STATE CONSTRUCTION
and
SUBSPACE IDENTIFICATION

Jan C. Willems

University of Leuven, Belgium

Kyoto University , May 26, 2003 **University of Osaka, June 6, 2003**

STATE SPACE SYSTEMS

THEME

How do we formalize the memory of a dynamical system?
When is a variable a state variable?
How do state equations look like?

STATE SPACE SYSTEMS

THEME

How do we formalize the memory of a dynamical system?
When is a variable a state variable?
How do state equations look like?

How are state equations constructed, algorithmically ?

THE NOTION OF STATE

A state system :=

A latent variable system in which the latent variable has a special property.

THE NOTION OF STATE

A state system :=

A latent variable system in which the latent variable has a special property.

The latent variable system

$$\Sigma_X = (\mathbb{T}, \mathbb{W}, \mathbb{X}, \mathfrak{B}_{\text{full}})$$

is said to be a *state system* if

$$(w_1, x_1), (w_2, x_2) \in \mathfrak{B}_{\text{full}}, t_0 \in \mathbb{T}, \quad \text{and} \quad x_1(t_0) = x_2(t_0)$$

imply

$$(w_1, x_1) \underset{t_0}{\wedge} (w_2, x_2) \in \mathfrak{B}_{\text{full}}.$$

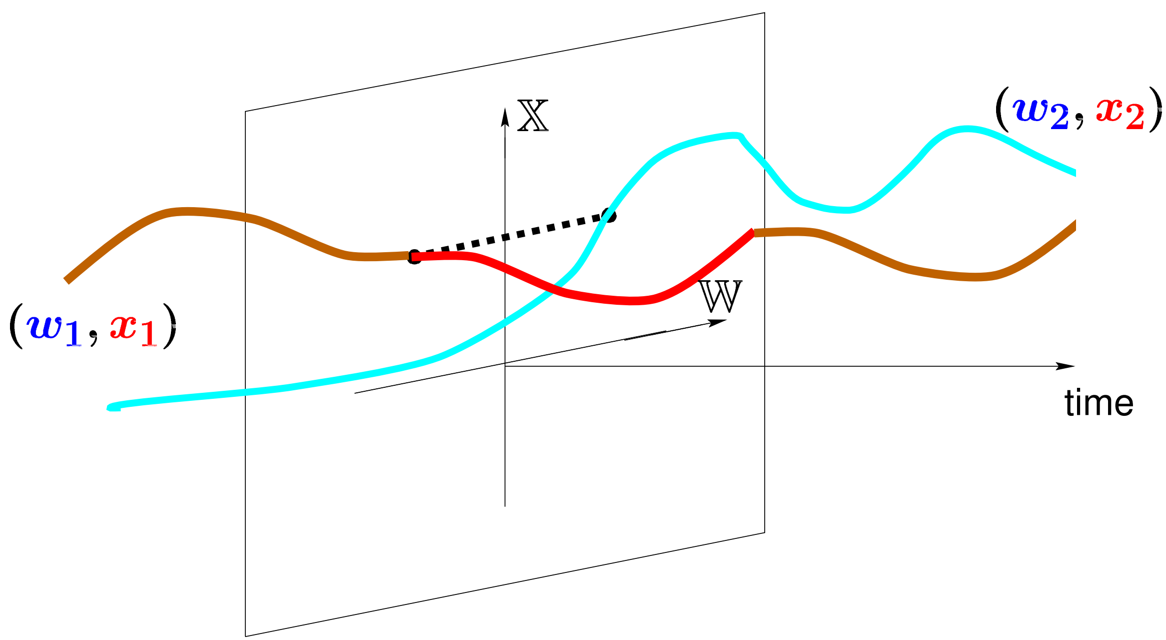
$\underset{t_0}{\wedge}$ **denotes *concatenation* at t_0, defined as**

$$f_1 \underset{t_0}{\wedge} f_2(t) := \begin{cases} f_1(t) \text{ for } t < t_0 \\ f_2(t) \text{ for } t \geq t_0 \end{cases}$$

$\underset{t_0}{\wedge}$ denotes ***concatenation*** at t_0, defined as

$$f_1 \underset{t_0}{\wedge} f_2(t) := \begin{cases} f_1(t) \text{ for } t < t_0 \\ f_2(t) \text{ for } t \geq t_0 \end{cases}$$

In pictures:



This definition is the implementation of the idea:

> ***The state at time t, $x(t)$, contains all the information (about (w, x)!) that is relevant for the future behavior.***

The state = the memory.

$\cong$ Markovianity!

This definition is the implementation of the idea:

> ***The state at time t, $x(t)$, contains all the information (about (w, x)!) that is relevant for the future behavior.***

The state = the memory.

> **The past and the future are 'independent', conditioned on (given) the present state.**

$\cong$ **Markovianity!**

Examples of state systems:

Examples of state systems:

1. Discrete-time systems.

Examples of state systems:

1. Discrete-time systems.

A latent variable system described by a difference equation that is

first order in the latent variable x, and

zero-th order in the manifest variable w:

$$\boxed{F(x(t+1), x(t), w(t), t) = 0.}$$

Examples of state systems:

1. **Discrete-time systems.**

2. **Continuous-time systems.**

Examples of state systems:

1. **Discrete-time systems.**

2. **Continuous-time systems.**

A latent variable system described by a differential equation that is

first order **in the latent variable x, and**

zero-th order **in the manifest variable w:**

$$F\left(\frac{d}{dt}x(t), x(t), w(t), t\right) = 0.$$

Examples of state systems:

1. **Discrete-time systems.**

2. **Continuous-time systems.**

In particular, the ubiquitous

$$\frac{d}{dt}x(t) = f(x(t), u(t)), \quad y(t) = h(x(t), u(t));$$

$$w(t) = (u(t), y(t)).$$

Examples of state systems:

1. Discrete-time systems.

2. Continuous-time systems.

3. Automata.

Examples of state systems:

1. Discrete-time systems.

2. Continuous-time systems.

3. Automata.

4. Trellis diagrams.

Examples of state systems:

1. Discrete-time systems.

2. Continuous-time systems.

3. Automata.

4. Trellis diagrams.

5. QM: $\frac{d}{dt}\psi = i\hbar H(\psi)\,, \quad p = |\psi|^2;$

ψ = the 'wave function';
$p(x,t)$ = the 'probability' density of the particle's position.

The wave function = latent, state, the observables = manifest??

For discrete time state systems $\rightsquigarrow$

<u>Theorem</u>: The latent variable system

$$\Sigma_X = (\mathbb{Z}, \mathbb{W}, \mathbb{X}, \mathfrak{B}_{\text{full}})$$

is a state system <u>*if*</u> (and *only if,* provided the system is 'complete')
$\mathfrak{B}_{\text{full}}$ admits a representation as a difference equation that is
first order in the latent variable x, and
zero-th order in the manifest variable w:

$$\boxed{F(x(t+1), x(t), w(t), t) = 0.}$$

STATE FOR DIFFERENTIAL SYSTEMS

Here we meet the notorious

$\mathfrak{C}^\infty$-difficulty:

concatenation and $\mathfrak{C}^\infty$ don't mix.

STATE FOR DIFFERENTIAL SYSTEMS

Here we meet the notorious

$\mathfrak{C}^\infty$-difficulty:

concatenation and $\mathfrak{C}^\infty$ don't mix.

We hence modify the state axiom to: The latent variable system[†] $\Sigma_X = (\mathbb{R}, \mathbb{R}^{\mathtt{w}}, \mathbb{R}^{\mathtt{n}}, \mathfrak{B}_{\text{full}}), \mathfrak{B}_{\text{full}} \in \mathfrak{L}^{\mathtt{w}+\mathtt{n}}$ is said to be a **state system** if

$$(w_1, x_1), (w_2, x_2) \in \mathfrak{B}_{\text{full}}, t_0 \in \mathbb{T}, \quad \text{and} \quad x_1(t_0) = x_2(t_0)$$

imply $\quad (w_1, x_1) \underset{t_0}{\wedge} (w_2, x_2) \in \mathfrak{B}_{\text{full}}^{\text{closure}}.$

'Closure' w.r.t., e.g., the $\mathfrak{L}^{\text{loc}}$-topology.

† $\mathfrak{L}^{\mathtt{w}}$:= the differential systems with $\mathtt{w}$ variables, see lecture 2.

STATE FOR DIFFERENTIAL SYSTEMS

Here we meet the notorious

$\mathfrak{C}^\infty$**-difficulty:**

concatenation and $\mathfrak{C}^\infty$ don't mix.

We hence modify the state axiom to: The **latent variable system**†
$\Sigma_X = (\mathbb{R}, \mathbb{R}^{\mathtt{w}}, \mathbb{R}^{\mathtt{n}}, \mathfrak{B}_{\text{full}}), \mathfrak{B}_{\text{full}} \in \mathfrak{L}^{\mathtt{w}+\mathtt{n}}$ is said to be a
state system if

$$(w_1, x_1), (w_2, x_2) \in \mathfrak{B}_{\text{full}}, t_0 \in \mathbb{T}, \quad \text{and} \quad x_1(t_0) = x_2(t_0)$$

imply $\quad (w_1, x_1) \underset{t_0}{\wedge} (w_2, x_2) \in \mathfrak{B}_{\text{full}}^{\text{closure}}.$

'Closure' w.r.t., e.g., the $\mathfrak{L}^{\text{loc}}$-topology.

Equivalent: if $(w_1, x_1) \underset{t_0}{\wedge} (w_2, x_2)$ is a **weak sol'n** of the ODE.

DESCRIPTOR SYSTEMS

Theorem: The latent variable system $(\mathbb{R}, \mathbb{R}^{\mathtt{w}}, \mathbb{R}^{\mathtt{n}}, \mathfrak{B}_{\mathrm{full}})$ with $\mathfrak{B}_{\mathrm{full}} \in \mathfrak{L}^{\mathtt{w}+\mathtt{n}}$ is a state system *if and only if* $\mathfrak{B}_{\mathrm{full}}$ admits a kernel representation that is

***first order* in the latent variable x, and**

***zero-th order* in the manifest variable w.**

DESCRIPTOR SYSTEMS

Theorem: The latent variable system $(\mathbb{R}, \mathbb{R}^{\mathtt{w}}, \mathbb{R}^{\mathtt{n}}, \mathfrak{B}_{\text{full}})$ with $\mathfrak{B}_{\text{full}} \in \mathfrak{L}^{\mathtt{w}+\mathtt{n}}$ is a state system *if and only if* $\mathfrak{B}_{\text{full}}$ admits a kernel representation that is

***first order* in the latent variable x, and**

***zero-th order* in the manifest variable w.**

In other words, iff there exist matrices $E, F, G \in \mathbb{R}^{\bullet\times\bullet}$ such that this kernel representation takes the form of a *descriptor system:*

$$\boxed{E\frac{d}{dt}x + Fx + Gw = 0.}$$

MINIMALITY

We can consider two types of minimality of state representations:

1. **Minimality of the number of *equations***
2. **Minimality of the number of *state variables***

MINIMALITY

We can consider two types of minimality of state representations:

1. Minimality of the number of *equations*
2. Minimality of the number of *state variables*

We discuss mainly the second one.

Definition: The state system $(\mathbb{R}, \mathbb{R}^{\mathtt{w}}, \mathbb{R}^{\mathtt{n}}, \mathfrak{B}_{\text{full}})$ with $\mathfrak{B}_{\text{full}} \in \mathfrak{L}^{\mathtt{w}+\mathtt{n}}$ is said to be ***state-minimal*** if, whenever $(\mathbb{R}, \mathbb{R}^{\mathtt{w}}, \mathbb{R}^{\mathtt{n}'}, \mathfrak{B}'_{\text{full}})$ with $\mathfrak{B}'_{\text{full}} \in \mathfrak{L}^{\mathtt{w}+\mathtt{n}'}$ is another state system with the same manifest behavior, there holds

$$\mathtt{n} \leq \mathtt{n}'.$$

Trimness

One more definition...

$\mathfrak{B} \in \mathfrak{L}^{\mathtt{w}}$ is said to be *trim* if, $\forall$ $\mathrm{w}_0 \in \mathbb{R}^{\mathtt{w}}$, $\exists w \in \mathfrak{B}$ such that $w(0) = \mathrm{w}_0$. The state system $(\mathbb{R}, \mathbb{R}^{\mathtt{w}}, \mathbb{R}^{\mathtt{n}}, \mathfrak{B}_{\mathrm{full}})$ with $\mathfrak{B}_{\mathrm{full}} \in \mathfrak{L}^{\mathtt{w}+\mathtt{n}}$ is said to be *state-trim* if, $\forall$ $\mathrm{x}_0 \in \mathbb{R}^{\mathtt{n}}$, $\exists (w, x) \in \mathfrak{B}_{\mathrm{full}}$ such that $x(0) = \mathrm{x}_0$.

Trimness

One more definition...

$\mathfrak{B} \in \mathfrak{L}^{\mathtt{w}}$ is said to be ***trim*** if, $\forall$ $\mathrm{w}_0 \in \mathbb{R}^{\mathtt{w}}$, $\exists w \in \mathfrak{B}$ such that $w(0) = \mathrm{w}_0$. The state system $(\mathbb{R}, \mathbb{R}^{\mathtt{w}}, \mathbb{R}^{\mathtt{n}}, \mathfrak{B}_{\mathrm{full}})$ with $\mathfrak{B}_{\mathrm{full}} \in \mathfrak{L}^{\mathtt{w}+\mathtt{n}}$ is said to be ***state-trim*** if, $\forall$ $\mathrm{x}_0 \in \mathbb{R}^{\mathtt{n}}$, $\exists (w, x) \in \mathfrak{B}_{\mathrm{full}}$ such that $x(0) = \mathrm{x}_0$.

Theorem:

The state system $(\mathbb{R}, \mathbb{R}^{\mathtt{w}}, \mathbb{R}^{\mathtt{n}}, \mathfrak{B}_{\mathrm{full}})$ with $\mathfrak{B}_{\mathrm{full}} \in \mathfrak{L}^{\mathtt{w}+\mathtt{n}}$ is ***state-minimal*** iff it is ***state trim*** and the state x is ***observable*** from w.

Trimness

One more definition...

$\mathfrak{B} \in \mathfrak{L}^{\mathtt{w}}$ is said to be *trim* if, $\forall$ $\mathrm{w}_0 \in \mathbb{R}^{\mathtt{w}}$, $\exists w \in \mathfrak{B}$ such that $w(0) = \mathrm{w}_0$. The state system $(\mathbb{R}, \mathbb{R}^{\mathtt{w}}, \mathbb{R}^{\mathtt{n}}, \mathfrak{B}_{\mathrm{full}})$ with $\mathfrak{B}_{\mathrm{full}} \in \mathfrak{L}^{\mathtt{w}+\mathtt{n}}$ is said to be *state-trim* if, $\forall$ $\mathrm{x}_0 \in \mathbb{R}^{\mathtt{n}}$, $\exists (w, x) \in \mathfrak{B}_{\mathrm{full}}$ such that $x(0) = \mathrm{x}_0$.

Theorem:
The state system $(\mathbb{R}, \mathbb{R}^{\mathtt{w}}, \mathbb{R}^{\mathtt{n}}, \mathfrak{B}_{\mathrm{full}})$ with $\mathfrak{B}_{\mathrm{full}} \in \mathfrak{L}^{\mathtt{w}+\mathtt{n}}$ is *state-minimal* iff it is *state trim* and the state x is *observable* from w.

Observability :$\Leftrightarrow$ x can be deduced from w.
I.e., $\exists$ $X \in \mathbb{R}^{\mathtt{n} \times \mathtt{w}}[\xi]$ such that

$$(w, x) \in \mathfrak{B}_{\mathrm{full}} \Leftrightarrow x = X\left(\tfrac{d}{dt}\right)w.$$

Trimness

One more definition...

$\mathfrak{B} \in \mathfrak{L}^{\mathtt{w}}$ is said to be ***trim*** if, $\forall$ $\mathrm{w}_0 \in \mathbb{R}^{\mathtt{w}}$, $\exists w \in \mathfrak{B}$ such that $w(0) = \mathrm{w}_0$. The state system $(\mathbb{R}, \mathbb{R}^{\mathtt{w}}, \mathbb{R}^{\mathtt{n}}, \mathfrak{B}_{\mathrm{full}})$ with $\mathfrak{B}_{\mathrm{full}} \in \mathfrak{L}^{\mathtt{w}+\mathtt{n}}$ is said to be ***state-trim*** if, $\forall$ $\mathrm{x}_0 \in \mathbb{R}^{\mathtt{n}}$, $\exists (w, x) \in \mathfrak{B}_{\mathrm{full}}$ such that $x(0) = \mathrm{x}_0$.

Theorem:

The state system $(\mathbb{R}, \mathbb{R}^{\mathtt{w}}, \mathbb{R}^{\mathtt{n}}, \mathfrak{B}_{\mathrm{full}})$ with $\mathfrak{B}_{\mathrm{full}} \in \mathfrak{L}^{\mathtt{w}+\mathtt{n}}$ is *state-minimal* iff it is *state trim* and the state x is *observable* from w.

State-minimal $\Leftrightarrow$ state-trim and state-observable.

Further results

1. State isomorphism theorem.

Assume $(\mathbb{R}, \mathbb{R}^{\mathtt{w}}, \mathbb{R}^{\mathtt{n}}, \mathfrak{B}_{\text{full}})$ and $(\mathbb{R}, \mathbb{R}^{\mathtt{w}}, \mathbb{R}^{\mathtt{n}}, \mathfrak{B}'_{\text{full}})$, $\mathfrak{B}_{\text{full}}, \mathfrak{B}'_{\text{full}} \in \mathfrak{L}^{\mathtt{w}+\mathtt{n}}$ both state-minimal, same manifest behavior

$\Rightarrow$ there exists a nonsingular $S \in \mathbb{R}^{\mathtt{n}\times\mathtt{n}}$ such that

$$[(w, x) \in \mathfrak{B}_{\text{full}} \text{ and } (w, x') \in \mathfrak{B}'_{\text{full}}] \Leftrightarrow [x' = Sx].$$

The minimal state representation is unique up to a choice of the basis in the state space.

Further results

1. **State isomorphism theorem.**

2. **Controllability.**

The manifest behavior is controllable iff there exists a state representation of it whose full behavior is controllable.

Further results

1. **State isomorphism theorem.**

2. **Controllability.**

The manifest behavior is controllable iff there exists a state-minimal state representation of it that is state-controllable.

Further results

1. **State isomorphism theorem.**

2. **Controllability.**

3. **Descriptor systems.**

$\exists$ algorithms acting on E, F, G in a descriptor representation to verify its state-minimality, its equation minimality, both combined.

Further results

1. State isomorphism theorem.

2. Controllability.

3. Descriptor systems.

$$E\frac{d}{dt}x + Fx + Gw = 0 \quad \text{and} \quad E'\frac{d}{dt}x' + F'x' + G'w = 0$$

are two minimal (state- and equation-minimal) representations of the same manifest behavior iff there exist nonsingular matrices $T, S \in \mathbb{R}^{\bullet \times \bullet}$ such that

$$E' = TES, F' = TES, G' = TG.$$

Further results

1. **State isomorphism theorem.**

2. **Controllability.**

3. **Descriptor systems.**

4. **Notation:**

$\mathtt{n}(\mathfrak{B})$:= the dimension of the **minimal** state associated with $\mathfrak{B}$.

All 'classical' results remain valid, except, (fortunately!) the celebrated (non-)equivalence:

state-minimality $\Leftrightarrow$ state-observability + state-controllability.

Non-controllable systems are very 'real' and they allow state-minimal (non-controllable) state representation.

Input/State/Output Systems

Finally...

It is possible to combine the input/output partition and the state representation, leading to the ubiquitous:

$$\boxed{\frac{d}{dt}x = Ax + Bu, \quad y = Cx + Du, \quad w = (u, y).}$$

u **is input := free,**
y **is output := bound by** u**,**
x **is state := 'splitting'.**

Theorem: Let $\mathfrak{B} \in \mathfrak{L}^{\mathtt{w}}$.

There exists a componentwise partition $w = (u, y)$, with $\dim(u) = \mathtt{m}(\mathfrak{B}), \dim(y) = \mathtt{p}(\mathfrak{B})$, and matrices

$$A \in \mathbb{R}^{\mathtt{n}(\mathfrak{B}) \times \mathtt{n}(\mathfrak{B})}, B \in^{\mathtt{n}(\mathfrak{B}) \times \mathtt{m}(\mathfrak{B})}, C \in \mathbb{R}^{\mathtt{p}(\mathfrak{B}) \times \mathtt{n}(\mathfrak{B})}, D \in \mathbb{R}^{\mathtt{p}(\mathfrak{B}) \times \mathtt{m}(\mathfrak{B})}$$

such that

$$\frac{d}{dt}x = Ax + Bu, \quad y = Cx + Du,$$

is a minimal (equation- and state-minimal) state repr'ion of $\mathfrak{B}$.

$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$ **is minimal (state + eq'n minimal)**

$\Leftrightarrow$ **it is state-minimal**

$\Leftrightarrow$ **it is state-observable**

$$\Leftrightarrow \text{rank}\left(\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{\dim(A)-1} \end{bmatrix}\right) = \dim(A).$$

$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$ **is state controllable (usual Kalman def'n)**

$$\Leftrightarrow \mathrm{rank}(\left[\, B \;\; AB \;\; \ldots \;\; A^{\dim(A)-1}B \,\right]) = \dim(A).$$

$\Rightarrow$ **the manifest behavior is controllable.**

$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$ **is state controllable (usual Kalman def'n)**

$$\Leftrightarrow \operatorname{rank}([\, B \;\; AB \;\; \ldots \;\; A^{\dim(A)-1}B \,]) = \dim(A).$$

$\Rightarrow$ **the manifest behavior is controllable.**

If $\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$ **is minimal (i.e., observable) then**

state controllable iff manifest behavior controllable.

$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$ **is state controllable (usual Kalman def'n)**

$$\Leftrightarrow \text{rank}([\,B \;\; AB \;\; \ldots \;\; A^{\dim(A)-1}B\,]) = \dim(A).$$

$\Rightarrow$ **the manifest behavior is controllable.**

If $\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$ **is minimal (i.e., observable) then**

state controllable iff manifest behavior controllable.

Watch out:

minimality of $\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$ $\Leftarrow$ **but** $\not\Rightarrow$ **controllable & observable.**

STATE CONSTRUCTION

!! Given a dynamical system $\Sigma = (\mathbb{T}, \mathbb{W}, \mathfrak{B})$
find a state representation $\Sigma_X = (\mathbb{T}, \mathbb{W}, \mathbb{X}, \mathfrak{B}_{\text{full}})$
for it !!

STATE CONSTRUCTION

!! Given a dynamical system $\Sigma = (\mathbb{T}, \mathbb{W}, \mathfrak{B})$
find a state representation $\Sigma_X = (\mathbb{T}, \mathbb{W}, \mathbb{X}, \mathfrak{B}_{\text{full}})$
for it !!

This problem is a jewel that has emerged in systems theory (and in computer science) in the sixties. It has ramifications in the theory of stochastic processes, in computer science and formal language theory, (more recently) model simplification, etc.

STATE CONSTRUCTION

!! Given a dynamical system $\Sigma = (\mathbb{T}, \mathbb{W}, \mathfrak{B})$ find a state representation $\Sigma_X = (\mathbb{T}, \mathbb{W}, \mathbb{X}, \mathfrak{B}_{\text{full}})$ for it !!

This problem is a jewel that has emerged in systems theory (and in computer science) in the sixties. It has ramifications in the theory of stochastic processes, in computer science and formal language theory, (more recently) model simplification, etc.

We assume henceforth $\mathbb{T} = \mathbb{R}$ or $\mathbb{Z}$ and $\Sigma = (\mathbb{T}, \mathbb{W}, \mathfrak{B})$ is time-invariant.

There are 2 main aspects:

1. **Abstract state construction: construct the state space from $\mathfrak{B}$**

There are 2 main aspects:

1. Abstract state construction: construct the state space from $\mathfrak{B}$
2. Find algorithms that pass from a behavioral equation representation of the manifest behavior $\mathfrak{B}$ to a specification of $\mathbb{X}$ and a behavioral equation representation of $\mathfrak{B}_{\mathrm{full}}$.

Useful general properties

A state system $\Sigma_{\mathbb{X}} = (\mathbb{T}, \mathbb{W}, \mathbb{X}, \mathfrak{B}_{\text{full}})$ is said to be *irreducible*

:⇔ [($f : \mathbb{X} \to \mathbb{X}'$, $\Sigma_{\mathbb{X}} = (\mathbb{T}, \mathbb{W}, \mathbb{X}', \mathfrak{B}'_{\text{full}})$ such that $\mathfrak{B}'_{\text{full}} = \{(w, f \circ x) \mid (x, w) \in \mathfrak{B}_{\text{full}}\}$ is a state system), $\Rightarrow$ (f is a bijection)].

Useful general properties

A state system $\Sigma_{\mathbb{X}} = (\mathbb{T}, \mathbb{W}, \mathbb{X}, \mathfrak{B}_{\text{full}})$ is said to be *irreducible*

$:\Leftrightarrow$ [($f : \mathbb{X} \to \mathbb{X}'$, $\Sigma_{\mathbb{X}} = (\mathbb{T}, \mathbb{W}, \mathbb{X}', \mathfrak{B}'_{\text{full}})$ such that $\mathfrak{B}'_{\text{full}} = \{(w, f \circ x) \mid (x, w) \in \mathfrak{B}_{\text{full}}\}$ is a state system),
$\Rightarrow$ (f is a bijection)].

Two state systems $\Sigma_{\mathbb{X}} = (\mathbb{T}, \mathbb{W}, \mathbb{X}, \mathfrak{B}_{\text{full}})$ and $\Sigma'_{\mathbb{X}} = (\mathbb{T}, \mathbb{W}, \mathbb{X}', \mathfrak{B}'_{\text{full}})$ are said to be *equivalent*

if there exists a bijection $f : \mathbb{X} \to \mathbb{X}'$ such that

$$[(w, x) \in \mathfrak{B}_{\text{full}}] \Leftrightarrow [(w, f \circ x) \in \mathfrak{B}'_{\text{full}}].$$

Useful general properties

A state system $\Sigma_{\mathbb{X}} = (\mathbb{T}, \mathbb{W}, \mathbb{X}, \mathfrak{B}_{\text{full}})$ is said to be *irreducible*

:$\Leftrightarrow$ [($f : \mathbb{X} \to \mathbb{X}'$, $\Sigma_{\mathbb{X}} = (\mathbb{T}, \mathbb{W}, \mathbb{X}', \mathfrak{B}'_{\text{full}})$ such that $\mathfrak{B}'_{\text{full}} = \{(w, f \circ x) \mid (x, w) \in \mathfrak{B}_{\text{full}}\}$ is a state system), $\Rightarrow$ (f is a bijection)].

Two state systems $\Sigma_{\mathbb{X}} = (\mathbb{T}, \mathbb{W}, \mathbb{X}, \mathfrak{B}_{\text{full}})$ and $\Sigma'_{\mathbb{X}} = (\mathbb{T}, \mathbb{W}, \mathbb{X}', \mathfrak{B}'_{\text{full}})$ are said to be *equivalent*

if there exists a bijection $f : \mathbb{X} \to \mathbb{X}'$ such that

$$[(w, x) \in \mathfrak{B}_{\text{full}}] \Leftrightarrow [(w, f \circ x) \in \mathfrak{B}'_{\text{full}}].$$

Clearly equivalent state systems represent the same manifest behavior.

Abstract state construction

We now address the question: Given $\Sigma = (\mathbb{T}, \mathbb{W}, \mathfrak{B})$, find a (irreducible) state space representation $\Sigma_X = (\mathbb{T}, \mathbb{W}, \mathbb{X}, \mathfrak{B}_{\text{full}})$ for it.

Abstract state construction

We now address the question: Given $\Sigma = (\mathbb{T}, \mathbb{W}, \mathfrak{B})$, find a (irreducible) state space representation $\Sigma_X = (\mathbb{T}, \mathbb{W}, \mathbb{X}, \mathfrak{B}_{\text{full}})$ for it.

The crucial idea is to define the state space!

When do two trajectories bring the system in the same state?

When is what is stored in the memory by the two trajectories the same?

Abstract state construction

We now address the question: Given $\Sigma = (\mathbb{T}, \mathbb{W}, \mathfrak{B})$, find a (irreducible) state space representation $\Sigma_X = (\mathbb{T}, \mathbb{W}, \mathbb{X}, \mathfrak{B}_{\text{full}})$ for it.

The crucial idea is to define the state space!

When do two trajectories bring the system in the same state?

When is what is stored in the memory by the two trajectories the same?

When the trajectories can be continued in the same way!

This idea of constructing an equivalence relation on the manifest behavior $\mathfrak{B}$, sometimes called ‘Nerode equivalence’, leads to our past canonical state representation.

This idea of constructing an equivalence relation on the manifest behavior $\mathfrak{B}$, sometimes called ‘Nerode equivalence’, leads to our past canonical state representation.

In the past canonical state construction, define the equivalence relation R_- by

$$[w_1 R_- w_2] :\Leftrightarrow [(w_1 \underset{0}{\wedge} w \in \mathfrak{B}) \Leftrightarrow (w_1 \underset{0}{\wedge} w \in \mathfrak{B})].$$

This idea of constructing an equivalence relation on the manifest behavior $\mathfrak{B}$, sometimes called ‘Nerode equivalence’, leads to our past canonical state representation.

In the past canonical state construction, define the equivalence relation R_- by

$$[w_1 R_- w_2] :\Leftrightarrow [(w_1 \underset{0}{\wedge} w \in \mathfrak{B}) \Leftrightarrow (w_1 \underset{0}{\wedge} w \in \mathfrak{B})].$$

Our concept of state being basically ‘time-symmetric’
$\Rightarrow$ future canonical state representation.

This idea of constructing an equivalence relation on the manifest behavior $\mathfrak{B}$, sometimes called ‘Nerode equivalence’, leads to our past canonical state representation.

In the past canonical state construction, define the equivalence relation R_- by

$$[w_1 R_- w_2] :\Leftrightarrow [(w_1 \underset{0}{\wedge} w \in \mathfrak{B}) \Leftrightarrow (w_1 \underset{0}{\wedge} w \in \mathfrak{B})].$$

In the future canonical state construction, define the equivalence relation R_+ by

$$[w_1 R_+ w_2] :\Leftrightarrow [(w \underset{0}{\wedge} w_1 \in \mathfrak{B}) \Leftrightarrow (w \underset{0}{\wedge} w_2 \in \mathfrak{B})].$$

Finally, combine both to the two-sided canonical state representation.

In the two-sided canonical state construction, define the equivalence rel. $R_\pm$ by

$$[w_1 R_\pm w_2] :\Leftrightarrow [((w_1 \underset{0}{\wedge} w \in \mathfrak{B}) \Leftrightarrow (w_1 \underset{0}{\wedge} w \in \mathfrak{B})) \\ \wedge ((w \underset{0}{\wedge} w_1 \in \mathfrak{B}) \Leftrightarrow (w \underset{0}{\wedge} w_2 \in \mathfrak{B}))].$$

Finally, combine both to the **two-sided canonical** state representation.

In the **two-sided canonical state construction,** define the equivalence rel. $R_\pm$ by

$$[w_1 R_\pm w_2] :\Leftrightarrow [((w_1 \underset{0}{\wedge} w \in \mathfrak{B}) \Leftrightarrow (w_1 \underset{0}{\wedge} w \in \mathfrak{B})) \\ \wedge ((w \underset{0}{\wedge} w_1 \in \mathfrak{B}) \Leftrightarrow (w \underset{0}{\wedge} w_2 \in \mathfrak{B}))].$$

Obviously,

$$[w_1 R_\pm w_2] \Leftrightarrow [(w_1 R_- w_2) \wedge (w_1 R_+ w_2)].$$

For the past-canonical state construction, define the state space by $\mathbb{X}_- = \mathfrak{B}(\text{mod } R_-)$ and the full behavior by

$$\mathfrak{B}_{\text{full},-} = \{(w, x) \mid (w \in \mathfrak{B}) \wedge (\sigma^t w \in (\sigma^t x)(0)\ \forall t \in \mathbb{T})\}.$$

For the past-canonical state construction, define the state space by $\mathbb{X}_- = \mathfrak{B}(\text{mod } R_-)$ and the full behavior by

$$\mathfrak{B}_{\text{full},-} = \{(w,x) \mid (w \in \mathfrak{B}) \wedge (\sigma^t w \in (\sigma^t x)(0)\ \forall t \in \mathbb{T})\}.$$

For the future-canonical state construction, define the state space by $\mathbb{X}_+ = \mathfrak{B}(\text{mod } R_+)$ and the full behavior by

$$\mathfrak{B}_{\text{full},+} = \{(w,x) \mid (w \in \mathfrak{B}) \wedge (\sigma^t w \in (\sigma^t x)(0)\ \forall t \in \mathbb{T})\}.$$

For the past-canonical state construction, define the state space by $\mathbb{X}_- = \mathfrak{B}(\text{mod } R_-)$ and the full behavior by

$$\mathfrak{B}_{\text{full},-} = \{(w,x) \mid (w \in \mathfrak{B}) \wedge (\sigma^t w \in (\sigma^t x)(0)\ \forall t \in \mathbb{T})\}.$$

For the future-canonical state construction, define the state space by $\mathbb{X}_+ = \mathfrak{B}(\text{mod } R_+)$ and the full behavior by

$$\mathfrak{B}_{\text{full},+} = \{(w,x) \mid (w \in \mathfrak{B}) \wedge (\sigma^t w \in (\sigma^t x)(0)\ \forall t \in \mathbb{T})\}.$$

For the two-sided-canonical state construction, define the state space by $\mathbb{X}_\pm = \mathfrak{B}(\text{mod } R_\pm)$ and the full behavior by

$$\mathfrak{B}_{\text{full},\pm} = \{(w,x) \mid (w \in \mathfrak{B}) \wedge (\sigma^t w \in (\sigma^t x)(0)\ \forall t \in \mathbb{T})\}.$$

The canonical state representations $\Sigma_- := (\mathbb{T}, \mathbb{W}, \mathbb{X}_-, \mathfrak{B}_-)$ and $\Sigma_+ := (\mathbb{T}, \mathbb{W}, \mathbb{X}_+, \mathfrak{B}_+)$ have very good properties.
In particular, they are irreducible.

The canonical state representations $\Sigma_- := (\mathbb{T}, \mathbb{W}, \mathbb{X}_-, \mathfrak{B}_-)$ and $\Sigma_+ := (\mathbb{T}, \mathbb{W}, \mathbb{X}_+, \mathfrak{B}_+)$ have very good properties.
In particular, they are irreducible.

The question when all irreducible state representations of a given system are equivalent has a very nice answer in terms of these canonical representations.

Indeed, the following conditions are equivalent:

1. **All irreducible state representations of a given system $(\mathbb{T}, \mathbb{W}, \mathfrak{B})$ are equivalent.**
2. $(\mathbb{T}, \mathbb{W}, \mathbb{X}_-, \mathfrak{B}_{\text{full},-})$ **and** $(\mathbb{T}, \mathbb{W}, \mathbb{X}_+, \mathfrak{B}_{\text{full},+})$ **are equivalent.**
3. $(\mathbb{T}, \mathbb{W}, \mathbb{X}_-, \mathfrak{B}_{\text{full},\pm})$ **is irreducible.**
4. $(\mathbb{T}, \mathbb{W}, \mathbb{X}_-, \mathfrak{B}_{\text{full},-})$ **and** $(\mathbb{T}, \mathbb{W}, \mathbb{X}_-, \mathfrak{B}_{\text{full},\pm})$ **are equivalent.**
5. $(\mathbb{T}, \mathbb{W}, \mathbb{X}_+, \mathfrak{B}_{\text{full},+})$ **and** $(\mathbb{T}, \mathbb{W}, \mathbb{X}_-, \mathfrak{B}_{\text{full},\pm})$ **are equivalent.**

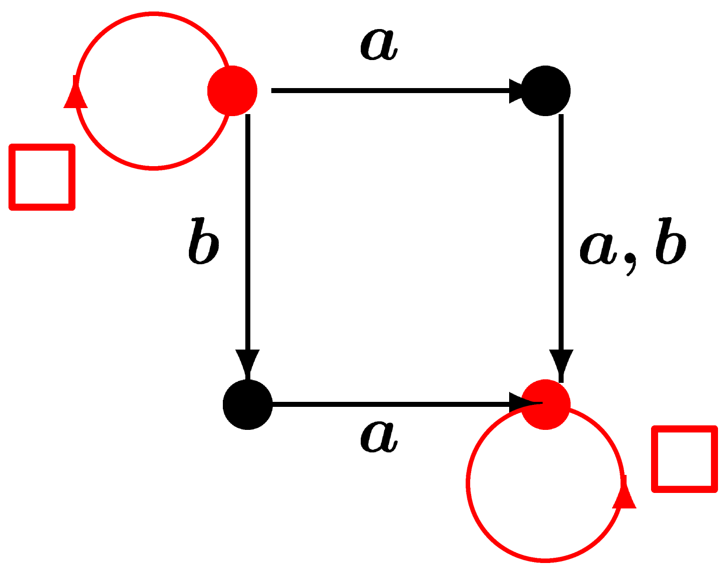
Example

$$\boxed{\mathfrak{L} = \{aa, ab, ba\}.}$$

Example

$$\mathfrak{L} = \{aa, ab, ba\}.$$

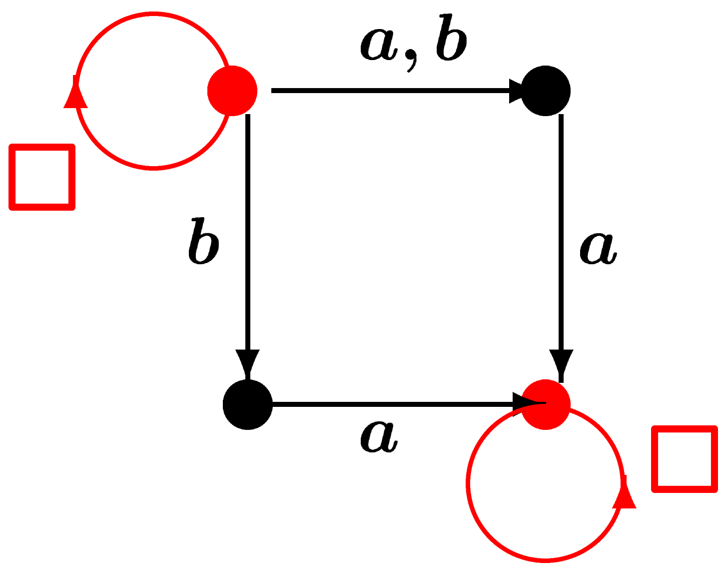
Past canonical state representation:



Example

$$\mathfrak{L} = \{aa, ab, ba\}.$$

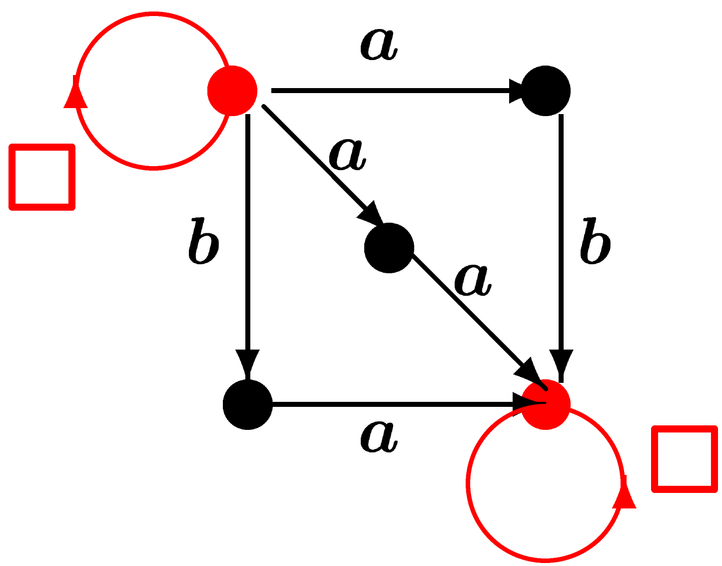
Future canonical state representation:



Example

$$\boxed{\mathfrak{L} = \{aa, ab, ba\}.}$$

Two-sided canonical state representation:



Example

$$\boxed{\mathfrak{L} = \{aa, ab, ba\}.}$$

This example demonstrates that not all irreducible state representations are equivalent.

Example

$$\boxed{\mathfrak{L} = \{aa, ab, ba\}.}$$

This example demonstrates that not all irreducible state representations are equivalent.

Important instances of systems for which all irreducible state representations are equivalent are linear and autonomous systems.

STATE CONSTRUCTION in DIFFERENTIAL SYSTEMS

Given a representation of the manifest behavior $\mathfrak{B} \in \mathfrak{L}^\bullet$, find a (state-minimal) state representation for it.

STATE CONSTRUCTION in DIFFERENTIAL SYSTEMS

Given a representation of the manifest behavior $\mathfrak{B} \in \mathfrak{L}^\bullet$,
find a (state-minimal) state representation for it.

Most logical : latent variable repr'on $\leadsto$ state repr'on.
However, it is most convenient to discuss kernel repr'ons first.

STATE MAPS

Let $X(\xi) \in \mathbb{R}^{\bullet \times \mathtt{w}}[\xi]$. The map $X(\frac{d}{dt})$ is called a ***state map*** for $\mathfrak{B} \in \mathfrak{L}^{\mathtt{w}}$ if the full behavior

$$\mathfrak{B}_{\mathrm{full}} = \{(w, x) \mid w \in \mathfrak{B} \text{ and } x = X(\frac{d}{dt})w\}$$

satisfies the axiom of state. *Minimal* state map: obvious.

STATE MAPS

Let $X(\xi) \in \mathbb{R}^{\bullet \times \mathtt{w}}[\xi]$. The map $X(\frac{d}{dt})$ is called a **state map** for $\mathfrak{B} \in \mathfrak{L}^{\mathtt{w}}$ if the full behavior

$$\mathfrak{B}_{\text{full}} = \{(w, x) \mid w \in \mathfrak{B} \text{ and } x = X(\frac{d}{dt})w\}$$

satisfies the axiom of state. *Minimal* state map: obvious.

In a state-minimal representation, x is always determined by a state map (because of observability), whence (minimal) state maps exist.

Algorithms for State Construction

Problem: **Given a ‘numerical’ specification of a dynamical system, end up with a ‘numerical’ specification of a state model.**

Algorithms for State Construction

<u>**Problem**</u>: **Given a ‘numerical’ specification of a dynamical system, end up with a ‘numerical’ specification of a state model.**

We only consider linear time-invariant differential systems

Algorithms for State Construction

Problem: Given a 'numerical' specification of a dynamical system, end up with a 'numerical' specification of a state model.

- Given the impulse response construct a state model $\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$.

$\leadsto$ the theory around the Hankel matrix.

Algorithms for State Construction

Problem: **Given a 'numerical' specification of a dynamical system, end up with a 'numerical' specification of a state model.**

- **Given the impulse response construct a state model** $\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$.
- **Given the transfer function construct a state model** $\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$.

Algorithms for State Construction

Problem: **Given a 'numerical' specification of a dynamical system, end up with a 'numerical' specification of a state model.**

- **Given the impulse response construct a state model** $\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$.
- **Given the transfer function construct a state model** $\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$.
- **Given a kernel, image, or latent variable representation, construct a (minimal) state model** (E, F, G) **or** $\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$.

Algorithms for State Construction

Problem: **Given a 'numerical' specification of a dynamical system, end up with a 'numerical' specification of a state model.**

- **Given the impulse response construct a state model** $\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$.
- **Given the transfer function construct a state model** $\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$.
- **Given a kernel, image, or latent variable representation, construct a (minimal) state model** (E, F, G) **or** $\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$.
- **Make sure** $\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$ **is in a special (e.g., balanced) form**

Define the *'shift-and-cut'* operator σ on $\mathbb{R}[\xi]$ as follows:

$$\sigma : p_0 + p_1\xi + \cdots + p_{n-1}\xi^{n-1} + p_n\xi^n$$
$$\mapsto \quad p_1 + p_2\xi + \cdots + p_{n-1}\xi^{n-2} + p_n\xi^{n-1}$$

Extend-able in the obvious term-by-term way to $\mathbb{R}^{\bullet\times\bullet}[\xi]$.

Define the ***'shift-and-cut'* operator** σ on $\mathbb{R}[\xi]$ as follows:

$$\sigma : p_0 + p_1\xi + \cdots + p_{n-1}\xi^{n-1} + p_n\xi^n$$
$$\mapsto \quad p_1 + p_2\xi + \cdots + p_{n-1}\xi^{n-2} + p_n\xi^{n-1}$$

Extend-able in the obvious term-by-term way to $\mathbb{R}^{\bullet\times\bullet}[\xi]$.

Repeated use of the cut-and-shift on $P \in \mathbb{R}^{\bullet\times\bullet}[\xi]$ yields the ***'stack' operator*** Σ_P, defined by

$$\Sigma_P := \begin{bmatrix} \sigma(P) \\ \sigma^2(P) \\ \vdots \\ \sigma^{\mathrm{degree}(P)}(P) \end{bmatrix}$$

FROM KERNEL to STATE REPRESENTATION

There is a construction (elegant in its simplicity) of a state map in terms of the cut-and-shift and stack operators!

<u>Theorem</u>: Let $R(\frac{d}{dt})w = 0$ be a kernel representation of $\mathfrak{B} \in \mathfrak{L}^{\mathtt{w}}$. Then $\Sigma_R(\frac{d}{dt})$ is a state map for $\mathfrak{B}$. The resulting state representation

$$\boxed{R(\frac{d}{dt})w = 0\,; \quad x = \Sigma_R(\frac{d}{dt})w}$$

FROM KERNEL to STATE REPRESENTATION

There is a construction (elegant in its simplicity) of a state map in terms of the cut-and-shift and stack operators!

Theorem: Let $R(\frac{d}{dt})w = 0$ be a kernel representation of $\mathfrak{B} \in \mathfrak{L}^{\mathtt{w}}$. Then $\Sigma_R(\frac{d}{dt})$ is a state map for $\mathfrak{B}$. The resulting state representation

$$\boxed{R(\frac{d}{dt})w = 0\,; \quad x = \Sigma_R(\frac{d}{dt})w}$$

Need not be minimal. It is trivially state-observable, but it may not be state-trim. Using Gröbner basis techniques it can be trimmed, leading to a minimal state representation.

SINGLE INPUT - SINGLE OUTPUT SYSTEMS

Apply this to

$$p(\frac{d}{dt})y = q(\frac{d}{dt})u$$

with

$$\begin{aligned} p(\xi) &= p_0 + p_1\xi + \cdots + p_{n-1}\xi^{n-1} + p_n\xi^n, \;\; p_n \neq 0 \\ q(\xi) &= q_0 + q_1\xi + \cdots + q_{n-1}\xi^{n-1} + q_n\xi^n \end{aligned}$$

SINGLE INPUT - SINGLE OUTPUT SYSTEMS

Apply this to

$$p(\frac{d}{dt})y = q(\frac{d}{dt})u$$

with

$$\begin{aligned} p(\xi) &= p_0 + p_1\xi + \cdots + p_{n-1}\xi^{n-1} + p_n\xi^n, \;\; p_n \neq 0 \\ q(\xi) &= q_0 + q_1\xi + \cdots + q_{n-1}\xi^{n-1} + q_n\xi^n \end{aligned}$$

The cut-and-shift and stack operators yield the polynomial matrix

$$\Sigma_R(\xi) = \begin{bmatrix} p_1+\cdots+p_{n-1}\xi^{n-2}+p_n\xi^{n-1} & -q_1-\cdots-q_{n-1}\xi^{n-2}-q_n\xi^{n-1} \\ p_2+\cdots+p_{n-1}\xi^{n-3}+p_n\xi^{n-2} & -q_2-\cdots-q_{n-1}\xi^{n-3}-q_n\xi^{n-2} \\ \vdots & \vdots \\ p_{n-1}+p_n\xi & -q_{n-1}-q_n\xi \\ p_n & -q_n \end{bmatrix}$$

It follows that $x = \Sigma_R(\frac{d}{dt})$ is a state map, in fact, a state minimal one, even if the system is not controllable, i.e., when p and q have a common factor.

To get more convenient minimal state maps, we can take any basis for span of the rows of X.

To get more convenient minimal state maps, we can take any basis for span of the rows of X.

One choice: take the rows of Σ_R in reverse order.

A small calculation shows that this choice of the state variables leads to the so-called *observer canonical form,* the i/s/o representation

$$A = \begin{bmatrix} -p_{n-1}/p_n & 1 & 0 & \cdots & 0 & 0 \\ -p_{n-2}/p_n & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ -p_0/p_n & 0 & 0 & \cdots & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} q_{n-1} - p_{n-1} q_n / p_n \\ q_{n-2} - p_{n-2} q_n / p_n \\ \vdots \\ q_0 - p_0 q_n / p_n \end{bmatrix},$$

$$C = \begin{bmatrix} 1/p_n & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}, \qquad D = \begin{bmatrix} q_n / p_n \end{bmatrix}.$$

Another immediate choice is to pick the state map

$$X(\xi) = \begin{bmatrix} 1 & \star \\ \xi & \star \\ \vdots & \vdots \\ \xi^{n-2} & \star \\ \xi^{n-1} & \star \end{bmatrix}$$

We need to compute the $\star$'s so that the combinations of the rows of Σ_R that yield the first column of X also give the second column.

The second column can be obtained by long hand division of q by p, i.e., by computing the polynomial $b(\xi) \in \mathbb{R}[\xi]$ defined by the equation

$$p(\xi)b(\xi^{-1}) = q(\xi) \quad (\text{ modulo } \xi^{-1}\mathbb{R}[\xi^{-1}]).$$

Then $$X(\xi) = \begin{bmatrix} 1 & b_0 \\ \xi & b_1 + b_0\xi \\ \vdots & \vdots \\ \xi^{n-2} & b_{n-2} + b_{n-3}\xi + \cdots + b_0\xi^{n-2} \\ \xi^{n-1} & b_{n-1} + b_{n-2}\xi + \cdots + b_0\xi^{n-1} \end{bmatrix}.$$

Then $$X(\xi) = \begin{bmatrix} 1 & b_0 \\ \xi & b_1 + b_0\xi \\ \vdots & \vdots \\ \xi^{n-2} & b_{n-2} + b_{n-3}\xi + \cdots + b_0\xi^{n-2} \\ \xi^{n-1} & b_{n-1} + b_{n-2}\xi + \cdots + b_0\xi^{n-1} \end{bmatrix}.$$

This leads to the ***observable canonical form,*** the i/s/o representation

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -\frac{p_0}{p_n} & -\frac{p_1}{p_n} & -\frac{p_2}{p_n} & \cdots & -\frac{p_{n-1}}{p_n} \end{bmatrix}, \quad B = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix},$$

$$C = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} b_0 \end{bmatrix}.$$

FROM IMAGE to STATE REPRESENTATION

<u>Theorem</u>: Let $w = M(\frac{d}{dt})\ell$ be an image representation of $\mathfrak{B} \in \mathfrak{L}^{\mathtt{w}}_{\mathrm{cont}}$, and Σ_M the stack operator induced by M.

Then

$$\boldsymbol{w = M(\frac{d}{dt})\ell\,; \quad x = \Sigma_M(\frac{d}{dt})\ell}$$

is a state representation of $\mathfrak{B}$.

Again, not necessarily minimal.

FROM IMAGE to STATE REPRESENTATION

Theorem: Let $w = M(\frac{d}{dt})\ell$ be an image representation of $\mathfrak{B} \in \mathfrak{L}^{\mathtt{w}}_{\mathrm{cont}}$, and Σ_M the stack operator induced by M.

Then

$$\boxed{w = M(\frac{d}{dt})\ell\,; \quad x = \Sigma_M(\frac{d}{dt})\ell}$$

is a state representation of $\mathfrak{B}$.

Again, not necessarily minimal.

Note: we obtain a state map that acts on ℓ. If $w = M(\frac{d}{dt})\ell$ is not observable, then the state may not be observable, whence not state-minimal.

SINGLE INPUT - SINGLE OUTPUT SYSTEMS

When the system is controllable, and given in image representation by

$$\begin{bmatrix} u \\ y \end{bmatrix} = \begin{bmatrix} p(\frac{d}{dt}) \\ q(\frac{d}{dt}) \end{bmatrix} \ell$$

with

$$\begin{aligned} p(\xi) &= p_0 + p_1\xi + \cdots + p_{n-1}\xi^{n-1} + p_n\xi^n, \;\; p_n \neq 0, \\ q(\xi) &= q_0 + q_1\xi + \cdots + q_{n-1}\xi^{n-1} + q_n\xi^n. \end{aligned}$$

The cut-and-shift and stack operators yield

$$X(\xi) = \begin{bmatrix} p_1 + \cdots + p_{n-1}\xi^{n-2} + p_n\xi^{n-1} \\ q_1 + \cdots + q_{n-1}\xi^{n-2} + q_n\xi^{n-1} \\ p_2 + \cdots + p_{n-1}\xi^{n-3} + p_n\xi^{n-2} \\ q_2 + \cdots + q_{n-1}\xi^{n-3} + q_n\xi^{n-2} \\ \vdots \\ p_{n-1} + p_n\xi \\ q_{n-1}n + q_n\xi \\ q_n \\ p_n \end{bmatrix}.$$

There are again two ready bases for the linear span of the rows of X:

There are again two ready bases for the linear span of the rows of X:

$$\begin{bmatrix} p_{\mathrm{n}} \\ p_{\mathrm{n}-1}+p_{\mathrm{n}}\xi \\ \vdots \\ p_2+\cdots+p_{\mathrm{n}-1}\xi^{\mathrm{n}-3}+p_{\mathrm{n}}\xi^{\mathrm{n}-2} \\ p_1+\cdots+p_{\mathrm{n}-1}\xi^{\mathrm{n}-2}+p_{\mathrm{n}}\xi^{\mathrm{n}-1} \end{bmatrix} \text{ and } \begin{bmatrix} 1 \\ \xi \\ \vdots \\ \xi^{\mathrm{n}-2} \\ \xi^{\mathrm{n}-1} \end{bmatrix}.$$

There are again two ready bases for the linear span of the rows of X:

The first choice leads to the ***controllable canonical form***

$$
\begin{aligned}
A &= \begin{bmatrix} -p_{n-1}/p_n & 1 & 0 & \cdots & 0 & 0 \\ -p_{n-2}/p_n & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ -p_0/p_n & 0 & 0 & \cdots & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}, \\
C &= \begin{bmatrix} b_1 & b_2 & \cdots & b_{n-1} & b_n \end{bmatrix}, \qquad D = \begin{bmatrix} b_0 \end{bmatrix}.
\end{aligned}
$$

There are again two ready bases for the linear span of the rows of X:

The second choice leads to the **controller canonical form**

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -\frac{p_0}{p_n} & -\frac{p_1}{p_n} & -\frac{p_2}{p_n} & \cdots & -\frac{p_{n-1}}{p_n} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \frac{1}{p_n} \end{bmatrix},$$

$$C = \begin{bmatrix} q_0 - p_0 \frac{q_n}{p_n} & q_1 - p_1 \frac{q_n}{p_n} & \cdots & q_{n-1} - p_{n-1} \frac{q_n}{p_n} \end{bmatrix}, D = \begin{bmatrix} q_n \end{bmatrix}.$$

FROM LATENT VARIABLE to STATE REPRESENTATION

Consider the latent variable system $\Sigma_X = (\mathbb{R}, \mathbb{R}^{\mathtt{w}_1+\mathtt{w}_2}, \mathbb{R}^{\mathtt{n}}, \mathfrak{B}_{\mathrm{full}})$ with $\mathfrak{B}_{\mathrm{full}} \in \mathfrak{L}^{\mathtt{w}_1+\mathtt{w}_2+\mathtt{n}}$. Eliminate $w_2 \rightsquigarrow$ $\Sigma'_X = (\mathbb{R}, \mathbb{R}^{\mathtt{w}_1}, \mathbb{R}^{\mathtt{n}}, \mathfrak{B}'_{\mathrm{full}})$. It is easy to deduce directly from the state axiom that Σ'_X is a state system if Σ_X is.

FROM LATENT VARIABLE to STATE REPRESENTATION

Consider the latent variable system $\Sigma_X = (\mathbb{R}, \mathbb{R}^{\mathtt{w}_1+\mathtt{w}_2}, \mathbb{R}^{\mathtt{n}}, \mathfrak{B}_{\mathrm{full}})$ with $\mathfrak{B}_{\mathrm{full}} \in \mathfrak{L}^{\mathtt{w}_1+\mathtt{w}_2+\mathtt{n}}$. Eliminate $w_2 \rightsquigarrow$ $\Sigma'_X = (\mathbb{R}, \mathbb{R}^{\mathtt{w}_1}, \mathbb{R}^{\mathtt{n}}, \mathfrak{B}'_{\mathrm{full}})$. It is easy to deduce directly from the state axiom that Σ'_X is a state system if Σ_X is.

Construction of a state representation for $\mathfrak{B}$:

1. $R(\frac{d}{dt})w = M(\frac{d}{dt})\ell$ latent variable representation for $\mathfrak{B}$.
2. Apply the cut-and-shift and stack operators to $[R \mid -M]$.
3. Obtain a state map

$$x = \Sigma_{[R \,|-M]}(\tfrac{d}{dt})\begin{bmatrix} w \\ \ell \end{bmatrix}.$$

FROM LATENT VARIABLE to STATE REPRESENTATION

Consider the latent variable system $\Sigma_X = (\mathbb{R}, \mathbb{R}^{\mathtt{w}_1+\mathtt{w}_2}, \mathbb{R}^{\mathtt{n}}, \mathfrak{B}_{\text{full}})$ with $\mathfrak{B}_{\text{full}} \in \mathfrak{L}^{\mathtt{w}_1+\mathtt{w}_2+\mathtt{n}}$. Eliminate $w_2 \rightsquigarrow$ $\Sigma'_X = (\mathbb{R}, \mathbb{R}^{\mathtt{w}_1}, \mathbb{R}^{\mathtt{n}}, \mathfrak{B}'_{\text{full}})$. It is easy to deduce directly from the state axiom that Σ'_X is a state system if Σ_X is.

Construction of a state representation for $\mathfrak{B}$:

1. $R(\frac{d}{dt})w = M(\frac{d}{dt})\ell$ latent variable representation for $\mathfrak{B}$.
2. Apply the cut-and-shift and stack operators to $[R \mid -M]$.
3. Obtain a state map

$$x = \Sigma_{[R\,|-M]}(\tfrac{d}{dt})\begin{bmatrix} w \\ \ell \end{bmatrix}.$$

$\rightsquigarrow$ a, not necessarily minimal, latent var'ble state repr'ion for $\mathfrak{B}$.

Notes

- **Basic idea of algorithms:**
  from latent variable representation directly to state model.

Notes

- **Basic idea of algorithms:**
  from latent variable representation directly to state model.
  This complements the existing algorithms
  transfer function $\longrightarrow$ i / s / o representation;
  impulse response $\longrightarrow$ i / s / o representation.

Notes

- **Basic idea of algorithms:**
  from latent variable representation directly to state model.
- $\exists$ **Gröbner basis techniques algorithms for state trimming.**

Notes

- **Basic idea of algorithms:**
  from latent variable representation directly to state model.
- $\exists$ **Gröbner basis techniques algorithms for state trimming.**
- **Our state construction is easily extended to state / input construction.**

Notes

- **Basic idea of algorithms:**
  from latent variable representation directly to state model.
- **$\exists$ Gröbner basis techniques algorithms for state trimming.**
- **Our state construction is easily extended to state / input construction.**
- **Examples of useful special (minimal) state representations:**

Notes

- **Basic idea of algorithms:**
  from latent variable representation directly to state model.
- $\exists$ **Gröbner basis techniques algorithms for state trimming.**
- **Our state construction is easily extended to state / input construction.**
- **Examples of useful special (minimal) state representations:**

  i/s/o representation:

$$\frac{d}{dt}x = Ax + Bu, \;\; y = Cx + Du, w = (u, y),$$

Notes

- **Basic idea of algorithms:**
  from latent variable representation directly to state model.
- $\exists$ **Gröbner basis techniques algorithms for state trimming.**
- **Our state construction is easily extended to state / input construction.**
- **Examples of useful special (minimal) state representations:**

  output nulling representation:

$$\frac{d}{dt}x = Ax + Bw, \;\; 0 = Cx + Dw,$$

Notes

- **Basic idea of algorithms:**
  from latent variable representation directly to state model.
- $\exists$ **Gröbner basis techniques algorithms for state trimming.**
- **Our state construction is easily extended to state / input construction.**
- **Examples of useful special (minimal) state representations:**

  driving variable representation:

$$\frac{d}{dt}x = Ax + Bv, \quad w = Cx + Dv.$$

Notes

- **Basic idea of algorithms:**
  from latent variable representation directly to state model.
- **$\exists$ Gröbner basis techniques algorithms for state trimming.**
- **Our state construction is easily extended to state / input construction.**
- **Examples of useful special (minimal) state representations:**

  Readily deduced from descriptor representation:

$$E\frac{d}{dt}x + Fx + Gw = 0.$$

BALANCED STATE CONSTRUCTION

THEME

!! Given a representation of a dynamical system,
find a representation of a reduced model !!

BALANCED STATE CONSTRUCTION

THEME

!! Given a representation of a dynamical system,
find a representation of a reduced model !!

We are looking for algorithms:

parameters of model $\longmapsto$ parameters of reduced model

BALANCED STATE CONSTRUCTION

THEME

!! Given a representation of a dynamical system, **find a representation of a reduced model !!**

We are looking for algorithms:

parameters of model $\longmapsto$ parameters of reduced model

For example,

model: discrete-time impulse response

reduced model: balanced reduced model

Algorithm: SVD of Hankel matrix.

BALANCED STATE CONSTRUCTION

THEME

!! Given a representation of a dynamical system, **find a representation of a reduced model !!**

We are looking for algorithms:

parameters of model $\longmapsto$ parameters of reduced model

For example,

model: transfer function

reduced model: balanced reduced model

Algorithm: ???

For simplicity, (today) only:

SISO systems & classical I/O balancing

For simplicity, (today) only:

SISO systems & classical I/O balancing

System $\cong \quad p, q \in \mathbb{R}[\xi], \mathrm{degree}(q) \leq \mathrm{degree}(p) =: \mathrm{n} \rightsquigarrow$

$$\boxed{p(\tfrac{d}{dt})y = q(\tfrac{d}{dt})u,}$$

relating the input $u : \mathbb{R} \to \mathbb{R}$ **to the output** $y : \mathbb{R} \to \mathbb{R}$**.**

For simplicity, (today) only:

SISO systems & classical I/O balancing

System $\cong$ $p, q \in \mathbb{R}[\xi]$, $\text{degree}(q) \leq \text{degree}(p) =: \text{n} \rightsquigarrow$

$$\boxed{p(\tfrac{d}{dt})y = q(\tfrac{d}{dt})u,}$$

relating the input $u : \mathbb{R} \to \mathbb{R}$ **to the output** $y : \mathbb{R} \to \mathbb{R}$.

Behavior:

$$\mathfrak{B}_{(p,q)} := \{(u,y) \in \mathfrak{L}_2^{\text{loc}}(\mathbb{R}, \mathbb{R}^2) \mid \text{diff. eq'n holds}\}.$$

CONTROLLABILITY & OBSERVABILITY

Well-known: $\mathfrak{B}_{(p,q)}$ is controllable iff p and q are co-prime.

CONTROLLABILITY & OBSERVABILITY

Well-known: $\mathfrak{B}_{(p,q)}$ **is controllable iff** p **and** q **are co-prime.**

Controllability $\Leftrightarrow$ $\exists$ **image representation for** $\mathfrak{B}_{(p,q)}$**:**

$$\boxed{u = p(\frac{d}{dt})\ell, \;\; y = q(\frac{d}{dt})\ell,}$$

$$\mathfrak{Im}_{(p,q)} := \{(u,y) \in \mathcal{L}_2^{\text{loc}}(\mathbb{R},\mathbb{R}^2) \mid \exists \ell : \mathbb{R} \to \mathbb{R} : \text{ diff. eq'n hol}$$

is *exactly* equal to $\mathfrak{B}_{(p,q)}$**.**

CONTROLLABILITY & OBSERVABILITY

Well-known: $\mathfrak{B}_{(p,q)}$ **is controllable iff** p **and** q **are co-prime.**

Controllability $\Leftrightarrow$ $\exists$ **image representation for** $\mathfrak{B}_{(p,q)}$**:**

$$\boxed{u = p(\frac{d}{dt})\ell, \;\; y = q(\frac{d}{dt})\ell,}$$

$$\mathfrak{Im}_{(p,q)} := \{(u,y) \in \mathcal{L}_2^{\text{loc}}(\mathbb{R},\mathbb{R}^2) \mid \exists \ell : \mathbb{R} \to \mathbb{R} : \text{ diff. eq'n hol}$$

is *exactly* equal to $\mathfrak{B}_{(p,q)}$**.** **Co-primeness of** p **and** q $\Rightarrow$

controllability of $\mathfrak{B}_{(p,q)}$ **&** **observability of** $\mathfrak{Im}_{(p,q)}$

observability means:

for every $(u,y) \in \mathfrak{Im}_{(p,q)} = \mathfrak{B}_{(p,q)}$, $\exists$ **unique** ℓ.

STATE POLYNOMIALS

Any set of polynomials $\{x_1, x_2, \ldots, x_\mathrm{n}\}$ that form a basis for $\mathbb{R}_{\mathrm{n}-1}[\xi] \Rightarrow$ a **minimal state representation** of $\mathfrak{B}_{(p,q)}$ with state

$$\boxed{x = (x_1(\frac{d}{dt})\ell, x_2(\frac{d}{dt})\ell, \ldots, x_{\mathrm{n}-1}(\frac{d}{dt})\ell).}$$

STATE POLYNOMIALS

Any set of polynomials $\{x_1, x_2, \ldots, x_{\mathrm{n}}\}$ that form a basis for $\mathbb{R}_{\mathrm{n}-1}[\xi] \Rightarrow$ a **minimal state representation** of $\mathfrak{B}_{(p,q)}$ with state

$$x = (x_1(\frac{d}{dt})\ell, x_2(\frac{d}{dt})\ell, \ldots, x_{\mathrm{n}-1}(\frac{d}{dt})\ell).$$

The associated system matrices are the (unique) solution matrix $\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$ of the following system of linear equations in $\mathbb{R}^{\mathrm{n}}[\xi]$:

$$\begin{bmatrix} \xi x_1(\xi) \\ \xi x_2(\xi) \\ \vdots \\ \xi x_{\mathrm{n}}(\xi) \\ q(\xi) \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x_1(\xi) \\ x_2(\xi) \\ \vdots \\ x_{\mathrm{n}}(\xi) \\ p(\xi) \end{bmatrix}.$$

BALANCING

In the context of the state construction through an image representation, being balanced becomes a property of the polynomials $x_1, x_2, \ldots, x_{\mathrm{n}}$.

The central problem is:

Choose the polynomials $x_1, x_2, \ldots, x_{\mathrm{n}}$ so that this

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

is balanced.

QDF's

The real two-variable polynomial

$$\Phi(\zeta,\eta) = \Sigma_{k,k'} \Phi_{k,k'} \zeta^{k} \eta^{k'}$$

induces the map

$$w \in \mathfrak{C}^{\infty}(\mathbb{R},\mathbb{R}) \mapsto \Sigma_{k,k'} \left(\frac{d^{k}}{dt^{k}} w\right) \Phi_{k,k'} \left(\frac{d^{k'}}{dt^{k'}} w\right) \in \mathfrak{C}^{\infty}(\mathbb{R},\mathbb{R}),$$

called a *a quadratic differential form* (QDF), denoted as Q_{Φ}.

THE CONTROLLABILITY GRAMIAN

We will consider the controllability and observability gramians as QDF's, acting on the latent variable ℓ of the image representation.

THE CONTROLLABILITY GRAMIAN

We will consider the controllability and observability gramians as QDF's, acting on the latent variable ℓ of the image representation.

The *controllability gramian* Q_K is defined as:

Let $\ell \in \mathfrak{C}^\infty(\mathbb{R}, \mathbb{R})$ and define $Q_K(\ell)$ by

$$Q_K(\ell)(0) := \text{infimum} \int_{-\infty}^{0} |p(\frac{d}{dt})\ell'(t)|^2 \, dt,$$

THE CONTROLLABILITY GRAMIAN

We will consider the controllability and observability gramians as QDF's, acting on the latent variable ℓ of the image representation.

The *controllability gramian* Q_K is defined as:

Let $\ell \in \mathfrak{C}^\infty(\mathbb{R}, \mathbb{R})$ and define $Q_K(\ell)$ by

$$Q_K(\ell)(0) := \text{infimum} \int_{-\infty}^{0} |p(\frac{d}{dt})\ell'(t)|^2 \, dt,$$

infimum over all $\ell' \in \mathfrak{C}^+(\mathbb{R}, \mathbb{R})$ that join the 'fixed' future ℓ at $t = 0$, i.e., such that $\ell(t) = \ell'(t)$ for $t \geq 0$.

THE OBSERVABILITY GRAMIAN

The ***observability gramian*** Q_W is defined as:

THE OBSERVABILITY GRAMIAN

The *observability gramian* Q_W is defined as: Let $\ell \in \mathfrak{C}^\infty(\mathbb{R},\mathbb{R})$ and define $Q_W(\ell)$ by

$$Q_W(\ell)(0) := \int_0^\infty |q(\frac{d}{dt})\ell'(t)|^2\, dt,$$

where $\ell' \in \mathfrak{D}(\mathbb{R},\mathbb{R})$ is such that

(i) $\ell|_{(-\infty,0)} = \ell'|_{(-\infty,0)}$,

(ii) $(p(\frac{d}{dt})\ell', q(\frac{d}{dt})\ell') \in \mathfrak{B}_{(p,q)}$,

(iii) $p(\frac{d}{dt})\ell'(t)|_{(0,\infty)} = 0$.

THE OBSERVABILITY GRAMIAN

The *observability gramian* Q_W is defined as: Let $\ell \in \mathfrak{C}^\infty(\mathbb{R},\mathbb{R})$ and define $Q_W(\ell)$ by

$$Q_W(\ell)(0) := \int_0^\infty |q(\frac{d}{dt})\ell'(t)|^2\,dt,$$

where $\ell' \in \mathfrak{D}(\mathbb{R},\mathbb{R})$ is such that

(i) $\ell|_{(-\infty,0)} = \ell'|_{(-\infty,0)}$,

(ii) $(p(\frac{d}{dt})\ell', q(\frac{d}{dt})\ell') \in \mathfrak{B}_{(p,q)}$,

(iii) $p(\frac{d}{dt})\ell'(t)|_{(0,\infty)} = 0.$

ℓ' smoothly cont's ℓ at $t = 0$ with $u|_{(0,\infty)} = p(\frac{d}{dt})\ell'|_{(0,\infty)} = 0.$

COMPUTATION of K and W

Given $\mathfrak{B}_{(p,q)}$, p, q **co-prime,** $\mathrm{degree}(q) \leq \mathrm{degree}(p) =: \mathtt{n}$, p **Hurwitz.**

COMPUTATION of K and W

Given $\mathfrak{B}_{(p,q)}$, p, q **co-prime,** $\text{degree}(q) \leq \text{degree}(p) =: \mathrm{n}$, p **Hurwitz.**

The controllability gramian and the observability gramian are QDF's, Q_K **and** Q_W**, with** $K, W \in \mathbb{R}[\zeta, \eta]$**. They can be computed as follows:**

COMPUTATION of K and W

Given $\mathfrak{B}_{(p,q)}$, p, q **co-prime,** $\text{degree}(q) \leq \text{degree}(p) =: \mathrm{n}$, p **Hurwitz.**

$$K(\zeta,\eta) = \frac{p(\zeta)p(\eta) - p(-\zeta)p(-\eta)}{\zeta + \eta}$$

COMPUTATION of K and W

Given $\mathfrak{B}_{(p,q)}$, p, q **co-prime,** $\mathrm{degree}(q) \leq \mathrm{degree}(p) =: \mathrm{n}$, p **Hurwitz.**

$$K(\zeta,\eta) = \frac{p(\zeta)p(\eta) - p(-\zeta)p(-\eta)}{\zeta+\eta}$$

$$W(\zeta,\eta) = \frac{p(\zeta)f(\eta) + f(\zeta)p(\eta) - q(\zeta)q(\eta)}{\zeta+\eta}$$

with $f \in \mathbb{R}_{\mathrm{n}-1}[\xi]$ **the (unique) solution of the Bezout-type equation**

$$p(\xi)f(-\xi) + f(\xi)p(-\xi) - q(\xi)q(-\xi) = 0.$$

BALANCED STATE REPRESENTATION

The minimal state repr. with polynomials $(x_1, x_2, \ldots, x_{\mathrm{n}})$ is *balanced* if

BALANCED STATE REPRESENTATION

The minimal state repr. with polynomials $(x_1, x_2, \ldots, x_{\mathrm{n}})$ is *balanced* if

(i) for ℓ_{k} such that $x_{\mathrm{k}'}(\frac{d}{dt})\ell_{\mathrm{k}}(0) = \delta_{\mathrm{kk}'}$ ($\delta_{\mathrm{kk}'}$: Kronecker delta):

$$Q_K(\ell_{\mathrm{k}})(0) = \frac{1}{Q_W(\ell_{\mathrm{k}})(0)}$$

states that are difficult to reach are also difficult to observe.

BALANCED STATE REPRESENTATION

The minimal state repr. with polynomials $(x_1, x_2, \ldots, x_n)$ is *balanced* if

(i) for ℓ_k such that $x_{k'}(\frac{d}{dt})\ell_k(0) = \delta_{kk'}$ ($\delta_{kk'}$: Kronecker delta):

$$Q_K(\ell_k)(0) = \frac{1}{Q_W(\ell_k)(0)}$$

states that are difficult to reach are also difficult to observe.

(ii) The state components are ordered so that 'easiest to reach first':

$$0 < Q_K(\ell_1)(0) \leq Q_K(\ell_2)(0) \leq \cdots \leq Q_K(\ell_n)(0),$$

and hence 'easiest to observe' first:

$$Q_W(\ell_1)(0) \geq Q_W(\ell_2)(0) \geq \cdots \geq Q_W(\ell_n)(0) > 0.$$

It is a standard result from linear algebra that there exist polynomials $(x_1^{\mathrm{bal}}, x_2^{\mathrm{bal}}, \ldots, x_{\mathrm{n}}^{\mathrm{bal}})$ that form a basis for $\mathbb{R}_{\mathrm{n}-1}[\xi]$, and real numbers $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\mathrm{n}} > 0$ such that K and W are factored as

$$K(\zeta,\eta) = \Sigma_{k=1}^{\mathrm{n}} \sigma_{\mathrm{k}}^{-1} x_{\mathrm{k}}^{\mathrm{bal}}(\zeta) x_{\mathrm{k}}^{\mathrm{bal}}(\eta)$$

$$W(\zeta,\eta) = \Sigma_{\mathrm{k}=1}^{\mathrm{n}} \sigma_{\mathrm{k}}\ x_{\mathrm{k}}^{\mathrm{bal}}(\zeta) x_{\mathrm{k}}^{\mathrm{bal}}(\eta)$$

It is a standard result from linear algebra that there exist polynomials $(x_1^{\mathrm{bal}}, x_2^{\mathrm{bal}}, \ldots, x_{\mathrm{n}}^{\mathrm{bal}})$ that form a basis for $\mathbb{R}_{\mathrm{n}-1}[\xi]$, and real numbers $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\mathrm{n}} > 0$ such that K and W are factored as

$$K(\zeta,\eta) = \Sigma_{k=1}^{\mathrm{n}} \sigma_{\mathrm{k}}^{-1} x_{\mathrm{k}}^{\mathrm{bal}}(\zeta) x_{\mathrm{k}}^{\mathrm{bal}}(\eta)$$

$$W(\zeta,\eta) = \Sigma_{\mathrm{k}=1}^{\mathrm{n}} \sigma_{\mathrm{k}} \; x_{\mathrm{k}}^{\mathrm{bal}}(\zeta) x_{\mathrm{k}}^{\mathrm{bal}}(\eta)$$

The σ_{k}'s are uniquely defined by K and W, the $x_{\mathrm{k}}^{\mathrm{bal}}$'s 'almost'.

THEOREM: These σ_{k}'s are the Hankel singular values of $\mathfrak{B}_{(p,q)}$ and

$$u = p(\frac{d}{dt})\ell, y = q(\frac{d}{dt})\ell,$$

$$x^{\mathrm{bal}} = (x_1^{\mathrm{bal}}(\frac{d}{dt})\ell, x_2^{\mathrm{bal}}(\frac{d}{dt})\ell, \ldots, x_{\mathrm{n}}^{\mathrm{bal}}(\frac{d}{dt})\ell)$$

is a balanced state space representation of $\mathfrak{B}_{(p,q)}$.

THEOREM: These $\sigma_{\rm k}$'s are the Hankel singular values of $\mathfrak{B}_{(p,q)}$ and

$$u = p(\frac{d}{dt})\ell, y = q(\frac{d}{dt})\ell,$$

$$x^{\rm bal} = (x_1^{\rm bal}(\frac{d}{dt})\ell, x_2^{\rm bal}(\frac{d}{dt})\ell, \ldots, x_{\rm n}^{\rm bal}(\frac{d}{dt})\ell)$$

is a balanced state space representation of $\mathfrak{B}_{(p,q)}$.
The balanced system matrices: sol'n of the following linear equations in $\mathbb{R}^{\rm n}[\xi]$:

$$\begin{bmatrix} \xi x_1^{\rm bal}(\xi) \\ \xi x_2^{\rm bal}(\xi) \\ \vdots \\ \xi x_{\rm n}^{\rm bal}(\xi) \\ q(\xi) \end{bmatrix} = \begin{bmatrix} A^{\rm bal} & B^{\rm bal} \\ C^{\rm bal} & D^{\rm bal} \end{bmatrix} \begin{bmatrix} x_1^{\rm bal}(\xi) \\ x_2^{\rm bal}(\xi) \\ \vdots \\ x_{\rm n}^{\rm bal}(\xi) \\ p(\xi) \end{bmatrix}.$$

ALGORITHM

DATA: $p, q \in \mathbb{R}[\xi]$, **co-prime**, $\mathrm{degree}(q) \leq \mathrm{degree}(p) := \mathtt{n}$,
p **Hurwitz.**

ALGORITHM

DATA: $p, q \in \mathbb{R}[\xi]$,

COMPUTE:

ALGORITHM

DATA: $p, q \in \mathbb{R}[\xi]$,

COMPUTE:

1. $K \in \mathbb{R}[\zeta, \eta]$,

$$\boxed{K(\zeta,\eta) = \frac{p(\zeta)p(\eta) - p(-\zeta)p(-\eta)}{\zeta + \eta}}$$

ALGORITHM

DATA: $p, q \in \mathbb{R}[\xi]$,

COMPUTE:

1. $K \in \mathbb{R}[\zeta, \eta]$,
2. $f \in \mathbb{R}_{n-1}[\xi]$ and $W \in \mathbb{R}[\zeta, \eta]$,

$$W(\zeta, \eta) = \frac{p(\zeta)f(\eta) + f(\zeta)p(\eta) - q(\zeta)q(\eta)}{\zeta + \eta}$$

$$p(\xi)f(-\xi) + f(\xi)p(-\xi) - q(\xi)q(-\xi) = 0.$$

ALGORITHM

DATA: $p, q \in \mathbb{R}[\xi]$,

COMPUTE:

1. $K \in \mathbb{R}[\zeta, \eta]$,
2. $f \in \mathbb{R}_{\mathrm{n}-1}[\xi]$ and $W \in \mathbb{R}[\zeta, \eta]$,
3. $(x_1^{\mathrm{bal}}, x_2^{\mathrm{bal}}, \ldots, x_{\mathrm{n}}^{\mathrm{bal}})$ and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\mathrm{n}} > 0$ by the expansions:

$$K(\zeta, \eta) = \Sigma_{k=1}^{\mathrm{n}} \sigma_{\mathrm{k}}^{-1} x_{\mathrm{k}}^{\mathrm{bal}}(\zeta) x_{\mathrm{k}}^{\mathrm{bal}}(\eta),$$

$$W(\zeta, \eta) = \Sigma_{\mathrm{k}=1}^{\mathrm{n}} \sigma_{\mathrm{k}}\ x_{\mathrm{k}}^{\mathrm{bal}}(\zeta) x_{\mathrm{k}}^{\mathrm{bal}}(\eta),$$

ALGORITHM

DATA: $p, q \in \mathbb{R}[\xi]$,

COMPUTE:

1. $K \in \mathbb{R}[\zeta, \eta]$,
2. $f \in \mathbb{R}_{n-1}[\xi]$ and $W \in \mathbb{R}[\zeta, \eta]$,
3. $(x_1^{\mathrm{bal}}, x_2^{\mathrm{bal}}, \ldots, x_n^{\mathrm{bal}})$ and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n > 0$
4. the balanced system matrices $\begin{bmatrix} A^{\mathrm{bal}} & B^{\mathrm{bal}} \\ C^{\mathrm{bal}} & D^{\mathrm{bal}} \end{bmatrix}$ by solving

$$\begin{bmatrix} \xi x_1^{\mathrm{bal}}(\xi) \\ \xi x_2^{\mathrm{bal}}(\xi) \\ \vdots \\ \xi x_n^{\mathrm{bal}}(\xi) \\ q(\xi) \end{bmatrix} = \begin{bmatrix} A^{\mathrm{bal}} & B^{\mathrm{bal}} \\ C^{\mathrm{bal}} & D^{\mathrm{bal}} \end{bmatrix} \begin{bmatrix} x_1^{\mathrm{bal}}(\xi) \\ x_2^{\mathrm{bal}}(\xi) \\ \vdots \\ x_n^{\mathrm{bal}}(\xi) \\ p(\xi) \end{bmatrix}.$$

ALGORITHM

DATA: $p, q \in \mathbb{R}[\xi]$,

COMPUTE:

1. $K \in \mathbb{R}[\zeta, \eta]$,
2. $f \in \mathbb{R}_{\mathrm{n}-1}[\xi]$ and $W \in \mathbb{R}[\zeta, \eta]$,
3. $(x_1^{\mathrm{bal}}, x_2^{\mathrm{bal}}, \ldots, x_{\mathrm{n}}^{\mathrm{bal}})$ and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\mathrm{n}} > 0$
4. the balanced system matrices $\begin{bmatrix} A^{\mathrm{bal}} & B^{\mathrm{bal}} \\ C^{\mathrm{bal}} & D^{\mathrm{bal}} \end{bmatrix}$

OUTPUT: a balanced state representation of $\mathfrak{B}_{(p,q)}$.

REMARKS

1. Model reduction by balanced truncation follows.

REMARKS

1. **Model reduction by balanced truncation follows.**

2. **These algorithms open up the possibility to involve 'fast' polynomial computations in order to obtain a balanced representation.**

REMARKS

1. **Model reduction** by balanced truncation follows.

2. These algorithms open up the possibility to involve **'fast' polynomial computations** in order to obtain a balanced representation.

3. The reduction algorithms solve linear equations in $\mathbb{R}_{\mathbf{n}-1}[\xi]$ **'approximately'**.

Suggests other (say, least squares) methods than simple truncation.

4. Instead of computing the σ_{k}'s and the $x_{\mathrm{k}}^{\mathrm{bal}}$'s by the factorization of K, W, we can also proceed by evaluating K and W at $\mathtt{n}$ distinct points

4. Instead of computing the σ_k's and the x_k^{bal}'s by the factorization of K, W, we can also proceed by **evaluating K and W at n distinct points**

Define

$$\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n)$$

$$K_\Lambda = \left[K(\lambda_k^*, \lambda_{k'})\right]_{k=1,\dots,n}^{k'=1,\dots,n} \quad W_\Lambda = \left[W(\lambda_k^*, \lambda_{k'})\right]_{k=1,\dots,n}^{k'=1,\dots,n}$$

$$\begin{aligned} X_\Lambda &= \left[x_k^{bal}(\lambda_{k'})\right]_{k=1,\dots,n}^{k'=1,\dots,n} \\ \Sigma &= \mathrm{diag}(\sigma_1, \sigma_2, \dots, \sigma_n) \end{aligned}$$

4. Instead of computing the σ_{k}'s and the $x_{\mathrm{k}}^{\mathrm{bal}}$'s by the factorization of K, W, we can also proceed by evaluating K and W at $\mathtt{n}$ distinct points

There holds

$$K_\Lambda = X_\Lambda^* \Sigma^{-1} X_\Lambda, W_\Lambda = X_\Lambda^* \Sigma X_\Lambda.$$

This implies that X_Λ and Σ can be computed directly from K_Λ, W_Λ.

4. Instead of computing the σ_{k}'s and the $x_{\mathrm{k}}^{\mathrm{bal}}$'s by the factorization of K, W, we can also proceed by **evaluating K and W at $\mathtt{n}$ distinct points**

There holds

$$K_{\Lambda} = X_{\Lambda}^{*}\Sigma^{-1}X_{\Lambda}, W_{\Lambda} = X_{\Lambda}^{*}\Sigma X_{\Lambda}.$$

This implies that X_{Λ} and Σ can be computed directly from K_{Λ}, W_{Λ}.

Once X_{Λ} is known, the matrices of the balanced state representation $\begin{bmatrix} A^{\mathrm{bal}} & B^{\mathrm{bal}} \\ C^{\mathrm{bal}} & D^{\mathrm{bal}} \end{bmatrix}$ is readily computed.

K_Λ follows immediately from evaluation of p at the λ_k's.

K_{Λ} follows immediately from evaluation of p at the λ_{k}'s.

Unfortunately, in order to compute W_{Λ} we have to solve for f.

However, if we take for the λ_k's the roots of p, assumed distinct,

then f is not needed,

and a very explicit expression for both K and W is obtained.

However, if we take for the λ_k's the roots of p, assumed distinct,

then f is not needed,

and a very explicit expression for both K and W is obtained.

In this case,

$$K_\Lambda = -\left[\frac{p(-\lambda_k^*)p(-\lambda_{k'})}{\lambda_k^* + \lambda_{k'}}\right]_{k=1,\ldots,n}^{k'=1,\ldots,n}$$

$$W_\Lambda = -\left[\frac{q(\lambda_k^*)q(\lambda_{k'})}{\lambda_k^* + \lambda_{k'}}\right]_{k=1,\ldots,n}^{k'=1,\ldots,n}$$

However, if we take for the λ_k's the roots of p, assumed distinct,

then f is not needed,

and a very explicit expression for both K and W is obtained.

In this case,

$$K_\Lambda = -\left[\frac{p(-\lambda_k^*)p(-\lambda_{k'})}{\lambda_k^* + \lambda_{k'}}\right]_{k=1,\ldots,n}^{k'=1,\ldots,n}$$

$$W_\Lambda = -\left[\frac{q(\lambda_k^*)q(\lambda_{k'})}{\lambda_k^* + \lambda_{k'}}\right]_{k=1,\ldots,n}^{k'=1,\ldots,n}$$

Balancing and model reduction: $\rightsquigarrow$ **the pencil**

$$\boxed{\left[\frac{p(-\lambda_k^*)p(-\lambda_{k'})}{\lambda_k^* + \lambda_{k'}}\right]_{k=1,\ldots,n}^{k'=1,\ldots,n} \; ; \; \left[\frac{q(\lambda_k^*)q(\lambda_{k'})}{\lambda_k^* + \lambda_{k'}}\right]_{k=1,\ldots,n}^{k'=1,\ldots,n}}$$

5. Heuristic: evaluate K, W at less than `n` points, obtain reduced model.

5. Heuristic: evaluate K, W at less than $\mathtt{n}$ points, obtain reduced model.

6. Suggests algorithms to fit the reduced order transfer function with the original transfer function at privileged points of the complex plane.

FROM TIME SERIES to LINEAR SYSTEM

FROM TIME SERIES to LINEAR SYSTEM

Problem of system identification:

Given an observed vector time-series (the 'data')

$$\hat{w}(1), \hat{w}(2), \hat{w}(3), \cdots, \hat{w}(t),$$

find a model for the system which produced this time-series.

FROM TIME SERIES to LINEAR SYSTEM

Problem of system identification:

Given an observed vector time-series (the ‘data’)

$$\hat{w}(1), \hat{w}(2), \hat{w}(3), \cdots, \hat{w}(t),$$

find a model for the system which produced this time-series.

Usual approach:

Assume an input/output partition: $w = \begin{bmatrix} u \\ y \end{bmatrix}$**, and assume the data produced by a stochastic system**

$$P(\sigma)y = Q(\sigma)u + N(\sigma)\varepsilon$$

with P, Q, N pol. matr., and ε something like gaussian, i.i.d.

! **Estimate**

$$\hat{P}_{\hat{w},t}, \hat{Q}_{\hat{w},t}, \hat{M}_{\hat{w},t}$$

from the data, and prove *consistency*

$$(\hat{P}_{\hat{w},t}, \hat{Q}_{\hat{w},t}, \hat{M}_{\hat{w},t}) \longrightarrow_{t\to\infty} (P, Q, N)$$

and other good features of the estimates.

! **Estimate**

$$\hat{P}_{\hat{w},t}, \hat{Q}_{\hat{w},t}, \hat{M}_{\hat{w},t}$$

from the data, and prove *consistency*

$$(\hat{P}_{\hat{w},t}, \hat{Q}_{\hat{w},t}, \hat{M}_{\hat{w},t}) \longrightarrow_{t\to\infty} (P, Q, N)$$

and other good features of the estimates.

***'Consistency paradigm':* If the data is produced by an element of the model class, then the algorithm should recover the model.**

Algorithms should work well for simulated data!

Our approach:

1. Exact modeling

Our approach:

1. **Exact modeling**

2. **Approximate modeling**

Our approach:

1. **Exact modeling**

2. **Approximate modeling**

3. **Stochastic modeling**

Our approach:

1. **Exact modeling**

2. **Approximate modeling**

3. **Stochastic modeling**

4. **Approximate stochastic modeling**

Assume an infinite 'observed' time-series

$$\hat{w} = (\hat{w}(1), \hat{w}(2), \hat{w}(3), \cdots, \hat{w}(t), \cdots)$$

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

Assume an infinite 'observed' time-series

$$\hat{w} = (\hat{w}(1), \hat{w}(2), \hat{w}(3), \cdots, \hat{w}(t), \cdots)$$

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

$\mathfrak{L}^{\mathtt{w}}$:= **set of discrete-time** $(\mathbb{T} = \mathbb{N})$ **linear difference systems.**

Assume an infinite 'observed' time-series

$$\hat{w} = (\hat{w}(1), \hat{w}(2), \hat{w}(3), \cdots, \hat{w}(t), \cdots)$$

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

Call $\mathfrak{B} \in \mathfrak{L}^{\mathtt{w}}$ unfalsified by $\hat{w}$ if $\hat{w} \in \mathfrak{B}$.

Assume an infinite 'observed' time-series

$$\hat{w} = (\hat{w}(1), \hat{w}(2), \hat{w}(3), \cdots, \hat{w}(t), \cdots)$$

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

Call $\mathfrak{B} \in \mathfrak{L}^{\mathtt{w}}$ unfalsified by $\hat{w}$ if $\hat{w} \in \mathfrak{B}$.

Call $\mathfrak{B}_1 \in \mathfrak{L}^{\mathtt{w}}$ more powerful than $\mathfrak{B}_2 \in \mathfrak{L}^{\mathtt{w}}$ if $\mathfrak{B}_1 \subset \mathfrak{B}_2$.

The more a model forbids, the better it is! (cfr Popper)

Assume an infinite 'observed' time-series

$$\hat{w} = (\hat{w}(1), \hat{w}(2), \hat{w}(3), \cdots, \hat{w}(t), \cdots)$$

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

Call $\mathfrak{B} \in \mathfrak{L}^{\mathtt{w}}$ **unfalsified by** $\hat{w}$ **if** $\hat{w} \in \mathfrak{B}$.

Call $\mathfrak{B}_1 \in \mathfrak{L}^{\mathtt{w}}$ **more powerful than** $\mathfrak{B}_2 \in \mathfrak{L}^{\mathtt{w}}$ **if** $\mathfrak{B}_1 \subset \mathfrak{B}_2$.

Call $\mathfrak{B}^{\star}_{\hat{w}} \in \mathfrak{L}^{\mathtt{w}}$ **the most powerful unfalsified model (MPUM) if**

(i) $\hat{w} \in \mathfrak{B}^{\star}_{\hat{w}}$**, and**
(ii) $\hat{w} \in \mathfrak{B} \in \mathfrak{L}^{\mathtt{w}} \;\Rightarrow\; \mathfrak{B} \subset \mathfrak{B}^{\star}_{\hat{w}}$

Proposition:

$$\mathfrak{B}^{\star}_{\hat{w}} \quad \textbf{exists!!}$$

Proposition:

$\mathfrak{B}^{\star}_{\hat{w}}$ **exists!!**

Easily generalized to a family of observed time-series.

SUBSPACE IDENTIFICATION

Construct first the underlying state sequence produced by $\hat{w}$ in $\mathfrak{B}^{\star}_{\hat{w}}$ and compute $\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$ from there!

SUBSPACE IDENTIFICATION

Construct first the underlying state sequence produced by $\hat{w}$ in $\mathfrak{B}^{\star}_{\hat{w}}$ and compute $\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$ from there!

There exist beautiful algorithms due to De Moor, Van Overschee, e.a. that do this.

SUBSPACE IDENTIFICATION

Construct first the underlying state sequence produced by $\hat{w}$ in $\mathfrak{B}^{\star}_{\hat{w}}$ and compute $\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$ from there!

There exist beautiful algorithms due to De Moor, Van Overschee, e.a. that do this.

To close, I will now add my own version...

Data:

$$\hat{w} = (\hat{w}(1), \hat{w}(2), \hat{w}(3), \cdots, \hat{w}(t), \cdots)$$

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

Data:

$$\hat{w} = (\hat{w}(1), \hat{w}(2), \hat{w}(3), \cdots, \hat{w}(t), \cdots)$$

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

Form the Hankel matrix of the data:

$$\mathfrak{H}_{\hat{w}} := \begin{bmatrix} \hat{w}(1) & \hat{w}(2) & \hat{w}(3) & \cdots & \hat{w}(t'') & \cdots \\ \hat{w}(2) & \hat{w}(3) & \hat{w}(4) & \cdots & \hat{w}(t''+1) & \cdots \\ \hat{w}(3) & \hat{w}(4) & \hat{w}(5) & \cdots & \hat{w}(t''+2) & \cdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots \\ \hat{w}(t') & \hat{w}(t'+1) & \hat{w}(t'+2) & \cdots & \hat{w}(t'+t''-1) & \cdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots \end{bmatrix}$$

Examine the rank of truncated Hankel matrices

$$\mathfrak{H}_{\hat{w}}^{t',\infty} := \begin{bmatrix} \hat{w}(1) & \hat{w}(2) & \hat{w}(3) & \cdots & \hat{w}(t'') & \cdots \\ \hat{w}(2) & \hat{w}(3) & \hat{w}(4) & \cdots & \hat{w}(t''+1) & \cdots \\ \hat{w}(3) & \hat{w}(4) & \hat{w}(5) & \cdots & \hat{w}(t''+2) & \cdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots \\ \hat{w}(t') & \hat{w}(t'+1) & \hat{w}(t'+2) & \cdots & \hat{w}(t'+t''-1) & \cdots \end{bmatrix}$$

for $t' = 1, 2, \cdots$ and determine a $t' = L$ until the 'permanent' rank increase by adding more block rows is stabilized.

Examine the rank of truncated Hankel matrices

$$\mathfrak{H}_{\hat{w}}^{t',\infty} := \begin{bmatrix} \hat{w}(1) & \hat{w}(2) & \hat{w}(3) & \cdots & \hat{w}(t'') & \cdots \\ \hat{w}(2) & \hat{w}(3) & \hat{w}(4) & \cdots & \hat{w}(t''+1) & \cdots \\ \hat{w}(3) & \hat{w}(4) & \hat{w}(5) & \cdots & \hat{w}(t''+2) & \cdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots \\ \hat{w}(t') & \hat{w}(t'+1) & \hat{w}(t'+2) & \cdots & \hat{w}(t'+t''-1) & \cdots \end{bmatrix}$$

for $t' = 1, 2, \cdots$ and determine a $t' = L$ until the 'permanent' rank increase by adding more block rows is stabilized.

The permanent rank increase = the number of input var. in $\mathfrak{B}_{\hat{w}}^{\star}$.

Determine vectors $r_1 \in \mathbb{R}^{\mathtt{n}_1 * \mathtt{w}}, r_2 \in \mathbb{R}^{\mathtt{n}_2 * \mathtt{w}}, \cdots, r_{\mathrm{g}} \in \mathbb{R}^{\mathtt{n}_{\mathrm{g}} * \mathtt{w}}$ **such that the vectors obtained by padding them with a multiple (possibly zero) of** $\mathtt{w}$ **zeros, form a left nullspace of** $\mathfrak{H}_{\hat{w}}^{L,\infty}$**.**

A typical such vector looks like

$$\left[\, 0 \cdots\ 0\ r_{\mathrm{k}}\ 0 \cdots\ 0 \,\right].$$

Determine vectors $r_1 \in \mathbb{R}^{\mathtt{n}_1 * \mathtt{w}}, r_2 \in \mathbb{R}^{\mathtt{n}_2 * \mathtt{w}}, \cdots, r_{\mathrm{g}} \in \mathbb{R}^{\mathtt{n}_{\mathrm{g}} * \mathtt{w}}$ such that the vectors obtained by padding them with a multiple (possibly zero) of $\mathtt{w}$ zeros, form a left nullspace of $\mathfrak{H}_{\hat{w}}^{L,\infty}$.
A typical such vector looks like

$$\left[\, 0 \cdots \; 0 \; r_{\mathrm{k}} \; 0 \cdots \; 0 \,\right].$$

Now pad with a multiple of $\mathtt{w}$ zeros before. A typical such vector:

$$\left[\, 0 \cdots \; 0 \; r_{\mathrm{k}} \,\right].$$

Let the first L blocks act on $\mathfrak{H}_{\hat{w}}^{L,\infty}$, obtain the state sequence

$$\left[\, \hat{x}(L), \, \hat{x}(L+1), \, \hat{x}(L+2), \, \cdots \,\right]$$

Determine vectors $r_1 \in \mathbb{R}^{\mathtt{n}_1 * \mathtt{w}}, r_2 \in \mathbb{R}^{\mathtt{n}_2 * \mathtt{w}}, \cdots, r_{\mathrm{g}} \in \mathbb{R}^{\mathtt{n}_{\mathrm{g}} * \mathtt{w}}$ such that the vectors obtained by padding them with a multiple (possibly zero) of $\mathtt{w}$ zeros, form a left nullspace of $\mathfrak{H}_{\hat{w}}^{L,\infty}$.
A typical such vector looks like

$$\left[\, 0 \cdots \; 0 \; r_{\mathrm{k}} \; 0 \cdots \; 0 \,\right].$$

Now pad with a multiple of $\mathtt{w}$ zeros before. A typical such vector:

$$\left[\, 0 \cdots \; 0 \; r_{\mathrm{k}} \,\right].$$

Let the first L blocks act on $\mathfrak{H}_{\hat{w}}^{L,\infty}$, obtain the state sequence

$$\left[\, \hat{x}(L), \, \hat{x}(L+1), \, \hat{x}(L+2), \, \cdots \,\right]$$

Note: there is no need to examine an infinite number of rows.

Now determine E, F, G by computing a left nullspace $\begin{bmatrix} -E & F & G \end{bmatrix}$ of the matrix

$$\begin{bmatrix} \hat{x}(L+1) & \hat{x}(L+2) & \hat{x}(L+3)\cdots \\ \hat{x}(L) & \hat{x}(L+1) & \hat{x}(L+2)\cdots \\ \hat{w}(L) & \hat{w}(L+1) & \hat{w}(L+2)\cdots \end{bmatrix}$$

Now determine E, F, G by computing a left nullspace $\begin{bmatrix} -E & F & G \end{bmatrix}$ of the matrix

$$\begin{bmatrix} \hat{x}(L+1) & \hat{x}(L+2) & \hat{x}(L+3)\cdots \\ \hat{x}(L) & \hat{x}(L+1) & \hat{x}(L+2)\cdots \\ \hat{w}(L) & \hat{w}(L+1) & \hat{w}(L+2)\cdots \end{bmatrix}$$

or $\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$ by first partitioning $\hat{w} = \begin{bmatrix} \hat{u} \\ \hat{y} \end{bmatrix}$ into inputs and outputs, and solving

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

for A, B, C, D.

Now determine E, F, G by computing a left nullspace $\begin{bmatrix} -E & F & G \end{bmatrix}$ of the matrix

$$\begin{bmatrix} \hat{x}(L+1) & \hat{x}(L+2) & \hat{x}(L+3)\cdots \\ \hat{x}(L) & \hat{x}(L+1) & \hat{x}(L+2)\cdots \\ \hat{w}(L) & \hat{w}(L+1) & \hat{w}(L+2)\cdots \end{bmatrix}$$

or $\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$ by first partitioning $\hat{w} = \begin{bmatrix} \hat{u} \\ \hat{y} \end{bmatrix}$ into inputs and outputs, and solving

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

for A, B, C, D.

Note: there is no need to examine an infinite number of rows.

Present research (jointly with I. Markovsky & P. Rapisarda):

Choose the basis of the computations, so that

$$\hat{x}(L), \hat{x}(L+1), \hat{x}(L+2), \cdots$$

corresponds to the state in a **balanced** basis, then truncate, and solve for $\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$ by least squares.

Present research (jointly with I. Markovsky & P. Rapisarda):

Choose the basis of the computations, so that

$$\hat{x}(L), \hat{x}(L+1), \hat{x}(L+2), \cdots$$

corresponds to the state in a balanced basis, then truncate, and solve for $\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$ by least squares.

Obtain stability, and an error bound.

The manuscript & copies of the lecture frames will be available from/at

`Jan.Willems@esat.kuleuven.ac.be`
`http://www.esat.kuleuven.ac.be/~jwillems`

Thank you for your attention !