(1), \tilde{w}(2), \ldots, \tilde{w}(t), \ldots$$

$$\downarrow$$

$$\tilde{X} = [\tilde{x}(1), \tilde{x}(2), \ldots, \tilde{x}(t), \ldots]$$

$$\downarrow$$

Row reduce $\tilde{X}$

$$\downarrow$$

LS solve

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

$$\downarrow$$

Model $\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$

Summary

State from data:

$$\tilde{w}(1), \tilde{w}(2), \ldots, \tilde{w}(t), \ldots$$
$$\downarrow$$
$$\tilde{x}(1), \tilde{x}(2), \ldots, \tilde{x}(t), \ldots$$

via left kernel of the data Hankel matrix

$$\begin{bmatrix} \tilde{w}(1) & \tilde{w}(2) & \cdots & \tilde{w}(t'') & \cdots \\ \tilde{w}(2) & \tilde{w}(3) & \cdots & \tilde{w}(t''+1) & \cdots \\ \tilde{w}(3) & \tilde{w}(4) & \cdots & \tilde{w}(t''+2) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \\ \tilde{w}(t') & \tilde{w}(t'+1) & \cdots & \tilde{w}(t'+t''-1) & \cdots \\ \tilde{w}(t'+1) & \tilde{w}(t'+2) & \cdots & \tilde{w}(t'+t'') & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

This is a module of dimension $\leq$ `w`

its generators lead to the state via shift-and-cut

Summary

$$\begin{bmatrix} a_1 & \cdots & a_{n_1-1} & a_{n_1} & 0 & \cdots \\ a_2 & \cdots & a_{n_1} & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n_1} & 0 & \cdots & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ c_1 & \cdots & \cdots & \cdots & c_{n_p-1} & c_{n_p} \\ c_2 & \cdots & \cdots & \cdots & c_{n_p} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ c_{n_p} & 0 & \cdots & \cdots & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} \tilde{x}(1) & \tilde{x}(2) & \cdots & \tilde{x}(t) & \cdots \end{bmatrix}$$

$$=$$

$$\begin{bmatrix} \tilde{w}(1) & \tilde{w}(2) & \cdots & \tilde{w}(t) & \cdots \\ \tilde{w}(2) & \tilde{w}(3) & \cdots & \tilde{w}(t+1) & \cdots \\ \tilde{w}(3) & \tilde{w}(4) & \cdots & \tilde{w}(t+2) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \\ \tilde{w}(c_{n_p}) & \tilde{w}(c_{n_p}+1) & \cdots & \tilde{w}(t+c_{n_p}-1) & \cdots \end{bmatrix}$$

Summary

This left kernel can be computed recursively by repeated use of the completion lemma and error propagation.

Requires computing $\mathtt{p}$ vectors in kernel of (truncated) Hankel matrices formed by 'error'.
This error time-series decreases each time in dimension.

Can be executed using numerical LA.
Very adapted to approximate computations.

Details & copies of the lecture frames are available from/at

`Jan.Willems@esat.kuleuven.be`

`http://www.esat.kuleuven.be/~jwillems`

Thank you

Thank you

Thank you

Thank you

Thank you

Thank you

Thank you

Thank you

Happy Birthday !!!