The Behavioral Approach to Systems Theory

Paolo Rapisarda, Un. of Southampton, U.K.
&
Jan C. Willems, K.U. Leuven, Belgium

MTNS 2006
Kyoto, Japan, July 24–28, 2006

Lecture 6: System Identification

Lecturer: Jan C. Willems

Issues to be discussed

- **Remarks on deterministic versus stochastic system identification.**

Issues to be discussed

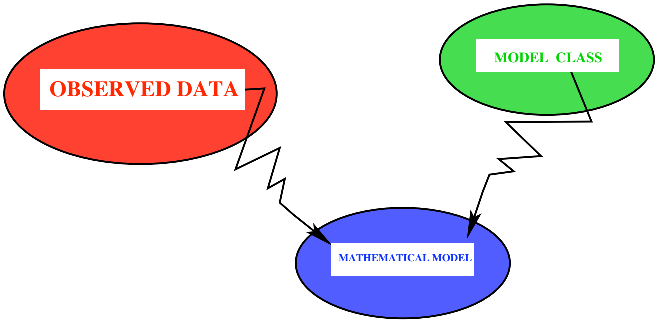
- **Remarks on deterministic versus stochastic system identification.**
- **Deterministic SYSID via the notion of the most powerful unfalsified model (MPUM)**

Issues to be discussed

- **Remarks on deterministic versus stochastic system identification.**
- **Deterministic SYSID via the notion of the most powerful unfalsified model (MPUM)**
- **What is subspace identification?**
- **Algorithms for state construction**
  - **by past/future intersection**
  - **(by oblique projection)**
  - **by recursive annihilator computation**

General Introduction

SYSID



Basic difficulties:
trade-off between overfitting and predictability
learning essential features / rejecting non-essential ones

SYSID

Data: an 'observed' vector time-series

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

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

$$\Downarrow$$

A dynamical model from a model class, e.g. a LTIDS

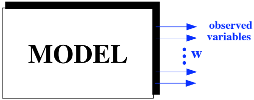
$$R_0 w(t) + R_1 w(t+1) + \cdots + R_L w(t+L) = 0$$

or

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

SYSID

'deterministic' ID



Model class:

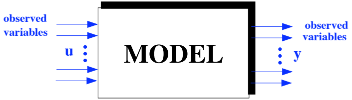
$$R_0 w(t) + R_1 w(t+1) + \cdots + R_L w(t+L) = 0$$

SYSID algorithm:

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

SYSID

'deterministic' ID: I/O form



Model class (with i/o partition):

$$P_0 y(t) + \cdots + P_L y(t+L) = Q_0 u(t) + \cdots + Q_L u(t+L),$$

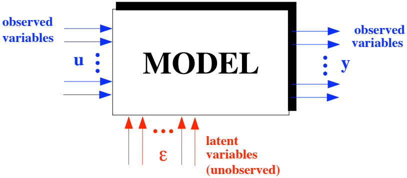
$$w = \Pi \begin{bmatrix} u \\ y \end{bmatrix}, \Pi \text{ permutation }, P(\xi)^{-1} Q(\xi) \text{ proper}$$

SYSID algorithm:

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

SYSID

ID with unobserved latent inputs



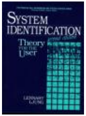
Model class:

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

$$P_0 y(t) + \cdots + P_L y(t+L) = Q_0 u(t) + \cdots + Q_L u(t+L) + M_0 \varepsilon(t) + \cdots + M_L \varepsilon(t+L)$$

SYSID algorithm (e.g. PEM):

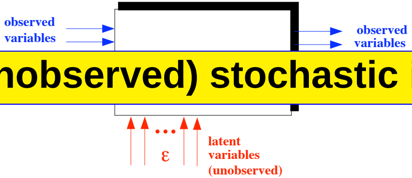
$$\boxed{\tilde{w}(1), \tilde{w}(2), \ldots, \tilde{w}(T)} \mapsto \boxed{\left(\hat{R}(\xi), \hat{M}(\xi)\right)}$$



Usual assumption: w, ε **stochastic.**

SYSID

ID with unobserved latent inputs



Why (unobserved) stochastic inputs?

Model class:

Why stochastics?

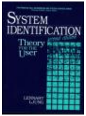
$$R_0 w(t) + R_1 w(t+1) + \cdots + R_L w(t+L)$$
$$= M_0\varepsilon(t) + M_1\varepsilon(t+1) + \cdots + M_L\varepsilon(t+L)$$

$$P_0 y(t) + \cdots + P_L y(t+L)$$
$$= Q_0 u(t) + \cdots + M_L\varepsilon(t+L)$$

Is this physics?

SYSID algorithm (e.g. PEM):

$$\tilde{w}(1), \tilde{w}(2), \ldots, \tilde{w}(T) \mapsto (\hat{R}(\xi), \hat{M}(\xi))$$



Usual assumption: w, ε **stochastic.**

SYSID

Assumptions:

- **Data:**

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

SYSID

Assumptions:

- **Data:**

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

- **Deterministic SYSID**

SYSID

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**
- **Exact modeling with an eye towards approximation**

SYSID

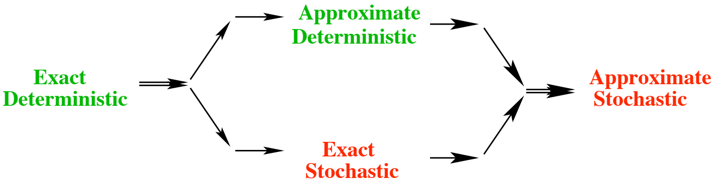
Assumptions:

- **Data:**

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

- **Deterministic SYSID**
- **Exact modeling with an eye towards approximation**

From the simple to the complex!



The MPUM

The exact deterministic SYSID principle

Most Powerful & Unfalsified

- **A model:= a subset $\mathfrak{B} \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$, the 'behavior'**
  A family of (vector) time series

Most Powerful & Unfalsified

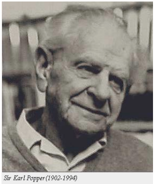
- A model:= a subset $\mathfrak{B} \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$, the 'behavior'
- $\mathfrak{B}$ is unfalsified by $\tilde{w}$ $:\Leftrightarrow \tilde{w} \in \mathfrak{B}$

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

Most Powerful & Unfalsified

- A model:= a subset $\mathfrak{B} \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$, the **'behavior'**
- $\mathfrak{B}$ is unfalsified by $\tilde{w}$ $:\Leftrightarrow \tilde{w} \in \mathfrak{B}$
- $\mathfrak{B}_1$ is more powerful than $\mathfrak{B}_2$ $:\Leftrightarrow \mathfrak{B}_1 \subset \mathfrak{B}_2$

Every model is prohibition.
The more a model forbids, the better it is.



Karl Popper
(1902-1994)

Most Powerful & Unfalsified

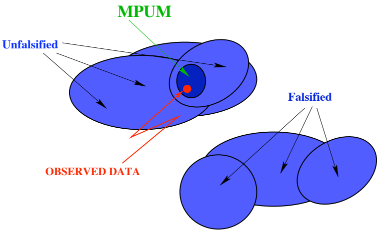
- A **model**:= a subset $\mathfrak{B} \subseteq (\mathbb{R}^{w})^{\mathbb{N}}$, the **'behavior'**
- $\mathfrak{B}$ is **unfalsified by $\tilde{w}$** $:\Leftrightarrow \tilde{w} \in \mathfrak{B}$
- **$\mathfrak{B}_1$ is more powerful than $\mathfrak{B}_2$** $:\Leftrightarrow \mathfrak{B}_1 \subset \mathfrak{B}_2$
- A **model class:** a family, $\mathbb{B}$, of models, e.g. $\mathfrak{L}^{w}$.

Most Powerful & Unfalsified

- A **model**:= a subset $\mathfrak{B} \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$, the **'behavior'**
- $\mathfrak{B}$ is **unfalsified by $\tilde{w}$** $:\Leftrightarrow \tilde{w} \in \mathfrak{B}$
- **$\mathfrak{B}_1$ is more powerful than $\mathfrak{B}_2$** $:\Leftrightarrow \mathfrak{B}_1 \subset \mathfrak{B}_2$
- A **model class:** a family, $\mathbb{B}$, of models, e.g. $\mathfrak{L}^{\mathtt{w}}$.
- The **MPUM** **'most powerful unfalsified model'** in $\mathbb{B}$ for $\tilde{w}$, denoted $\mathfrak{B}^*_{\tilde{w}}$:
  1. $\mathfrak{B}^*_{\tilde{w}} \in \mathbb{B}$
  2. $\tilde{w} \in \mathfrak{B}^*_{\tilde{w}}$
  3. $\mathfrak{B} \in \mathbb{B}$ and $\tilde{w} \in \mathfrak{B}$ $\Rightarrow \mathfrak{B}^*_{\tilde{w}} \subseteq \mathfrak{B}$

Most Powerful & Unfalsified

- A **model**:= a subset $\mathfrak{B} \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$, the **'behavior'**
- $\mathfrak{B}$ is **unfalsified by $\tilde{w}$** $:\Leftrightarrow \tilde{w} \in \mathfrak{B}$
- **$\mathfrak{B}_1$ is more powerful than $\mathfrak{B}_2$** $:\Leftrightarrow \mathfrak{B}_1 \subset \mathfrak{B}_2$
- A **model class:** a family, $\mathbb{B}$, of models, e.g. $\mathfrak{L}^{\mathtt{w}}$.
- The **MPUM** **'most powerful unfalsified model'** in $\mathbb{B}$ for $\tilde{w}$, denoted $\mathfrak{B}^*_{\tilde{w}}$:
  1. $\mathfrak{B}^*_{\tilde{w}} \in \mathbb{B}$
  2. $\tilde{w} \in \mathfrak{B}^*_{\tilde{w}}$
  3. $\mathfrak{B} \in \mathbb{B}$ and $\tilde{w} \in \mathfrak{B}$ $\Rightarrow \mathfrak{B}^*_{\tilde{w}} \subseteq \mathfrak{B}$



Most Powerful & Unfalsified

- A **model**:= a subset $\mathfrak{B} \subseteq (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$, the **'behavior'**
- $\mathfrak{B}$ is **unfalsified by $\tilde{w}$** $:\Leftrightarrow \tilde{w} \in \mathfrak{B}$
- **$\mathfrak{B}_1$ is more powerful than $\mathfrak{B}_2$** $:\Leftrightarrow \mathfrak{B}_1 \subset \mathfrak{B}_2$
- A **model class:** a family, $\mathbb{B}$, of models, e.g. $\mathfrak{L}^{\mathtt{w}}$.
- The **MPUM** **'most powerful unfalsified model'** in $\mathbb{B}$ for $\tilde{w}$, denoted $\mathfrak{B}^*_{\tilde{w}}$:
  1. $\mathfrak{B}^*_{\tilde{w}} \in \mathbb{B}$
  2. $\tilde{w} \in \mathfrak{B}^*_{\tilde{w}}$
  3. $\mathfrak{B} \in \mathbb{B}$ and $\tilde{w} \in \mathfrak{B}$
     $\Rightarrow \mathfrak{B}^*_{\tilde{w}} \subseteq \mathfrak{B}$
- **Given $\tilde{w}$ and $\mathbb{B}$, does $\mathfrak{B}^*_{\tilde{w}}$ exist?**
- 'Exact' SYSID: Construct algorithms $\tilde{w} \mapsto \mathfrak{B}^*_{\tilde{w}}$

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

Exceedingly familiar: 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

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'**
- $R(\sigma)w = 0$
- $P(\sigma)y = Q(\sigma)u, \quad w \cong \begin{bmatrix} u \\ y \end{bmatrix}$

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

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

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'**
- $R(\sigma)w = 0$
- $P(\sigma)y = Q(\sigma)u$, $\quad w \cong \begin{bmatrix} u \\ y \end{bmatrix}$
- $\sigma x = Ax + Bu,\ y = Cx + Du,$ $\quad w \cong \begin{bmatrix} u \\ y \end{bmatrix}$
- $\exists$ **a matrix of rational functions** G **such that** $\mathfrak{B}$ = **sol'ns of**

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

without LOG strictly proper
with LOG (stabilizability) proper stable rational.

The lag

$\mathtt{L} : \mathfrak{L}^{\mathtt{w}} \to \mathbb{Z}_+$,
$\mathtt{L}(\mathfrak{B}) =$ smallest L such that there is a kernel repr.:

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

Polynomial matrix in $R(\sigma)w = 0$ has $\text{degree}(R) \leq \mathtt{L}$.

One the important 'integer invariants': maps : $\mathfrak{L}^{\mathtt{w}} \to \mathbb{Z}_+$,.

Others:
$\mathtt{m}, \mathtt{p}, \mathtt{n}$: number of inputs, outputs, states,
$\nu_1, \cdots, \nu_{\mathtt{p}}$: (kernel) lag indices, observability indices,
$\kappa_1, \cdots, \kappa_{\mathtt{m}}$: (image) lag indices, controllability indices.

The MPUM in $\mathfrak{L}^{\mathrm{w}}$

Theorem: For infinite obs. interval, $T = \infty$ (our case),

the MPUM for $\tilde{w}$ in $\mathfrak{L}^{\mathrm{w}}$ exists.

In fact,

$$\mathfrak{B}^*_{\tilde{w}} = \mathrm{span}(\{\tilde{w}, \sigma\tilde{w}, \sigma^2\tilde{w}, \ldots\})^{\mathrm{closure}}$$

Same is true for model class $\mathfrak{L}^{\mathrm{w}}$ with lag $\leq \ell$.

We are looking for effective computational algorithms to go from $\tilde{w}$ to (a representation of) $\mathfrak{B}^*_{\tilde{w}}$,

e.g., a kernel representation $\leadsto$ the corresponding R;
e.g. a generating set of annihilators
e.g., the matrices $\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$ of an i/s/o representation of $\mathfrak{B}^*_{\tilde{w}}$.

The Hankel matrix of the data

The key role is played by the 'Hankel matrix' of the data

Hermann Hankel
1839-1873

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

Persistency of excitation

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

Question: Is it possible to recover the system that generated the data? 'Identifiability'.

Persistency of excitation

Assume that

1. $\tilde{w} \in \mathfrak{B}_{[1,T]}$
2. $\mathfrak{B} \in \mathfrak{L}^{w}$
3. $\mathfrak{B}$ **controllable**
4. $\Delta > \mathrm{L}(\mathfrak{B})$
5. $\tilde{w} = \begin{bmatrix} \tilde{u} \\ \tilde{y} \end{bmatrix}$, $\tilde{u}$ **persistently exciting** **of order** $\Delta + \mathrm{n}(\mathfrak{B})$

This means that

$$\begin{bmatrix} \tilde{u}(1) & \tilde{u}(2) & \cdots & \tilde{u}(T - \Delta - \mathrm{n}(\mathfrak{B}) - 1) \\ \tilde{w}(2) & \tilde{w}(3) & \cdots & \tilde{w}(T - \Delta - \mathrm{n}(\mathfrak{B})) \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{w}(\Delta + \mathrm{n}(\mathfrak{B})) & \tilde{w}(\Delta + \mathrm{n}(\mathfrak{B}) + 1) & \cdots & \tilde{w}(T) \end{bmatrix}$$

has full row rank.

Persistency of excitation

Assume that

1. $\tilde{w} \in \mathfrak{B}_{[1,T]}$
2. $\mathfrak{B} \in \mathfrak{L}^{\mathtt{w}}$
3. $\mathfrak{B}$ **controllable**
4. $\Delta > \mathrm{L}(\mathfrak{B})$
5. $\tilde{w} = \begin{bmatrix} \tilde{u} \\ \tilde{y} \end{bmatrix}$, $\tilde{u}$ **persistently exciting** **of order** $\Delta + \mathrm{n}(\mathfrak{B})$

Then the left kernel of the data Hankel matrix

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

is a set of generators of $\mathfrak{N}_{\mathfrak{B}}^{\mathbb{R}[\xi]} \Leftrightarrow$ **its column span** $= \mathfrak{B}_{[1,L]}$

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 Identification

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

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 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, ...
- SVD model reduce by first lowering 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.

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

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

Has been generalized to stochastic systems.

From data to state

$\tilde{w} \longmapsto \tilde{x}$

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 (> the lag of the MPUM).

Identify somehow, directly from the data, state map

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

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

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

or

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

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

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

There are many algorithms. We discuss two.

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

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

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

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

The **intersection** of the span of the rows of $\mathcal{H}_-$ with those of $\mathcal{H}_+$ = 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.

Existing algorithms (N4SID, MOESP,...): past/future part.

How can we compute this intersection?

$$\begin{bmatrix} a_1 \\ a_2 \end{bmatrix}^\top \begin{bmatrix} M_1 \\ M_2 \end{bmatrix}^\top = 0 \quad \Rightarrow a_1^\top M_1 = -a_2^\top M_2 \text{ : common linear combinations.}$$

$$0 = \left[\begin{array}{c} a_1 \\ \hline a_2 \end{array}\right]^\top \left[\begin{array}{ccccc} \tilde{w}(1) & \tilde{w}(2) & \cdots & \tilde{w}(t) & \cdots \\ \tilde{w}(2) & \tilde{w}(3) & \cdots & \tilde{w}(t+1) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \\ \tilde{w}(\Delta) & \tilde{w}(\Delta+1) & \cdots & \tilde{w}(t+\Delta-1) & \cdots \\ \hline \tilde{w}(\Delta+1) & \tilde{w}(\Delta+2) & \cdots & \tilde{w}(t+\Delta) & \cdots \\ \tilde{w}(\Delta+2) & \tilde{w}(\Delta+3) & \cdots & \tilde{w}(t+\Delta+1) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \\ \tilde{w}(2\Delta) & \tilde{w}(2\Delta+1) & \cdots & \tilde{w}(t+2\Delta-1) & \cdots \end{array}\right]$$

How can we compute this intersection?

$$\begin{bmatrix} a_1 \\ a_2 \end{bmatrix}^\top \begin{bmatrix} M_1 \\ M_2 \end{bmatrix}^\top = 0 \quad \Rightarrow a_1^\top M_1 = -a_2^\top M_2 : \text{common linear combinations.}$$

$$0 = \left[\begin{array}{c} a_1 \\ \hline a_2 \end{array}\right]^\top \left[\begin{array}{ccccc} \tilde{w}(1) & \tilde{w}(2) & \cdots & \tilde{w}(t) & \cdots \\ \tilde{w}(2) & \tilde{w}(3) & \cdots & \tilde{w}(t+1) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \\ \tilde{w}(\Delta) & \tilde{w}(\Delta+1) & \cdots & \tilde{w}(t+\Delta-1) & \cdots \\ \hline \tilde{w}(\Delta+1) & \tilde{w}(\Delta+2) & \cdots & \tilde{w}(t+\Delta) & \cdots \\ \tilde{w}(\Delta+2) & \tilde{w}(\Delta+3) & \cdots & \tilde{w}(t+\Delta+1) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \\ \tilde{w}(2\Delta) & \tilde{w}(2\Delta+1) & \cdots & \tilde{w}(t+2\Delta-1) & \cdots \end{array}\right]$$

Hankel structure $\Rightarrow$ the left kernel of the whole matrix can be computed from the kernel of the upper part $\rightsquigarrow$ following algorithm

$\tilde{w} \longmapsto \tilde{x}$

Compute 'the' left annihilators of the Hankel matrix:

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

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

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'

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

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

Computing the kernel of a Hankel matrix

Hankel kernel

Leads to the problem:

Compute the left kernel of a (block) Hankel matrix

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

Hankel kernel

Identify each left annihilator with a vector polynomial

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

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

Hankel kernel

This kernel is closed under addition

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

Hankel kernel

and under shifting

$$[a_0 \ a_1 \ \cdots \ a_\Delta \quad 0 \quad 0 \ \cdots]$$
$$\Downarrow$$
$$[\ 0 \ a_0 \quad \cdots \ a_{\Delta-1} \ a_\Delta \ 0 \ \cdots]$$

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

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

$$\Rightarrow \quad a(\xi) + b(\xi) \ \text{ and } \ \xi a(\xi) \ \in \text{ left kernel.}$$

Hankel kernel

$\boldsymbol{a}(\xi) = \boldsymbol{a}_0 + \boldsymbol{a}_1\xi + \cdots + \boldsymbol{a}_\Delta\xi^\Delta$ $\in$ **left kernel**
$\boldsymbol{b}(\xi) = \boldsymbol{b}_0 + \boldsymbol{b}_1\xi + \cdots + \boldsymbol{b}_\Delta\xi^\Delta$ $\in$ **left kernel**

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

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

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

Left kernel is in a real sense always finite dim. (dim. p $\leq$ w).

State from 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 from generators

Generators

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

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 from 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.

Recursive computation

Recursive computation

Suppose we found a left annihilator of

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

Recursive computation

Suppose we found a left annihilator of

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

Use this to simplify finding other left annihilators of

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

Completion lemma

Key question: **Given $\mathfrak{B} \in \mathfrak{L}^{\mathtt{w}}$, $\exists$ a complement?**

i.e. $\mathfrak{B}' \in \mathfrak{L}^{\mathtt{w}}$ such that $\mathfrak{B} \oplus \mathfrak{B}' = (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$?

Meaning in terms of kernel or image representations?

Completion lemma

Key question: **Given $\mathfrak{B} \in \mathfrak{L}^{\mathtt{w}}$, $\exists$ a complement?**

i.e. $\mathfrak{B}' \in \mathfrak{L}^{\mathtt{w}}$ such that $\mathfrak{B} \oplus \mathfrak{B}' = (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$?

Meaning in terms of kernel or image representations?

There exists a complement iff $\mathfrak{B}$ is controllable.

Completion lemma

Key question: **Given $\mathfrak{B} \in \mathfrak{L}^{\mathtt{w}}$, $\exists$ a complement?**

i.e. $\mathfrak{B}' \in \mathfrak{L}^{\mathtt{w}}$ such that $\mathfrak{B} \oplus \mathfrak{B}' = (\mathbb{R}^{\mathtt{w}})^{\mathbb{N}}$?

Meaning in terms of kernel or image representations?

There exists a complement iff $\mathfrak{B}$ is controllable.

Given R, complete with R' such that $\begin{bmatrix} R \\ R' \end{bmatrix}$ is unimodular.

Given M, complete with M', s.t. $\begin{bmatrix} M & M' \end{bmatrix}$ is unimodular.

Given basis of rat. annihilators, find complementary basis.

Recursive computation

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 $=$ non-zero constant, inv. as a pol. matrix.

Recursive computation

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 $=$ non-zero constant, inv. 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

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

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

Bézout

Recursive computation

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

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

Recursive computation

Assume

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

Recursive computation

Assume

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

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

Recursive computation

Assume

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

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

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

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

Recursive computation

Assume

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

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

Compute

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

$\tilde{e}$ 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...

Recursive computation

Assume

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

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

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, assuming MPUM contr., left kernel by computing p times a left kernel vector.

Recursion can be combined with the state computation.

Summary

Summary

- **Approximation in SYSID is cloesr to the physics that stochasticity**
- **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

- Approximation in SYSID is cloesr to the physics that stochasticity
- Subspace ID
- State construction
  - Past/future intersection
  - Oblique projection
  - Generators left kernel of Hankel + cut-and-shift
- Central pbm: computation of generators of left kernel of Hankel matrix

Summary

- **Approximation in SYSID is cloesr to the physics that stochasticity**
- **Subspace ID**
- **State construction**
  - **Past/future intersection**
  - **Oblique projection**
  - **Generators left kernel of Hankel + cut-and-shift**
- **Central pbm: computation of generators of left kernel of Hankel matrix**
- **Computation can be carried out recursively in the case that the MPUM is controllable.**

Summary

- **Approximation in SYSID is cloesr to the physics that stochasticity**
- **Subspace ID**
- **State construction**
  - **Past/future intersection**
  - **Oblique projection**
  - **Generators left kernel of Hankel + cut-and-shift**
- **Central pbm: computation of generators of left kernel of Hankel matrix**
- **Computation can be carried out recursively in the case that the MPUM is controllable.**
- **Key step: completion lemma. Given $\mathfrak{B} \in \mathfrak{L}^{\mathtt{w}}$, find $\mathfrak{B}' \in \mathfrak{L}^{\mathtt{w}}$ such that $\mathfrak{B} \oplus \mathfrak{B}' =$ everything.**