STATE CONSTRUCTION in SYSID

Jan C. Willems

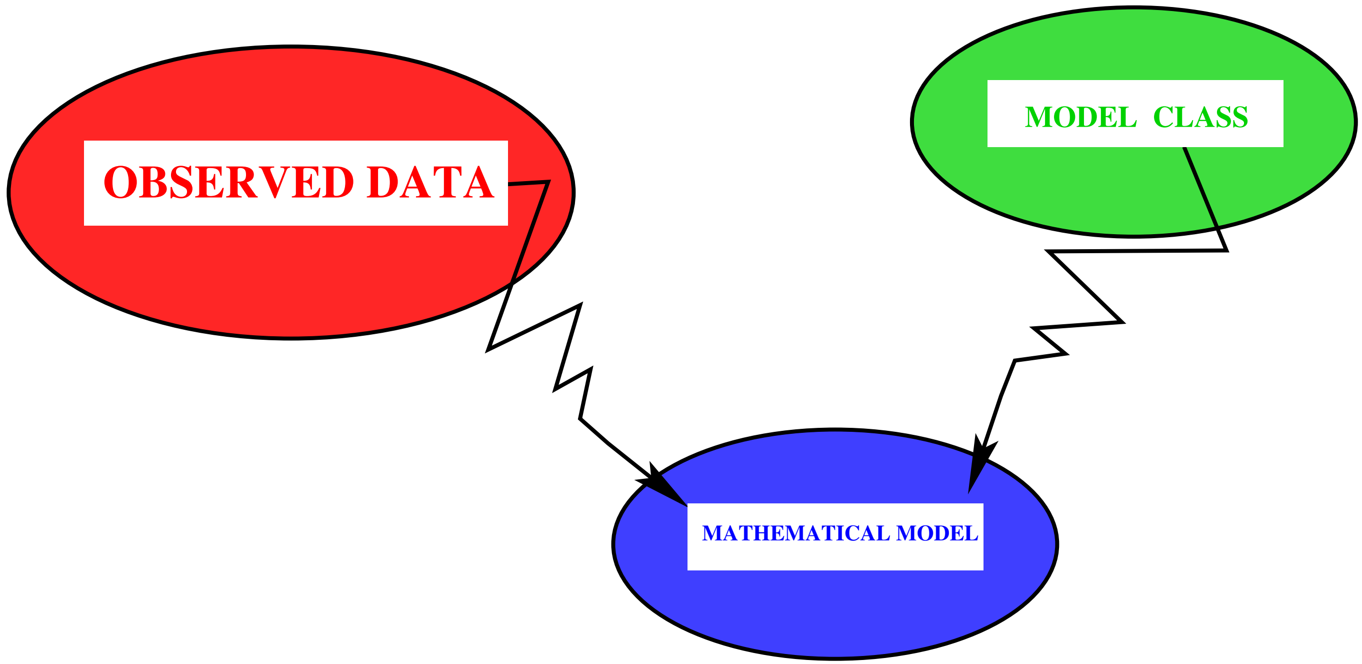
K.U. Leuven, Belgium

SYSID 2006, Newcastle, Australia **March 29, 2006**

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

Problem

SYSID



SYSID

Data: an 'observed' vector time-series

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

$w(t) \in \mathbb{R}^{\mathtt{w}}$

T finite, infinite, or $T \to \infty$

$$\Downarrow$$

A **dynamical model** from a **model class**, e.g. a difference equation

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

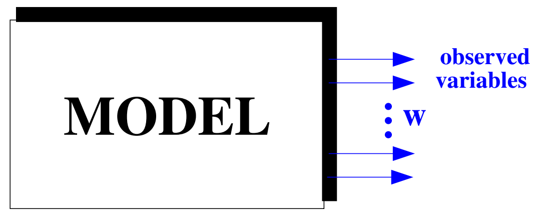
OR

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

(PEM, EIV, etc.)

SYSID

‘deterministic’ ID



Model class:

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

SYSID algorithm:

$$\tilde{w}(1), \tilde{w}(2), \ldots, \tilde{w}(T) \;\mapsto\; \hat{R}_0, \hat{R}_1, \ldots, \hat{R}_{\hat{L}}$$

$\exists$ **always an i/o partition** $w = \Pi \begin{bmatrix} u \\ y \end{bmatrix}$, Π **a permutation.**

Case of interest

Assumptions:

- **Data:**

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

- **Deterministic SYSID**
- **I/O partition known if advantageous**
- **Exact modeling with an eye towards approximations**

Equivalent representations of the model class

The model class $\mathfrak{L}^{\mathtt{w}}$

Our model class is an exceedingly familiar one: $\mathfrak{L}^{\mathtt{w}}$.

$\mathfrak{B} \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$ 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 $\mathfrak{L}^{\mathtt{w}}$

$\mathfrak{B} \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$ 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'**
- **$\exists$ matrices $R_0, R_1, \ldots, R_L$ such that $\mathfrak{B}$: all w that satisfy**

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

In the obvious polynomial matrix notation

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

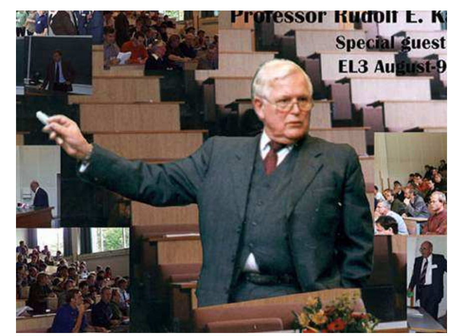
- **Including input/output partition**

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

The model class $\mathfrak{L}^{\mathtt{w}}$

$\mathfrak{B} \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$ 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
  $$\sigma x = Ax + Bu, \; y = Cx + Du, \quad w \cong \begin{bmatrix} u \\ y \end{bmatrix}$$



The lag

$\mathtt{L} : \mathfrak{L}^{\mathtt{w}} \rightarrow \mathbb{Z}_+,$

$\mathtt{L}(\mathfrak{B}) =$ **smallest** L **such that there is a kernel representation:**

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

Polynomial matrix in

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

has $\mathrm{degree}(R) \leq \mathtt{L}$**.**

The MPUM

ID principle: associate with $\tilde{w}(1), \tilde{w}(2), \ldots, \tilde{w}(t), \cdots$

$$\Downarrow$$

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

Exact definition: tomorrow —

today think of the MPUM as the system that produced the data

under persistency of excitation

From data to model to state

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

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

$$\begin{aligned} \boldsymbol{x}(t+1) &= A\boldsymbol{x}(t) + B\boldsymbol{u}(t), \\ \boldsymbol{y}(t) &= C\boldsymbol{x}(t) + D\boldsymbol{u}(t), \qquad \boldsymbol{w}(t) \cong \begin{bmatrix} \boldsymbol{u}(t) \\ \boldsymbol{y}(t) \end{bmatrix} \end{aligned}$$

and compute the (unique) state trajectory

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

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 the data $\tilde{w}$, then this equation provides a way of

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

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

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

This idea yields a very attractive SYSID procedure:

- **Truncation** at suff. large t, **missing data** : cancel columns
- **Model reduce** using SVD e.a. by lowering the row dim. of

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

- **Solve for** $\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$ **using Least Squares**

$\leadsto$ what has come to be known as ‘subspace ID’ .

From data to state

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

How does this work?

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

$$\Downarrow$$

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

This is a very nice system theoretic question.

Note that classical realization theory is a special case: data is impulse response.

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

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

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

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

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

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

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

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

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

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

We give 3 (related) algorithms.

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

$$
\left[\begin{array}{c} \mathcal{H}_- \\ \hline \mathcal{H}_+ \end{array}\right] = \left[\begin{array}{cccc}
\tilde{w}(1) & \cdots & \tilde{w}(t) & \cdots \\
\tilde{w}(2) & \cdots & \tilde{w}(t+1) & \cdots \\
\vdots & \vdots & \vdots & \vdots \\
\tilde{w}(\Delta) & \cdots & \tilde{w}(t+\Delta-1) & \cdots \\
\hline
\tilde{w}(\Delta+1) & \cdots & \tilde{w}(t+\Delta) & \cdots \\
\tilde{w}(\Delta+2) & \cdots & \tilde{w}(t+\Delta+1) & \cdots \\
\vdots & \vdots & \vdots & \vdots \\
\tilde{w}(2\Delta) & \cdots & \tilde{w}(t+2\Delta-1) & \cdots
\end{array}\right]
\begin{array}{c} \uparrow \\ \uparrow \\ \uparrow \\ \text{PAST} \\ \hline \text{FUTURE} \\ \downarrow \\ \downarrow \\ \downarrow \end{array}
$$

$\tilde{w} \mapsto \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$ **by past/future intersection**

$$\left[\begin{array}{c} \mathcal{H}_- \\ \hline \mathcal{H}_+ \end{array}\right] = \left[\begin{array}{cccc} \tilde{w}(1) & \cdots & \tilde{w}(t) & \cdots \\ \tilde{w}(2) & \cdots & \tilde{w}(t+1) & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{w}(\Delta) & \cdots & \tilde{w}(t+\Delta-1) & \cdots \\ \hline \tilde{w}(\Delta+1) & \cdots & \tilde{w}(t+\Delta) & \cdots \\ \tilde{w}(\Delta+2) & \cdots & \tilde{w}(t+\Delta+1) & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{w}(2\Delta) & \cdots & \tilde{w}(t+2\Delta-1) & \cdots \end{array}\right] \begin{array}{c} \uparrow \\ \uparrow \\ \uparrow \\ \text{PAST} \\ \hline \text{FUTURE} \\ \downarrow \\ \downarrow \\ \downarrow \end{array}$$

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

$$\left[\begin{array}{cccccc} \tilde{x}(\Delta+1) & \tilde{x}(\Delta+2) & \cdots & \tilde{x}(t+\Delta) & \cdots \end{array}\right] \longleftarrow \boxed{\text{PRESENT STATE}}$$

State = what is common between past and future.
Numerical implementation $\rightsquigarrow$ subspace ID

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

Solve for G

$$\left[\begin{array}{ccc} \tilde{w}(1) & \cdots & \tilde{w}(T-2\Delta+1) \\ \vdots & \vdots & \vdots \\ \tilde{w}(\Delta) & \cdots & \tilde{w}(T-\Delta) \\ \hline \tilde{u}(\Delta+1) & \cdots & \tilde{u}(T-\Delta+1) \\ \vdots & \vdots & \vdots \\ \tilde{u}(2\Delta) & \cdots & \tilde{u}(T) \end{array}\right] G = \left[\begin{array}{ccc} \tilde{w}(1) & \cdots & \tilde{w}(T-2\Delta+1) \\ \vdots & \vdots & \vdots \\ \tilde{w}(\Delta) & \cdots & \tilde{w}(T-\Delta) \\ \hline 0 & \cdots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \cdots & 0 \end{array}\right]$$

$$\left[\begin{array}{ccc} \tilde{y}(\Delta+1) & \cdots & \tilde{y}(T-\Delta+1) \\ \vdots & \vdots & \vdots \\ \tilde{y}(2\Delta) & \cdots & \tilde{y}(T) \end{array}\right] G = \left[\begin{array}{ccc} \tilde{x}(\Delta+1) & \cdots & \tilde{x}(T-\Delta+1) \end{array}\right]$$

Computes $\tilde{x}$!

$\cong$ ‘oblique projection

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

These algorithms do not make use of the Hankel structure.

Recent development: uses the Hankel structure, together with shift-and-cut state construction algorithm.

$\tilde{w} \mapsto \begin{bmatrix} A & B \\ C & D \end{bmatrix}$ via left annihilators

Implementation. Compute 'the' left annihilators of $\mathcal{H}$:

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

$\tilde{w} \mapsto \begin{bmatrix} A & B \\ C & D \end{bmatrix}$ via left annihilators

Implementation. Compute 'the' left annihilators of $\mathcal{H}$:

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

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

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

$$\tilde{w} \mapsto \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

It actually suffices to compute a set of generators for the $\mathbb{R}[\xi]$-module generated by the left kernel.

Open question:

Construct a balanced state trajectory directly from the data.

Conclusions

Conclusions

- **Subspace ID: data $\Rightarrow$ state trajectory $\Rightarrow$ $\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$.**
- **Copes well with approximation, model reduction.**
- **We have reviewed 3 algorithms:**
  1. **past/future intersection**
  2. **oblique projection**
  3. **cut-and-shift: most attractive;**
     uses Hankel structure & module structure of left kernel.

Tomorrow: how to compute the left annihilators of $\mathcal{H}$ recursively...

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

Thank you