RECURSIVE COMPUTATION

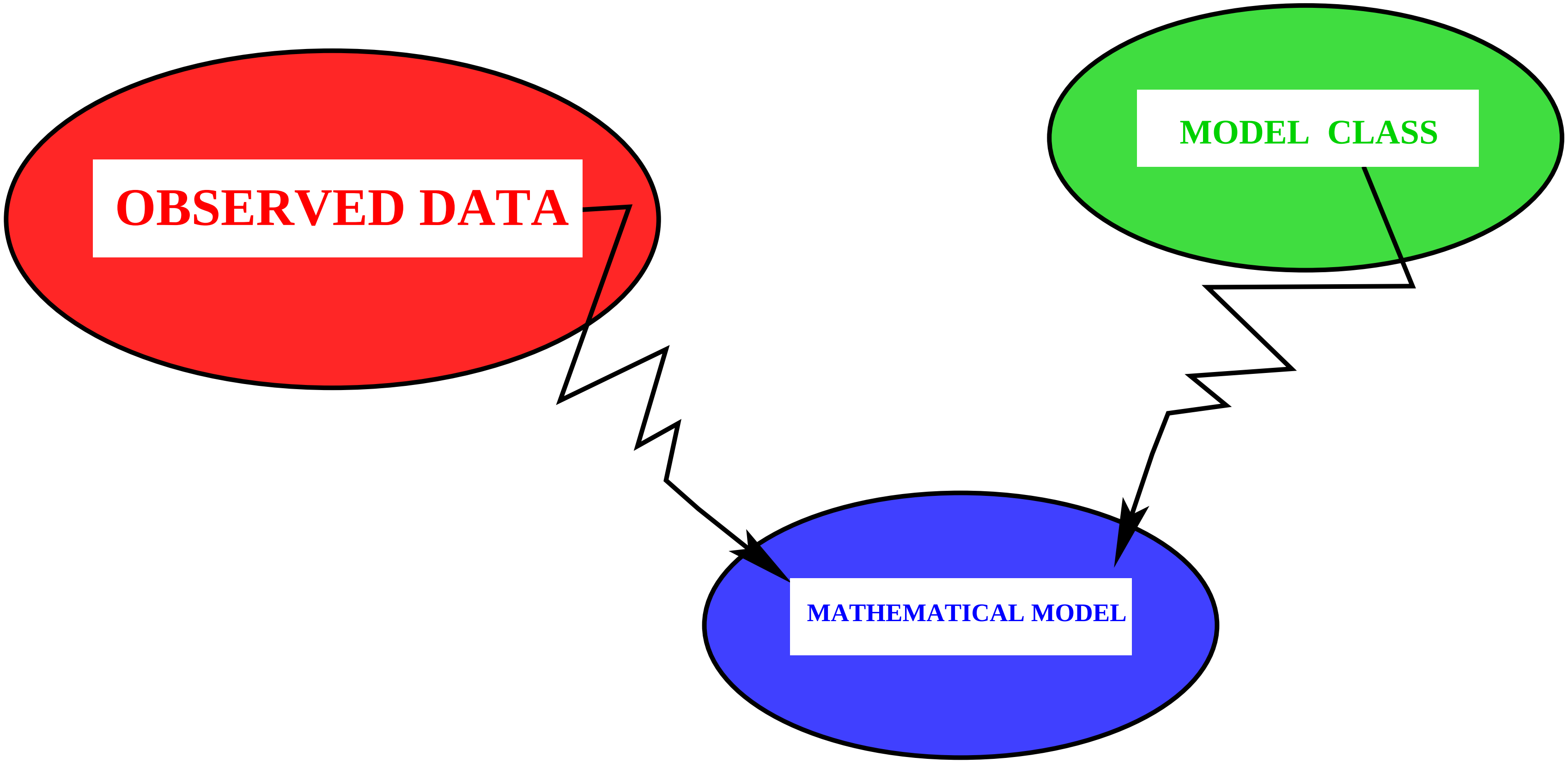
OF THE MPUM

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

Seminar M.I.T. **December 10, 2007**

System identification (SYSID)

System ID



Case of interest today

Observations: **vector time-series**

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

Case of interest today

Observations: **vector time-series**

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

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

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

Usually

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

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

Case of interest today

Observations: **vector time-series**

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

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

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

Usually

$$\begin{aligned} R_0 w(t) + R_1 w(t+1) + \cdots + R_L w(t+L) \\ = M_0 \varepsilon(t) + M_1 \varepsilon(t+1) + \cdots + M_L \varepsilon(t+L) \end{aligned}$$

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

Case of interest today

Observations: **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 to approximation)
3. **linear, time-invariant, controllable systems**

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

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

$\mathscr{B}$ **belongs to the** **model class** $\mathscr{L}^{\mathtt{w}}$ $:\Leftrightarrow$

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

  $:\Leftrightarrow$ **'prefix determined'**
- $\exists$ **matrices** $R_0, R_1, \ldots, R_L$ **such that**

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

$$\boxed{R(\sigma) w = 0} \qquad \sigma := \textbf{ left shift}$$

The model class

$\mathscr{B}$ **belongs to the model class** $\mathscr{L}^{\mathtt{w}}$ $:\Leftrightarrow$

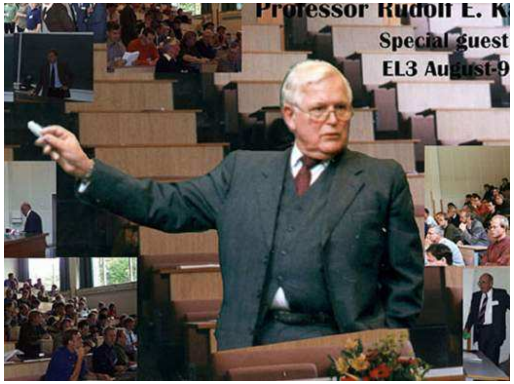
- $\mathscr{B}$ **is linear, shift-invariant, and closed**
- $\mathscr{B}$ **is linear, time-invariant, and complete**
- $R(\sigma)w = 0$**, including input/output partition** $\rightsquigarrow$

$$P(\sigma)y = Q(\sigma)u, \quad w \cong \begin{bmatrix} u \\ y \end{bmatrix} \quad \det(P) \neq 0, P^{-1}Q \text{ **proper**}$$

The model class

$\mathscr{B}$ **belongs to the model class** $\mathscr{L}^{\mathtt{w}}$ $:\Leftrightarrow$

- $\mathscr{B}$ **is linear, shift-invariant, and closed**
- $\mathscr{B}$ **is linear, time-invariant, and complete**
- $R(\sigma)w = 0$
- $P(\sigma)y = Q(\sigma)u \quad w \cong \begin{bmatrix} u \\ y \end{bmatrix}$
- $\exists$ **matrices** A, B, C, D **such that** $\mathscr{B}$ **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 MPUM

The MPUM

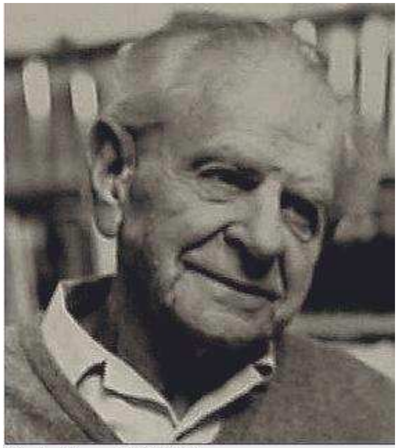
Given the observed (infinite-horizon) time-series

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

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

$\mathscr{B}_1 \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$ **is 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
(1902-1994)

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

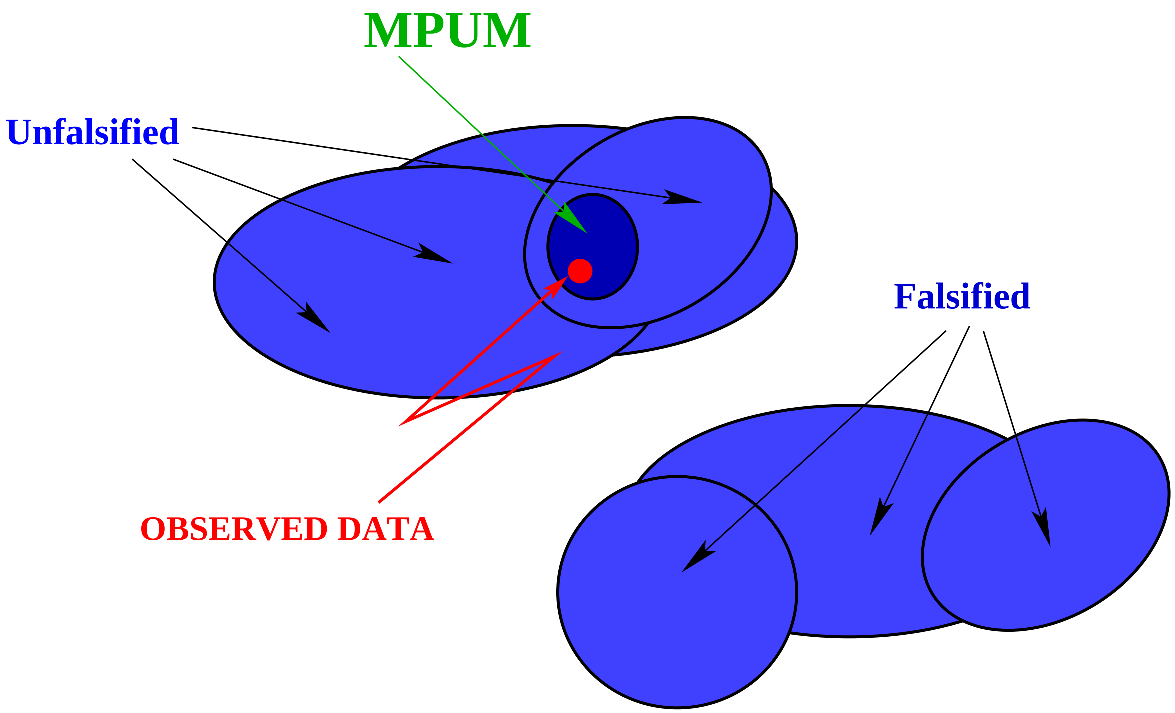
$\mathscr{B}_1 \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$ **is more powerful than** $\mathscr{B}_2 \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$ $:\Leftrightarrow \mathscr{B}_1 \subset \mathscr{B}_2$

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

:= the model in $\mathscr{L}^{\mathtt{w}}$
that explains the observations $\rightsquigarrow$ $\tilde{w} \in \mathscr{B}$ **'unfalsified'**
+ as little else as possible $\rightsquigarrow$ **'most powerful'**

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

The MPUM



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

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

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

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

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

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 the MPUM in $\mathscr{L}^{\mathtt{w}}$ that generated the data.

This is what is meant by 'exact' modeling

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

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

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 the MPUM in $\mathscr{L}^{\mathtt{w}}$ that generated the data.

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

Note that classical realization theory is a special case

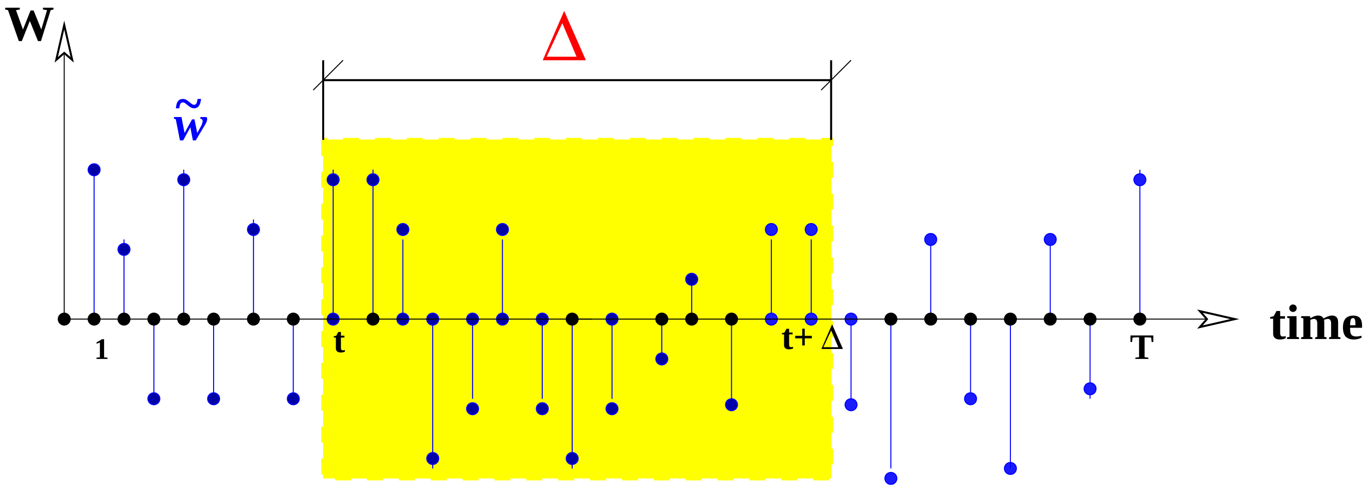
The annihilators

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 annihilators

$$r_0 \tilde{w}(t) + r_1 \tilde{w}(t+1) + \cdots + r_\Delta \tilde{w}(t+\Delta) = 0, \quad r_{\mathtt{k}}\text{'s} \in \mathbb{R}^{1\times\mathtt{w}}$$

$\rightsquigarrow$ **'annihilator'** $\qquad n(\xi) = r_0 + r_1\xi + \cdots + r_\Delta \xi^\Delta \in \mathbb{R}[\xi]^{1\times\mathtt{w}}$

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 annihilators

$$r_0 \tilde{w}(t) + r_1 \tilde{w}(t+1) + \cdots + r_\Delta \tilde{w}(t+\Delta) = 0, \quad r_{\mathtt{k}}\text{'s} \in \mathbb{R}^{1\times \mathtt{w}}$$

$\rightsquigarrow$ **'annihilator'** $\qquad n(\xi) = r_0 + r_1\xi + \cdots + r_\Delta \xi^\Delta \in \mathbb{R}[\xi]^{1\times \mathtt{w}}$

Collect 'all' the annihilators into $\quad R = \begin{bmatrix} n_1 \\ n_2 \\ \vdots \\ n_{\mathtt{p}} \end{bmatrix} \in \mathbb{R}[\xi]^{\mathtt{p}\times \mathtt{w}}$

$\rightsquigarrow$ **MPUM** $\qquad \boxed{R(\sigma)w = 0}$

Algorithmic question

Given

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

find 'all' $R_0, R_1, \ldots R_L$ **such that** $R(\sigma)\tilde{w} = 0$**, i.e.**

$$\boxed{\begin{bmatrix} R_0 & R_1 \cdots R_L & 0 \cdots \end{bmatrix}} \begin{bmatrix} \tilde{w}(1) & \tilde{w}(2) & \cdots & \tilde{w}(t'') & \cdots \\ \tilde{w}(2) & \tilde{w}(3) & \cdots & \tilde{w}(t''+1) & \cdots \\ \tilde{w}(3) & \tilde{w}(4) & \cdots & \tilde{w}(t''+2) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \tilde{w}(t') & \tilde{w}(t'+1) & \cdots & \tilde{w}(t'+t''-1) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} = 0$$

That is how the 'Hankel matrix of the data' emerges

Algorithmic question

Compute the left kernel of the data Hankel matrix

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

∞-dimensional ...

The module of left annihilators

Annihilators as polynomials

Identify elements of the left kernel with vector polynomials

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

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

Annihilators as polynomials

Closed under addition

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

Annihilators as polynomials

Closed under addition

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

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

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

Annihilators as polynomials

and under shifting

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

$$\Downarrow$$

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

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

Annihilators as polynomials

and under shifting

$$\begin{bmatrix} a_0 & a_1 & \cdots & a_\Delta & 0 & 0 & \cdots \end{bmatrix}$$

$$\Downarrow$$

$$\begin{bmatrix} 0 & a_0 & \cdots & a_{\Delta-1} & a_\Delta & 0 & \cdots \end{bmatrix}$$

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

$$a(\xi) = a_0 + a_1\xi + \cdots + a_\Delta \xi^\Delta \quad \in \textbf{ left kernel}$$
$$b(\xi) = b_0 + b_1\xi + \cdots + b_\Delta \xi^\Delta \quad \in \textbf{ left kernel}$$

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

Annihilators as polynomials

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

$a(\xi) = a_0 + a_1\xi + \cdots + a_\Delta \xi^\Delta \quad \in$ **left kernel**
$b(\xi) = b_0 + b_1\xi + \cdots + b_\Delta \xi^\Delta \quad \in$ **left kernel**

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

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

and is therefore finitely generated:
$\exists$ **annihilators** $a(\xi), b(\xi), \cdots, z(\xi)$
that yield all other annihilators under shifting and +

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

Annihilators as polynomials

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

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

How can we a module basis of the left kernel?

Recursive computation of the generators

$\tilde{w} \mapsto$ left kernel

Suppose we found a left annihilator of

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

Does this simplify finding other left annihilators of

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

The unimodular completion lemma

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

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

$R \in \mathbb{R}[\xi]^{\bullet\times\bullet}$ **is left prime as a polynomial matrix**

$:\Leftrightarrow\ \ R = FR' \quad \Rightarrow F$ **unimodular**

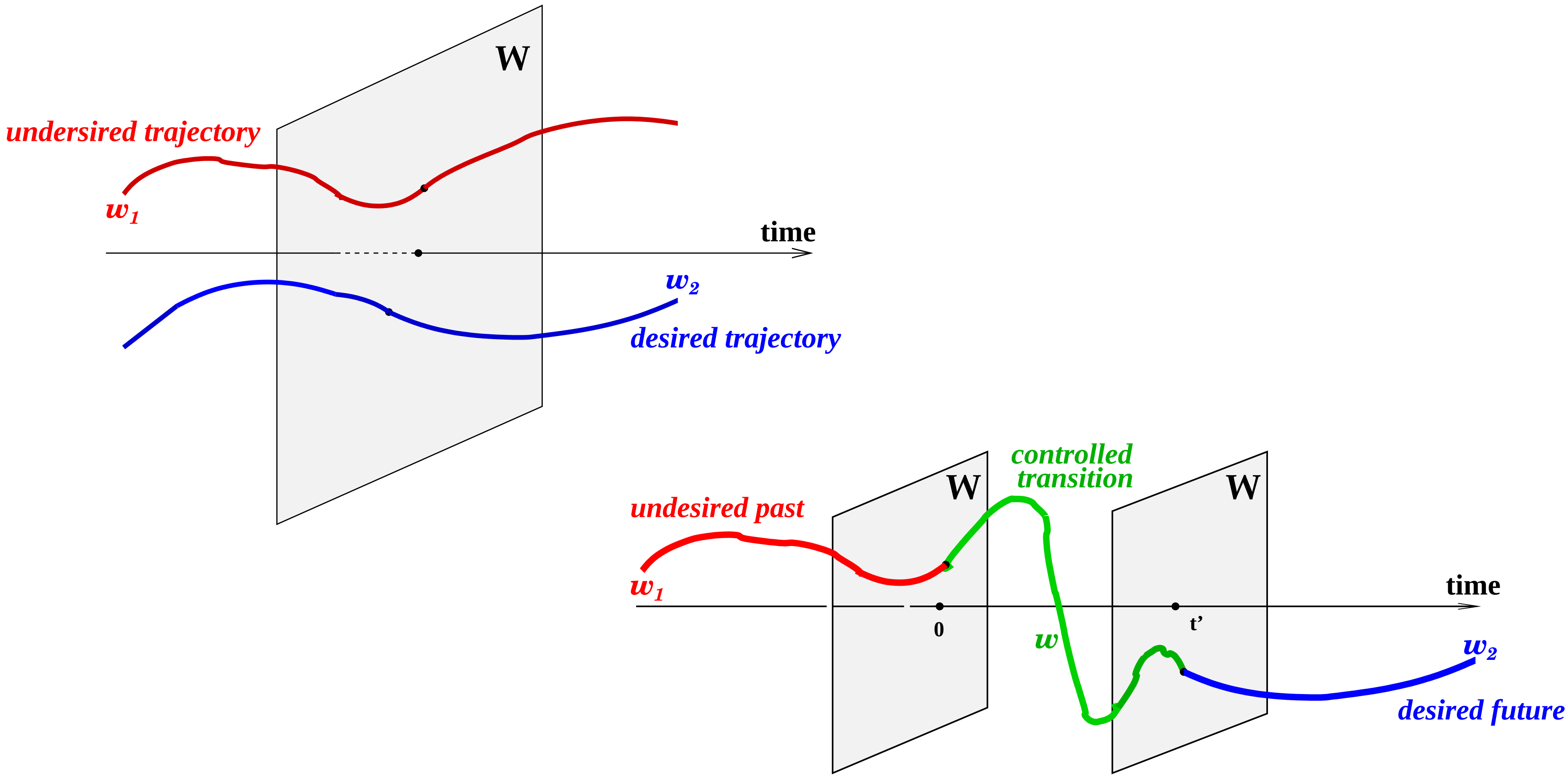
$\Leftrightarrow R(\lambda)$ **has full row rank** $\forall\ \lambda \in \mathbb{C}$

$\Leftrightarrow R(\sigma)w = 0$ **defines a controllable system**

$\Leftrightarrow \exists$ **a unimodular completion**

The unimodular completion lemma

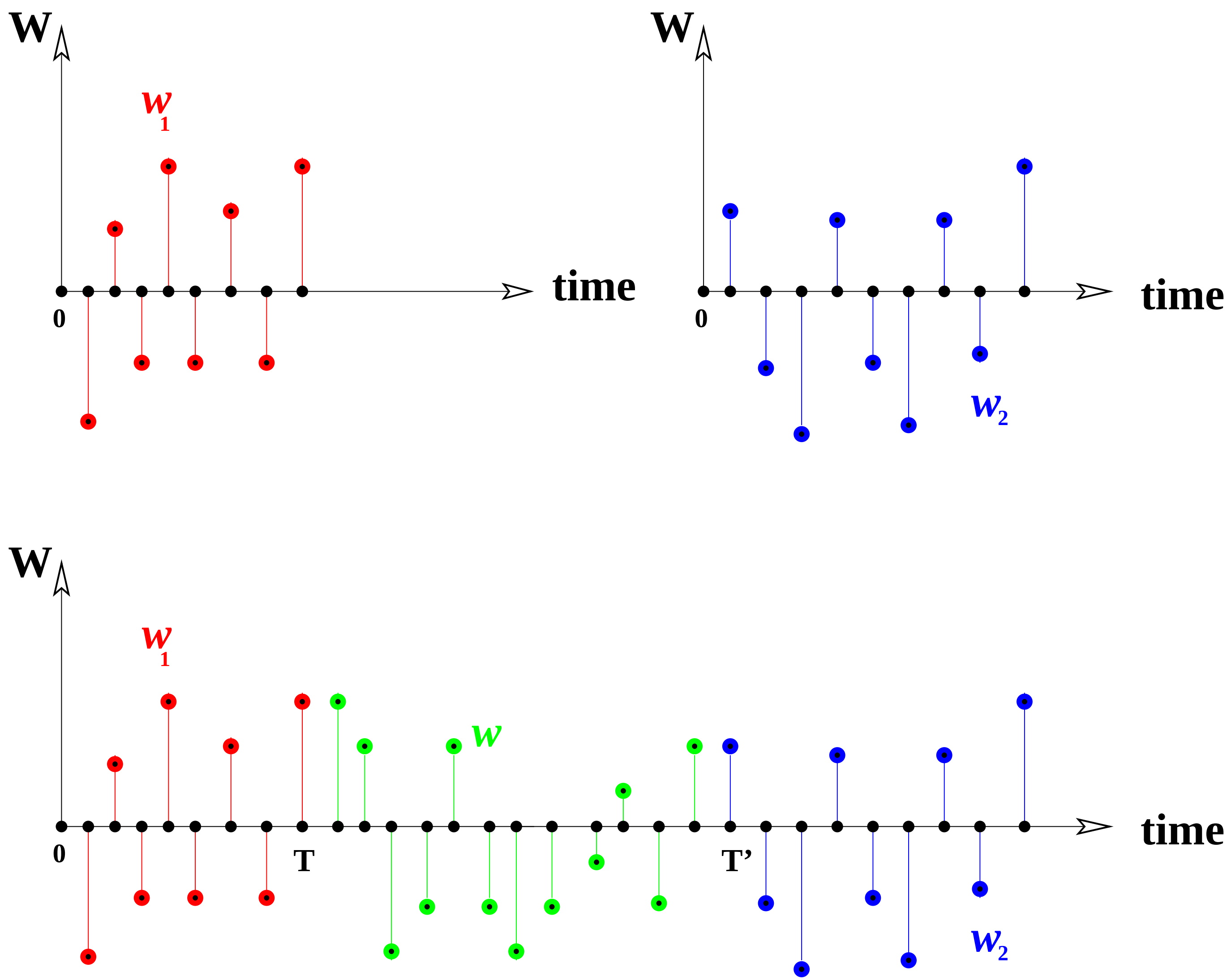
Controllability is used here in the behavioral sense.



Behavioral controllability of a dynamical system

The unimodular completion lemma

Controllability is used here in the behavioral sense.



The unimodular completion lemma

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

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

Example: $\mathtt{p} = 1, \mathtt{w} = 2,$

$$\begin{aligned} R(\xi) &= [r_1(\xi) \;\; r_2(\xi)] \\ E(\xi) &= [-y(\xi) \;\; x(\xi)] \end{aligned} \qquad \rightsquigarrow \qquad \begin{bmatrix} r_1 & r_2 \\ -y & x \end{bmatrix}$$

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

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

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

The unimodular completion lemma

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

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

Equivalent proposition:

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

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

iff $\mathscr{B}$ **is controllable.**

The unimodular completion lemma

Controllability - assumed where needed

$$\downarrow\downarrow\downarrow$$

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

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

Application to left kernel computation

Assume

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

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 $a(\xi) \mapsto E_a(\xi)$ **,** $\begin{bmatrix} a(\xi) \\ E_a(\xi) \end{bmatrix}$ **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 $a(\xi) \mapsto E_a(\xi)$

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

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

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 a second annihilator $b(\xi)E_a(\xi)$

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

Yields a third annihilator $c(\xi)E_b(\xi)E_a(\xi)$

Application to left kernel computation

Recursively,

$$\tilde{w} \mapsto \boxed{a(\xi)} \mapsto E_a(\xi)$$
$$\mapsto \tilde{e}_{E_a} \mapsto \boxed{b(\xi)} \mapsto E_b(\xi)$$
$$\vdots$$
$$\mapsto \tilde{e}_{E_y} \mapsto \boxed{z(\xi)}$$

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

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

Application to left kernel computation

Recursively,

$$\tilde{w} \mapsto \boxed{a(\xi)} \mapsto E_a(\xi)$$
$$\mapsto \tilde{e}_{E_a} \mapsto \boxed{b(\xi)} \mapsto E_b(\xi)$$
$$\vdots$$
$$\mapsto \tilde{e}_{E_y} \mapsto \boxed{z(\xi)}$$

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

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

Amenable to approximate LA SVD LS implementation

From data to state

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

Go directly from an external trajectory

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

to the corresponding MPUM state trajectory

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

Effective in SYSID $\rightsquigarrow$ **Subspace ID**

Subspace ID

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

$$\downarrow$$

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

$$\downarrow$$

row reduce $\tilde{X}$

$$\downarrow$$

LS solve

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

$$\downarrow$$

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

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

Go directly from an external trajectory

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

to the corresponding MPUM state trajectory

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

How does this work? Nice system theoretic question!

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

Henceforth, assume Δ sufficiently large.

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

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

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

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

or

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

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

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

There are many algorithms that do this. We discuss two.

From data to state by past/future intersection

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

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

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

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

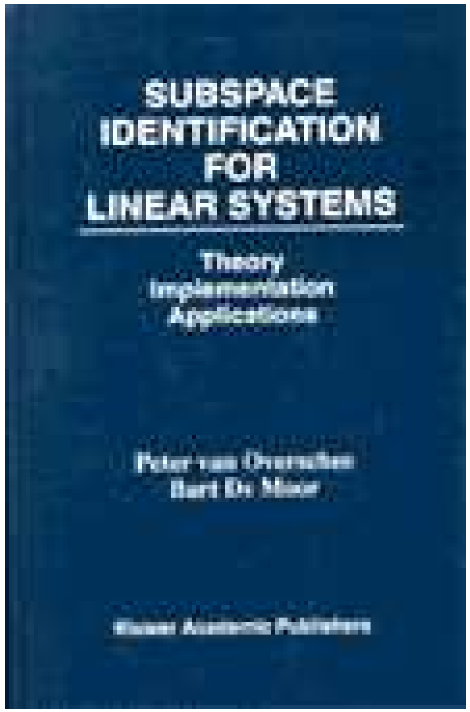
Intersection of spans of rows of $\mathscr{H}_-$ and $\mathscr{H}_+$ = state space.
The common linear combinations

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

State = what is common between past and future.

Stochastic versions

This past/future idea is used effectively, also for ARMAX systems, in 'subspace ID'



From data to state using the left annihilators

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

Compute 'the' left annihilators of the Hankel matrix:

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

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

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

Then

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

$$=$$

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

Then

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

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

$$=$$

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

$\uparrow\uparrow\uparrow$

‘shift-and-cut’

Paolo Rapisarda

Subspace ID and the module of annihilators

State construction via the generators

Generators

$$\begin{array}{l}
[a_0 \quad a_1 \quad \cdots \quad a_{\mathtt{n}_1}] \\
[b_0 \quad b_1 \quad \cdots \quad \cdots \quad b_{\mathtt{n}_2}] \\
\qquad\qquad \vdots \\
[z_0 \quad z_1 \quad \cdots \quad \cdots \quad \cdots \quad z_{\mathtt{n}_\mathtt{p}}]
\end{array}$$

State construction via the generators

Generators

$$\begin{bmatrix} a_0 & a_1 & \cdots & a_{\mathrm{n}_1} \end{bmatrix}$$
$$\begin{bmatrix} b_0 & b_1 & \cdots & \cdots & b_{\mathrm{n}_2} \end{bmatrix}$$
$$\vdots$$
$$\begin{bmatrix} z_0 & z_1 & \cdots & \cdots & \cdots & z_{\mathrm{n}_\mathrm{p}} \end{bmatrix}$$

Then

$$\begin{bmatrix} a_1 & \cdots & a_{\mathrm{n}_1-1} & a_{\mathrm{n}_1} & 0 & \cdots \\ a_2 & \cdots & a_{\mathrm{n}_1} & 0 & 0 & \cdots \\ \vdots & \vdots\vdots\vdots & \vdots & \vdots & \vdots & \vdots \\ a_{\mathrm{n}_1} & 0 & \cdots & 0 & 0 & \cdots \\ \vdots & \vdots\vdots\vdots & \vdots & \vdots & \vdots & \vdots \\ z_1 & \cdots & \cdots & \cdots & z_{\mathrm{n}_\mathrm{p}-1} & z_{\mathrm{n}_\mathrm{p}} \\ z_2 & \cdots & \cdots & \cdots & z_{\mathrm{n}_\mathrm{p}} & 0 \\ \vdots & \vdots\vdots\vdots & \vdots & \vdots & \vdots & \vdots \\ z_{\mathrm{n}_\mathrm{p}} & 0 & \cdots & \cdots & 0 & 0 \end{bmatrix}
\begin{bmatrix} \tilde{x}(1) & \tilde{x}(2) & \cdots & \tilde{x}(t) & \cdots \end{bmatrix}
=
\begin{bmatrix} \tilde{w}(1) & \tilde{w}(2) & \cdots & \tilde{w}(t) & \cdots \\ \tilde{w}(2) & \tilde{w}(3) & \cdots & \tilde{w}(t+1) & \cdots \\ \tilde{w}(3) & \tilde{w}(4) & \cdots & \tilde{w}(t+2) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \\ \tilde{w}(c_{\mathrm{n}_\mathrm{p}}) & \tilde{w}(c_{\mathrm{n}_\mathrm{p}}+1) & \cdots & \tilde{w}(t+c_{\mathrm{n}_\mathrm{p}}-1) & \cdots \end{bmatrix}$$

State construction via the generators

Then

$$
\begin{bmatrix}
a_1 & \cdots & a_{\mathrm{n}_1-1} & a_{\mathrm{n}_1} & 0 & \cdots \\
a_2 & \cdots & a_{\mathrm{n}_1} & 0 & 0 & \cdots \\
\vdots & \vdots\vdots\vdots & \vdots & \vdots & \vdots & \vdots \\
a_{\mathrm{n}_1} & 0 & \cdots & 0 & 0 & \cdots \\
\vdots & \vdots\vdots\vdots & \vdots & \vdots & \vdots & \vdots \\
z_1 & \cdots & \cdots & \cdots & z_{\mathrm{n}_\mathrm{p}-1} & z_{\mathrm{n}_\mathrm{p}} \\
z_2 & \cdots & \cdots & \cdots & z_{\mathrm{n}_\mathrm{p}} & 0 \\
\vdots & \vdots\vdots\vdots & \vdots & \vdots & \vdots & \vdots \\
z_{\mathrm{n}_\mathrm{p}} & 0 & \cdots & \cdots & 0 & 0
\end{bmatrix}
\begin{bmatrix} \tilde{x}(1) & \tilde{x}(2) & \cdots & \tilde{x}(t) & \cdots \end{bmatrix}
=
\begin{bmatrix}
\tilde{w}(1) & \tilde{w}(2) & \cdots & \tilde{w}(t) & \cdots \\
\tilde{w}(2) & \tilde{w}(3) & \cdots & \tilde{w}(t+1) & \cdots \\
\tilde{w}(3) & \tilde{w}(4) & \cdots & \tilde{w}(t+2) & \cdots \\
\vdots & \vdots & \vdots & \vdots & \\
\tilde{w}(c_{\mathrm{n}_\mathrm{p}}) & \tilde{w}(c_{\mathrm{n}_\mathrm{p}}+1) & \cdots & \tilde{w}(t+c_{\mathrm{n}_\mathrm{p}}-1) & \cdots
\end{bmatrix}
$$

Combined with 'shortest lag' conditions $\rightsquigarrow$ minimal state.

Summary

Summary

Subspace ID:

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

$$\downarrow$$

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

$$\downarrow$$

row reduce $\tilde{X}$

$$\downarrow$$

LS solve

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

$$\downarrow$$

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

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

This is a module of dimension $\leq \mathtt{w}$

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

Summary

Assume $a, b, \cdots, z$ **basis of left-annihilator module**

$$\begin{bmatrix} a_1 & \cdots & a_{\mathrm{n}_1-1} & a_{\mathrm{n}_1} & 0 & \cdots \\ a_2 & \cdots & a_{\mathrm{n}_1} & 0 & 0 & \cdots \\ \vdots & \vdots\vdots\vdots & \vdots & \vdots & \vdots & \vdots \\ a_{\mathrm{n}_1} & 0 & \cdots & 0 & 0 & \cdots \\ \vdots & \vdots\vdots\vdots & \vdots & \vdots & \vdots & \vdots \\ z_1 & \cdots & \cdots & \cdots & z_{\mathrm{n_p}-1} & z_{\mathrm{n_p}} \\ z_2 & \cdots & \cdots & \cdots & z_{\mathrm{n_p}} & 0 \\ \vdots & \vdots\vdots\vdots & \vdots & \vdots & \vdots & \vdots \\ z_{\mathrm{n_p}} & 0 & \cdots & \cdots & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{x}(1) & \tilde{x}(2) & \cdots & \tilde{x}(t) & \cdots \end{bmatrix} = \begin{bmatrix} \tilde{w}(1) & \tilde{w}(2) & \cdots & \tilde{w}(t) & \cdots \\ \tilde{w}(2) & \tilde{w}(3) & \cdots & \tilde{w}(t+1) & \cdots \\ \tilde{w}(3) & \tilde{w}(4) & \cdots & \tilde{w}(t+2) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \\ \tilde{w}(c_{\mathrm{n_p}}) & \tilde{w}(c_{\mathrm{n_p}}+1) & \cdot & \tilde{w}(t+c_{\mathrm{n_p}}-1) & \cdots \end{bmatrix}$$

Summary

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

Summary

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

Completion Lemma:

Given $R(\xi) \in \mathbb{R}^{\mathtt{p} \times \mathtt{w}}[\xi]$ **left prime, compute** $E(\xi) \in \mathbb{R}^{(\mathtt{w}-\mathtt{p}) \times \mathtt{w}}[\xi]$ **:**

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

Summary

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

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

Complete $a(\xi) \mapsto E_a(\xi)$**, compute the 'error'** $\tilde{e} = E_a(\sigma)\tilde{w}$**, replace** $\tilde{w}$ **by** $\tilde{e}$**, and proceed recursively, until 'presistency of excitation'**

Summary

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

Requires computing p vectors in kernel of Hankel matrix of the 'errors'. This error time series decreases in dimension at each step.

Can be executed using numerical LA.
Adapted to approximate computations.
Extensions to T finite, etc.

Note the crucial role played by the module structure!

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