RECURSIVE COMPUTATION

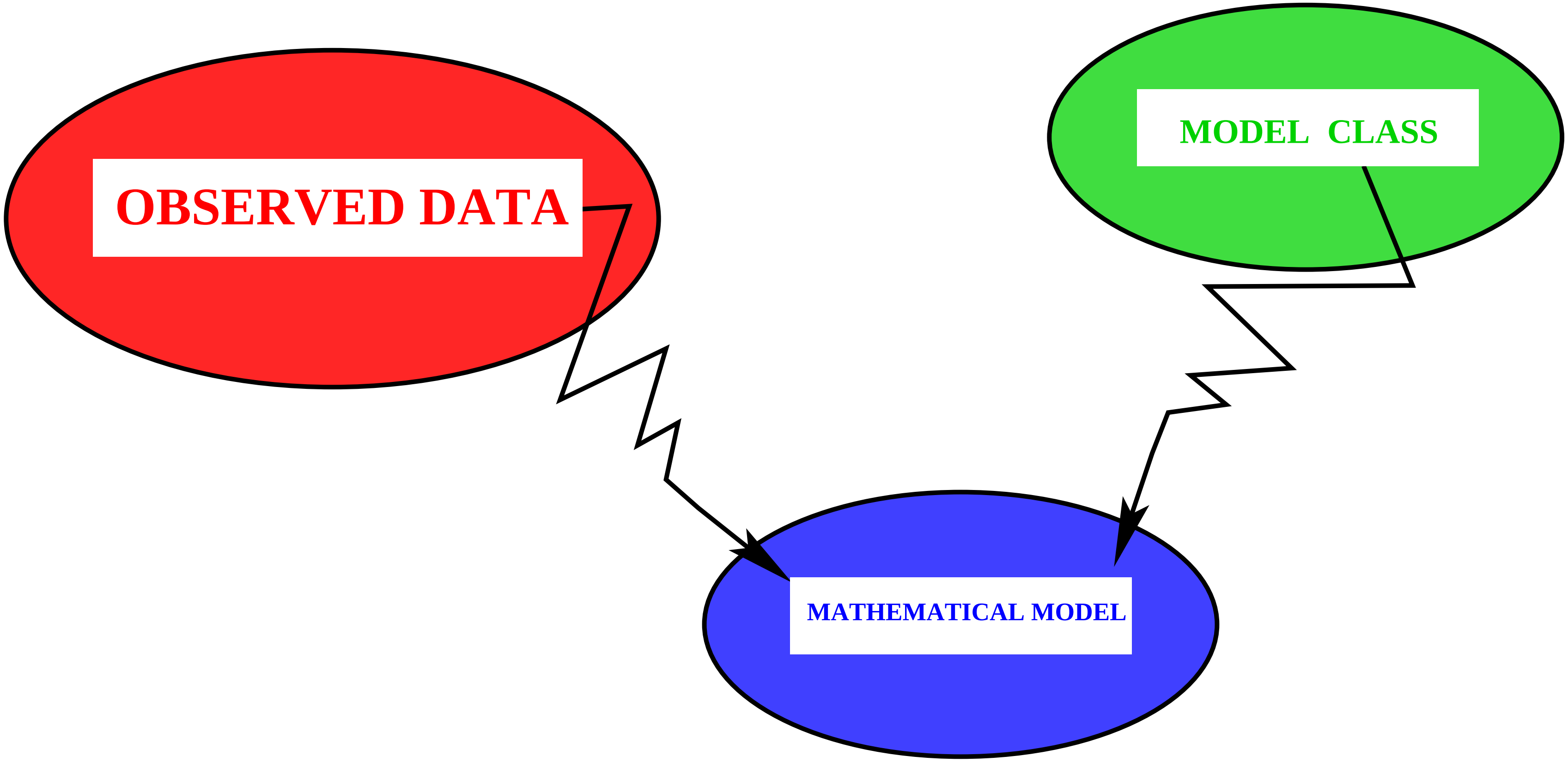
OF THE MPUM

Jan C. Willems
K.U. Leuven, Flanders, Belgium

Workshop on Uncertain Dynamical Systems, Awaji, Japan **July 4, 2008**

System identification

System ID



Case of interest today

Observed data: **a vector time-series**

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

Case of interest today

Observed data: **a vector time-series**

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

Model class: **linear time-invariant systems**

$$\boxed{R_0 w(t) + R_1 w(t+1) + \cdots + R_L w(t+L) = 0}$$

with $R_0, R_1, \cdots, R_L \in \mathbb{R}^{\bullet \times \mathtt{w}}$

Case of interest today

Model class: **linear time-invariant systems**

$$R_0 w(t) + R_1 w(t+1) + \cdots + R_L w(t+L) = 0$$

Usually the model class considered is

$$P_0 y(t) + P_1 y(t+1) + \cdots + Q_0 u(t) + Q_1 u(t+1) + \cdots$$
$$= M_0 \varepsilon(t) + M_1 \varepsilon(t+1) + \cdots + M_L \varepsilon(t+L)$$

with an input/output partition, $w = \begin{bmatrix} u \\ y \end{bmatrix}$

Case of interest today

Usually the model class considered is

$$P_0 y(t) + P_1 y(t+1) + \cdots + Q_0 u(t) + Q_1 u(t+1) + \cdots$$
$$= M_0 \varepsilon(t) + M_1 \varepsilon(t+1) + \cdots + M_L \varepsilon(t+L)$$

and the ε's: random variables
to account for unobserved inputs,
measurement noise, modeling errors, etc.

Provides SYSID algorithms with a 'certificate'.

Case of interest today

Model class used today

$$R_0 w(t) + R_1 w(t+1) + \cdots + R_L w(t+L)$$
$$= M_0 a(t) + M_1 a(t+1) + \cdots + M_L a(t+L)$$

with algorithms amenable to approximate modeling.

Case of interest today

Observed data: **a vector time-series**

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

We consider the simple case:

1. $T = \infty$
2. **exact, deterministic, modeling**
   (with an eye towards approximations)
3. **model class: linear time-invariant systems**

The model class: $\mathscr{L}^{\mathtt{w}}$

Linear time-invariant dynamical systems

described by difference equations

The model class

A (deterministic) dynamical system is a subset

$$\mathscr{B} \subseteq \left(\mathbb{R}^{\mathtt{w}}\right)^{\mathbb{N}}$$

The family of time-series $\mathscr{B} \subseteq \left(\mathbb{R}^{\mathtt{w}}\right)^{\mathbb{N}}$ is called the behavior of the model

The model class

$\mathscr{B} \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$ **belongs to the model class** $\mathscr{L}^{\mathtt{w}}$ $:\Leftrightarrow$

- $\mathscr{B}$ **is linear, shift-invariant, and closed**

The model class

$\mathscr{B} \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$ **belongs to the model class** $\mathscr{L}^{\mathtt{w}}$ $:\Leftrightarrow$

- $\mathscr{B}$ **is linear, shift-invariant, and closed**
- $\exists$ **matrices** $R_0, R_1, \ldots, R_L \in \mathbb{R}^{\bullet \times \mathtt{w}}$ **such that**

$$\mathscr{B} = \text{ all } w : \mathbb{N} \to \mathbb{R}^{\mathtt{w}} \text{ that satisfy}$$

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

i.e.,

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

in the obvious polynomial matrix notation with

$$R(\xi) := R_0 + R_1\xi + \cdots + R_L\xi^L \quad \text{and} \quad \sigma := \text{ left shift}$$

The model class

$\mathscr{B} \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$ **belongs to the model class** $\mathscr{L}^{\mathtt{w}}$ $:\Leftrightarrow$

- $\mathscr{B}$ **is linear, shift-invariant, and closed**
- $R(\sigma)w = 0$
- $\exists$ **matrices** A, B, C, D **such that**
  $\mathscr{B}$ **consists of all** $w's$ **generated by**

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

Rudolf Kalman

The model class

$\mathscr{B} \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$ **belongs to the model class** $\mathscr{L}^{\mathtt{w}}$ $:\Leftrightarrow$

- $\mathscr{B}$ **is linear, shift-invariant, and closed**
- $R(\sigma)w = 0$
- $x(t+1) = Ax(t) + Bu(t),\ y(t) = Cx(t) + Du(t)$ $\quad w \cong \begin{bmatrix} u \\ y \end{bmatrix}$
- $\cdots$ **very many relevant equivalent representations**

The MPUM

The MPUM

Given the observed (infinite-horizon) time-series

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

The MPUM

Given the observed (infinite-horizon) time-series

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

Call the model $\mathscr{B} \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$ **unfalsified** $:\Leftrightarrow \tilde{w} \in \mathscr{B}$

The MPUM

Given the observed (infinite-horizon) time-series

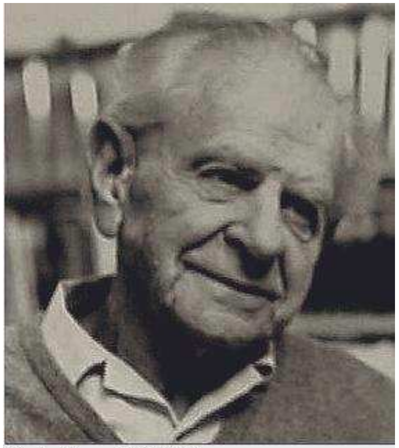
$$\tilde{w} = (\tilde{w}(1), \tilde{w}(2), \ldots, \tilde{w}(t), \ldots) \qquad \tilde{w}(t) \in \mathbb{R}^{\mathtt{w}}$$

Call the model $\mathscr{B} \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$ **unfalsified** $:\Leftrightarrow \tilde{w} \in \mathscr{B}$

Call the model $\mathscr{B}_1 \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$ **more powerful than** $\mathscr{B}_2 \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$

$$:\Leftrightarrow \mathscr{B}_1 \subset \mathscr{B}_2$$

The more a model forbids, the better it is!


Sir Karl Popper (1902-1994)

Karl Popper

The MPUM

Call the model $\mathscr{B} \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$ **unfalsified** $:\Leftrightarrow \tilde{w} \in \mathscr{B}$

Call the model $\mathscr{B}_1 \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$ **more powerful than** $\mathscr{B}_2 \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$

$:\Leftrightarrow \mathscr{B}_1 \subset \mathscr{B}_2$

The most powerful unfalsified model (MPUM) in $\mathscr{L}^{\mathtt{w}}$

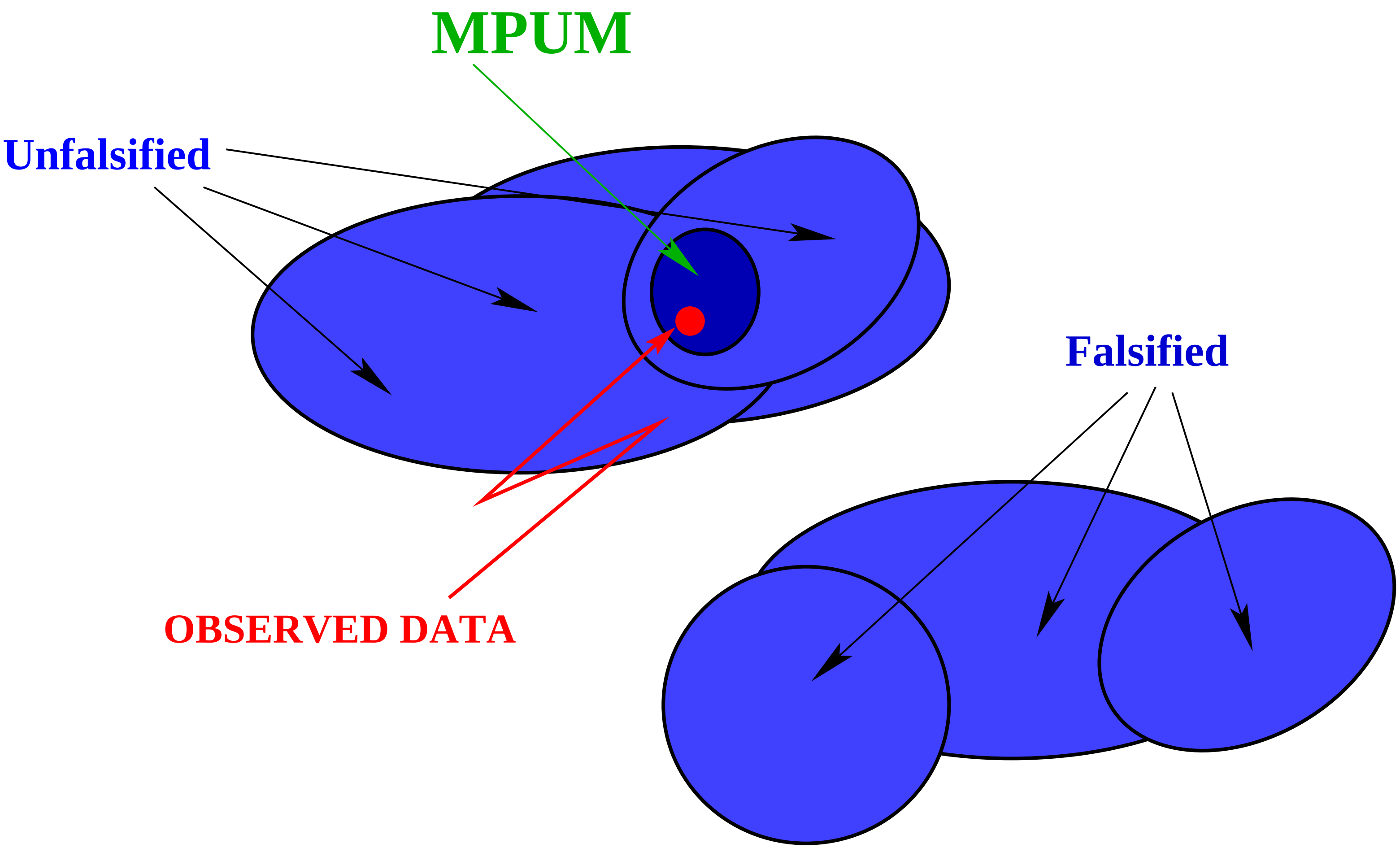
:= the model $\mathscr{B}^\star$ **in** $\mathscr{L}^{\mathtt{w}}$

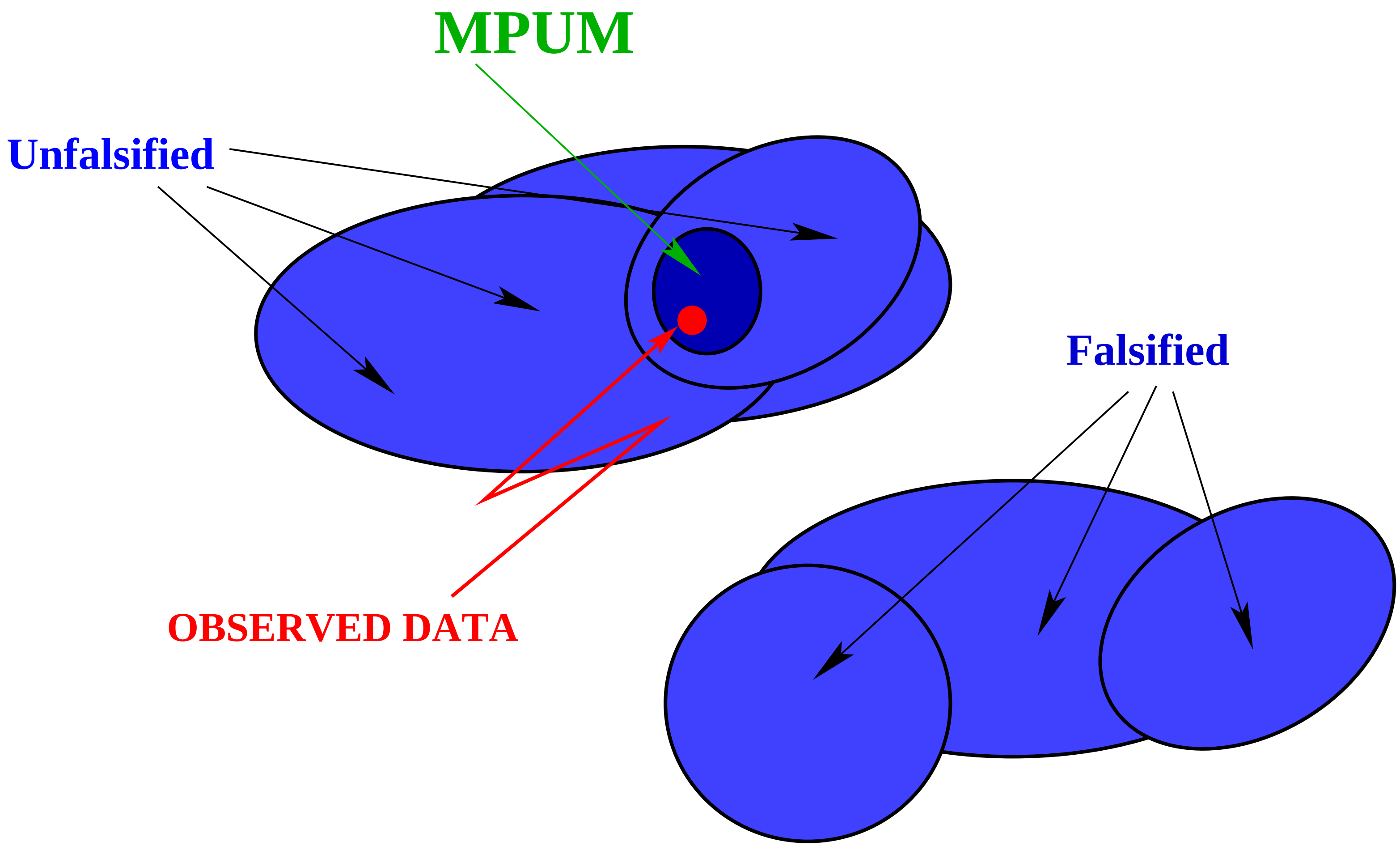
that explains the observations $\rightsquigarrow \tilde{w} \in \mathscr{B}^\star$ **'unfalsified'**

+ as little else as possible $\rightsquigarrow \mathscr{B}^\star$ **'more powerful'**

than any other unfalsified model in $\mathscr{L}^{\mathtt{w}}$

The MPUM = the smallest unfalsified model in $\mathscr{L}^{\mathtt{w}}$


MPUM
Unfalsified
Falsified
OBSERVED DATA



Does the MPUM in $\mathscr{L}^{\mathtt{w}}$ exist?

The MPUM in $\mathscr{L}^{\mathtt{w}}$

MPUM = $\left(\mathtt{linearspan}\{\tilde{w}, \sigma\tilde{w}, \sigma^2\tilde{w}, \ldots\}\right)^{\mathtt{closure}}$

linear, shift-invariant, closed $(\Rightarrow \in \mathscr{L}^{\mathtt{w}}!)$

The MPUM in $\mathscr{L}^{\mathtt{w}}$

MPUM = $(\mathtt{linearspan}\{\tilde{w}, \sigma\tilde{w}, \sigma^2\tilde{w}, \ldots\})^{\mathtt{closure}}$

linear, shift-invariant, closed $(\Rightarrow \in \mathscr{L}^{\mathtt{w}}!)$

Our pbm: Given the observed (infinite horizon) 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 (a representation of) the MPUM in $\mathscr{L}^{\mathtt{w}}$
that generated these observations.

This is what is meant by ‘exact’, ‘deterministic’ modeling

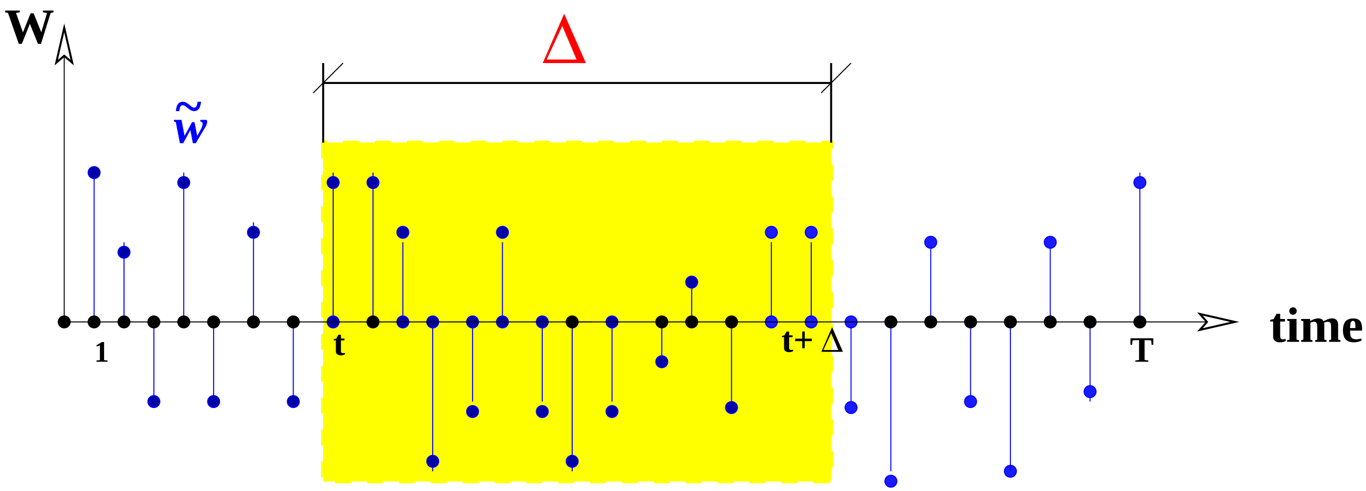
The annihilators

Algorithmic question

Observations

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

Look through the window



for weighted convolution sums that annihilate $\tilde{w}$

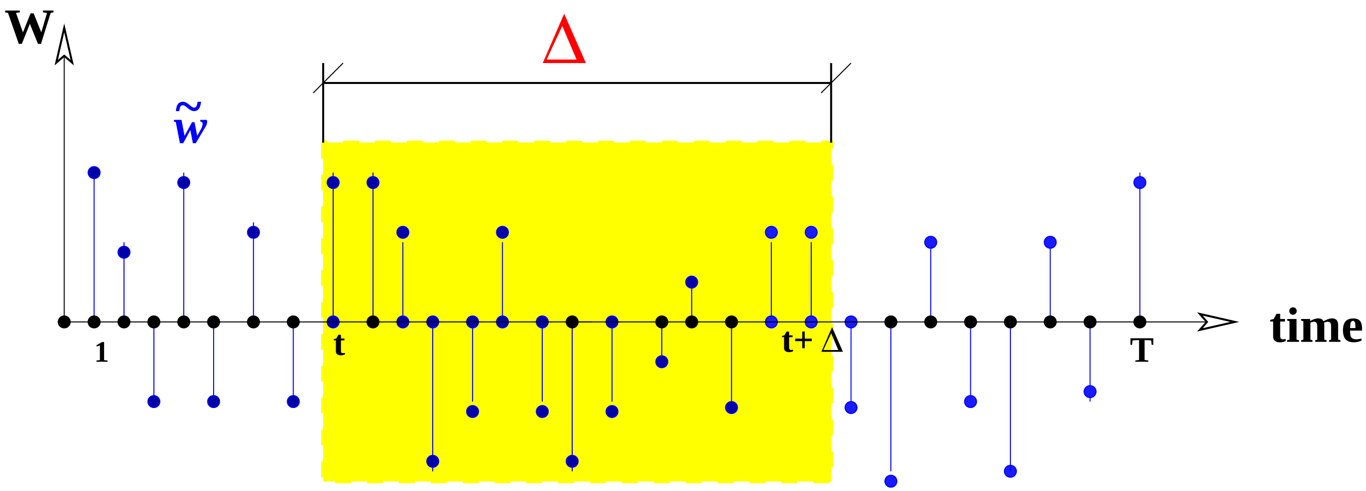
$$a_0 \tilde{w}(t) + a_1 \tilde{w}(t+1) + \cdots + a_\Delta \tilde{w}(t+\Delta) = 0, \qquad a_{\mathtt{k}} \in \mathbb{R}^{1\times \mathtt{w}}$$

Algorithmic question

Observations

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

Look through the window



for weighted convolution sums that annihilate $\tilde{w}$

$$a_0\tilde{w}(t) + a_1\tilde{w}(t+1) + \cdots + a_\Delta \tilde{w}(t+\Delta) = 0, \qquad a_{\mathtt{k}} \in \mathbb{R}^{1\times \mathtt{w}}$$

$\rightsquigarrow$ **'annihilator'** $\qquad a(\xi) = a_0 + a_1\xi + \cdots + a_\Delta \xi^\Delta \in \mathbb{R}[\xi]^{1\times \mathtt{w}}$

Algorithmic question

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

$a_0 + a_1\xi + \ldots + a_L\xi^L$ **is an annihilator** $\Leftrightarrow$

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

Hermann Hankel

$\leadsto$ **the 'Hankel matrix of the data' emerges**

Algorithmic question

Finding the annihilators $\Leftrightarrow$

compute the 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 left kernel is $\{0\}$ **or** ∞**-dimensional ...**

The module of left annihilators

The annihilators as a polynomial module

We identify elements of the left kernel with vector polynomials

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

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

The annihilators as a polynomial module

Closed under addition

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

The annihilators as a polynomial module

Closed under addition

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

The annihilators as a polynomial module

Closed under addition

$$\begin{bmatrix} a_0 & \cdots & a_\Delta & 0 & \cdots \end{bmatrix}$$

$$\begin{bmatrix} b_0 & \cdots & b_\Delta & 0 & \cdots \end{bmatrix}$$

$$\Downarrow$$

$$\begin{bmatrix} a_0 + b_0 \cdots a_\Delta + b_\Delta & 0 & \cdots \end{bmatrix}$$

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

The annihilators as a polynomial module

and under shifting

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

The annihilators as a polynomial module

and under shifting

$$[a_0 \quad a_1 \quad \cdots \quad a_\Delta \quad 0 \quad 0 \cdots]$$

$$\Downarrow$$

$$[\,0 \quad a_0 \quad \cdots \quad a_{\Delta-1} \quad a_\Delta \quad 0 \cdots]$$

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

The annihilators as a polynomial module

$a, b \in \mathbb{R}[\xi]^{1\times \mathtt{w}}$

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

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

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

The annihilators as a polynomial module

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

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

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

$\Rightarrow$ **The left kernel is an** $\mathbb{R}[\xi]$**-submodule of** $\mathbb{R}[\xi]^{1\times \mathtt{w}}$

The annihilators as a polynomial module

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

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

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

The left kernel is an $\mathbb{R}[\xi]$-submodule of $\mathbb{R}[\xi]^{1\times \mathtt{w}}$

and therefore finitely generated $\Rightarrow$

$\exists$ **annihilators** $a(\xi), b(\xi), \cdots, z(\xi)$

'generators' that yield all other annihilators by shifting & lin. combinations

Emmy Noether

$\cong$ **Left kernel is effectively finite dimensional! (dimension $\leq \mathtt{w}$)**

The annihilators as a polynomial module

Collect the generators into a matrix $R = \begin{bmatrix} a \\ b \\ \vdots \\ z \end{bmatrix}$

$\rightsquigarrow$ **MPUM** $R(\sigma)w = 0$

How can we obtain a module basis of the left kernel of the Hankel matrix of the data?

Recursive computation of the generators

$\tilde{w} \mapsto$ left kernel

Suppose we have a left annihilator of

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

Does this simplify finding other left annihilators of

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

Unimodular completion

The unimodular completion lemma

<u>Lemma</u>: $R(\xi) \in \mathbb{R}^{\mathtt{p}\times\mathtt{w}}[\xi]$ **left prime** $\Leftrightarrow \exists\ E(\xi) \in \mathbb{R}^{(\mathtt{w}-\mathtt{p})\times\mathtt{w}}[\xi]$ **:**

$$\begin{bmatrix} R(\xi) \\ E(\xi) \end{bmatrix}$$ **is unimodular (**$\det =$ **a non-zero constant)**

defining property of *Hermite rings*

Charles Hermite

The unimodular completion lemma

<u>Lemma</u>: $R(\xi) \in \mathbb{R}^{\mathtt{p}\times\mathtt{w}}[\xi]$ **left prime** $\Leftrightarrow \exists\ E(\xi) \in \mathbb{R}^{(\mathtt{w}-\mathtt{p})\times\mathtt{w}}[\xi]$:

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

<u>Example</u>: $\mathtt{p} = 1, \mathtt{w} = 2,$

$$\begin{matrix} R(\xi) = [r_1(\xi) \ \ r_2(\xi)] \\ E(\xi) = [-y(\xi) \ \ x(\xi)] \end{matrix} \quad \leadsto \quad \begin{bmatrix} r_1 & r_2 \\ -y & x \end{bmatrix}$$

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

$$r_1(\xi)x(\xi) + r_2(\xi)y(\xi) = 1 \quad \textbf{(Bézout eq'n)}$$

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

Étienne Bézout

The unimodular completion lemma

Lemma: $R(\xi) \in \mathbb{R}^{\mathtt{p}\times\mathtt{w}}[\xi]$ **left prime** $\Leftrightarrow \exists \; E(\xi) \in \mathbb{R}^{(\mathtt{w}-\mathtt{p})\times\mathtt{w}}[\xi]$:

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

Equivalent proposition:

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

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

iff $\mathscr{B}$ **is controllable (in the behavioral sense).**

The unimodular completion lemma

Lemma: $R(\xi) \in \mathbb{R}^{\mathtt{p}\times\mathtt{w}}[\xi]$ **left prime** $\Leftrightarrow \exists \; E(\xi) \in \mathbb{R}^{(\mathtt{w}-\mathtt{p})\times\mathtt{w}}[\xi]$ **:**

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

Equivalent proposition:

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

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

iff $\mathscr{B}$ **is controllable (in the behavioral sense).**

Controllability - henceforth assumed where needed.

Recursive computation of the generators

Application to left kernel computation

Assume

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

$\rightsquigarrow$ **annihilator**

$$a(\xi) = a_0 + a_1\xi + \cdots + a_{\mathtt{n}_1}\xi^{\mathtt{n}_1} \in \mathbb{R}[\xi]^{1\times\mathtt{w}}$$

Application to left kernel computation

Assume

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

$\rightsquigarrow$ **annihilator**

$$a(\xi) = a_0 + a_1\xi + \cdots + a_{\mathtt{n}_1}\xi^{\mathtt{n}_1} \in \mathbb{R}[\xi]^{1\times\mathtt{w}}$$

Complete $a(\xi) \mapsto E_a(\xi) \in \mathbb{R}[\xi]^{(\mathtt{w}-1)\times\mathtt{w}}$, $\begin{bmatrix} a(\xi) \\ E_a(\xi) \end{bmatrix}$ **unimodular**

Application to left kernel computation

Assume

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

Complete $a(\xi) \mapsto E_a(\xi)$ $\qquad a(\sigma)\tilde{w} = 0$

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

means: projecting $\tilde{w}$ onto $\mathscr{B}'$ in

$$\{w \mid a(\sigma)w = 0\} \oplus \mathscr{B}' = (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$$

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

Application to left kernel computation

Assume

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

Compute annihilator

$$\begin{bmatrix} b_0 & b_1 & \cdots & b_{\mathtt{n}_2} \end{bmatrix} \begin{bmatrix} \tilde{e}(1) & \tilde{e}(2) & \cdots & \tilde{e}(t) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \tilde{e}(\mathtt{n}_2+1) & \tilde{e}(\mathtt{n}_2+2) & \cdots & \tilde{e}(t+\mathtt{n}_2) & \cdots \end{bmatrix} = 0$$

Yields $b(\xi)$ **and a second annihilator** $b(\xi)E_a(\xi)$ **for** $\tilde{w}$

Complete $b \rightsquigarrow E_b$, **compute** $\tilde{\tilde{e}} = E_b(\sigma)\tilde{e}$, **find annihilator** c

Yields a third annihilator $c(\xi)E_b(\xi)E_a(\xi)$ **for** $\tilde{w}$, **etc.**

Application to left kernel computation

Recursively,

$$\tilde{w} \mapsto a(\xi) \mapsto E_a(\xi) \mapsto E_a(\sigma)\tilde{w} = \tilde{e}_{E_a}$$

$$\tilde{e}_{E_a} \mapsto b(\xi) \mapsto E_b(\xi) \mapsto E_b(\sigma)\tilde{e}_{E_a} = \tilde{e}_{E_b}$$

$$\vdots$$

$$\tilde{e}_{E_y} \mapsto z(\xi)$$

$\rightsquigarrow$ **annihilators** $a,\ bE_a,\ cE_bE_a,\ \ldots\ ,\ zE_y \cdots E_bE_a$

$\Rightarrow$ **a module basis of the left kernel obtained by computing** p **times a left kernel vector.**

Le coup de grâce

$\rightsquigarrow$ **MPUM**

$$\begin{bmatrix} a \\ bE_a \\ cE_bE_a \\ \vdots \\ zE_y\cdots E_bE_a \end{bmatrix}(\sigma)\, w = 0$$

Le coup de grâce

Amenable to approximate LA LS SVD implementation by examining the SV's of the truncated Hankel matrices

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

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

etc.

Concluding Remarks

Summary

- **Exact deterministic modeling** $\rightsquigarrow$ **MPUM**

Summary

- **Exact deterministic modeling $\rightsquigarrow$ MPUM**
- **The left kernel of a Hankel matrix is a polynomial module therefore finitely generated.**

  Hence 'effectively' finite dimensional

Summary

- **Exact deterministic modeling** $\rightsquigarrow$ **MPUM**
- **The left kernel of a Hankel matrix is a polynomial module therefore finitely generated.**

  Hence 'effectively' finite dimensional
- **A basis can be computed recursively using the completion lemma and error propagation**

Summary

- **Exact deterministic modeling $\rightsquigarrow$ MPUM**
- **The left kernel of a Hankel matrix is a polynomial module therefore finitely generated.**

  Hence 'effectively' finite dimensional
- **A basis can be computed recursively using the completion lemma and error propagation**
- **Requires computing p vectors in kernel of Hankel matrix of the 'errors'.**

Summary

- **Exact deterministic modeling $\leadsto$ MPUM**
- **The left kernel of a Hankel matrix is a polynomial module therefore finitely generated.**

  Hence 'effectively' finite dimensional
- **A basis can be computed recursively using the completion lemma and error propagation**
- **Requires computing $\mathtt{p}$ vectors in kernel of Hankel matrix of the 'errors'.**
- **These computations can be executed (approximately) using numerical LA.**

Summary

- **Exact deterministic modeling $\rightsquigarrow$ MPUM**
- **The left kernel of a Hankel matrix is a polynomial module therefore finitely generated.**

  Hence 'effectively' finite dimensional
- **A basis can be computed recursively using the completion lemma and error propagation**
- **Requires computing $\mathtt{p}$ vectors in kernel of Hankel matrix of the 'errors'.**
- **These computations can be executed (approximately) using numerical LA.**
- **Note the crucial role of the (Hermite) module structure!**

Extensions

- T **finite, missing data**

Extensions

- T **finite, missing data**
- **Multiple observed time-series**

Extensions

- T **finite, missing data**
- **Multiple observed time-series**
- **Lack of controllability**

Extensions

- T **finite, missing data**
- **Multiple observed time-series**
- **Lack of controllability**
- **Orthogonality in the matrix completion lemma.**

Given $\mathscr{B} \in \mathscr{L}^{\mathtt{w}}$**, controllable, find** $\mathscr{B}' \in \mathscr{L}^{\mathtt{w}}$ **such that**

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

with

$$\mathscr{B} \cap \ell_2(\mathbb{N}, \mathbb{R}^{\mathtt{w}}) \ \perp \ \mathscr{B}' \cap \ell_2(\mathbb{N}, \mathbb{R}^{\mathtt{w}})$$

$\rightsquigarrow$ **slightly more complex, better behaved algorithms.**

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

Jan.Willems@esat.kuleuven.be

http://www.esat.kuleuven.be/∼jwillems

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

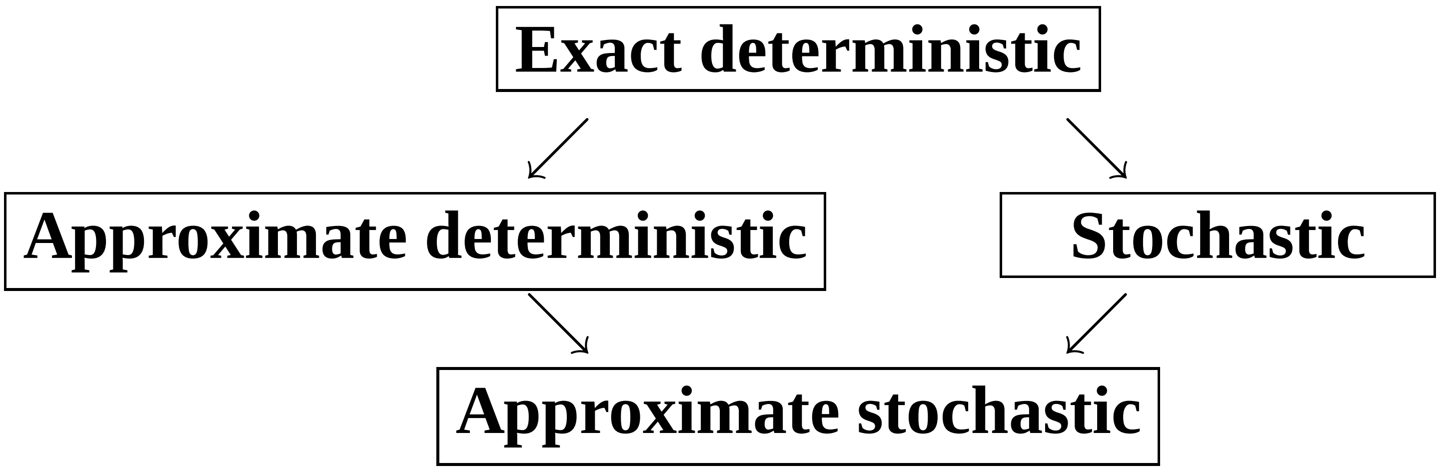
Thank you

Thank you

Thank you

Thank you

Thank you


Exact deterministic
Approximate deterministic
Stochastic
Approximate stochastic

From Ivan's Ph.D. dissertation

Markovsky, I., et al., *Exact and Approximate Modeling of Linear Systems: A Behavioral Approach*, SIAM series on Mathematical Modeling and Computation, volume 11, 2006.

Comparison

Performance

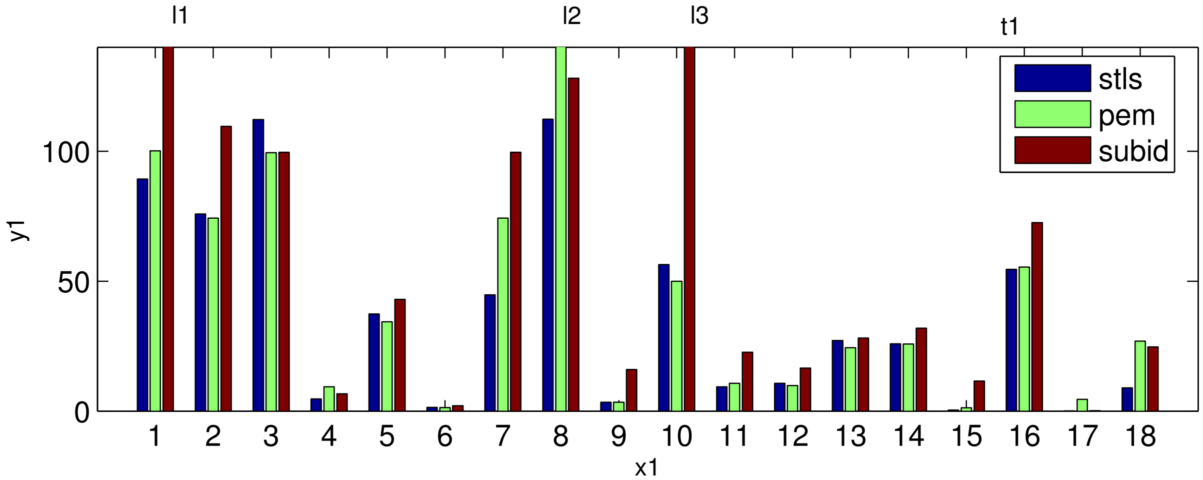
#	Data set name	T	m	p	l
1	Data of the western basin of Lake Erie	57	5	2	1
2	Data of Ethane-ethylene column	90	5	3	1
3	Data of a 120 MW power plant	200	5	3	2
4	Heating system	801	1	1	2
5	Data from an industrial dryer	867	3	3	1
6	Data of a hair dryer	1000	1	1	5
7	Data of the ball-and-beam setup in SISTA	1000	1	1	2
8	Wing flutter data	1024	1	1	5
9	Data from a flexible robot arm	1024	1	1	4
10	Data of a glass furnace (Philips)	1247	3	6	1
11	Heat flow density through a two layer wall	1680	2	1	2
12	Simulation of a pH neutralization process	2001	2	1	6
13	Data of a CD-player arm	2048	2	2	1
14	Data from an industrial winding process	2500	5	2	2
15	Liquid-saturated heat exchanger	4000	1	1	2
16	Data from an evaporator	6305	3	3	1
17	Continuous stirred tank reactor	7500	1	2	1
18	Model of a steam generator	9600	4	4	1

Comparison

Performance

Compare the misfit on the last 30% of the outputs and the execution time for computing the ID model from the first 70% of the data.

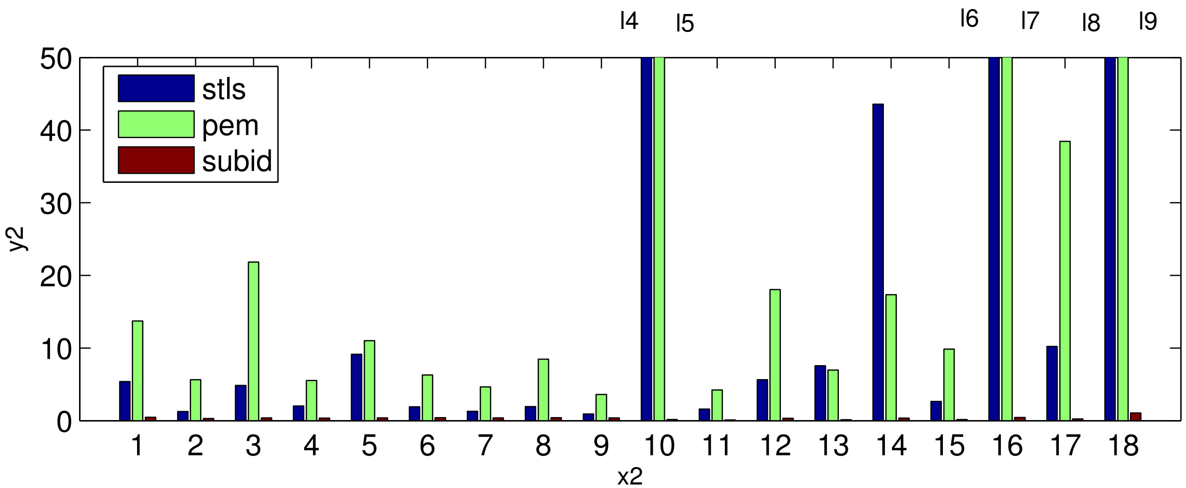
Misfit



Comparison

Performance

Execution time



Comparison

Performance