Behavioral Approach to System Theory

Ivan Markovsky

About the course

lectures

- give enough background information for the exercises
- extras: optional presentations on special topics

exercises

- the core part of the course
- extras: open-ended explorations

mini-projects

- for those who need evaluation
- and for those who want to learn more

Outline

Introduction: the need

Basics: notation and conventions

Data-driven interpolation and approximation

Outline

Introduction: the need

Basics: notation and conventions

Data-driven interpolation and approximation

The classical approach views system as input-output map

$$\text{input} \longrightarrow \boxed{\text{system}} \longrightarrow \text{output}$$

the system is a signal processor

accepts input and produces output signal

intuition: the input causes the output

The input-output map view of the system is deficient: it ignores the initial condition

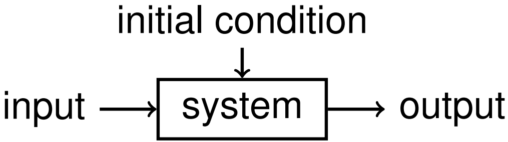
example: mass driven by external force

- input $\leftrightarrow$ force
- output $\leftrightarrow$ position
- ??? $\leftrightarrow$ position and velocity at start (initial condition)

input-output maps assume zero initial condition

how to account for nonzero initial condition?

Taking into account the initial condition leads to the state-space approach



paradigm shift from “classical” to “modern”

classical: scalar transfer function

modern: multivariable state-space

The modern state-space paradigm brought new theory, problems, and methods

state-space theory

- manifests the “finite memory” structure of the system
- brought the concepts of controllability and observability
- deals seamlessly with time-varying and MIMO systems

new problems / solution methods

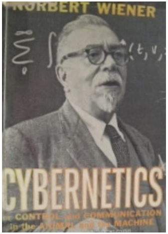
- linear quadratic optimal control (LQ control)
- optimal state estimation (the Kalman filter)
- balanced model reduction

amenable for numerical computations

A case in point: optimal filtering (signal from noise separation)

Wiener filter (1942)

- transfer functions approach
- assumes stationarity
- no practical real-time method



Kalman filter (1960)

- state-space approach
- non-stationary processes
- recursive real-time solution

There are other awkward things with the input/output thinking

modeling from first principles leads to relations

the input/output partitioning is not unique

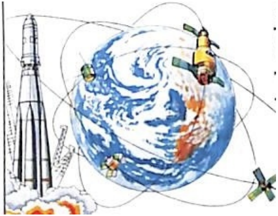
interconnection of systems is variables sharing

First principles modeling leads to relations

natural phenomena rarely operate as signal processors

the variables of interest satisfy relations, not functions

example: planetary orbits



More basic example: Ohmic resistor voltage and current satisfy relation

to-be-modeled variables: voltage V and current I

Ohm's law:

- $V = RI$, with R the resistance
- $I = CV$, with $C := 1/R$ the conductance

Q: how to fit the limit cases

- open circuit — $R = \infty$, $C = 0$
- short circuit — $R = 0$, $C = \infty$

neatly in a unified framework?

A: V, I satisfy (linear) relation

The behavioral approach was put forward by Jan C. Willems in the 1980's

3-part, 70-page, 1986–1987 Automatica paper:

Part I. Finite dimensional linear time invariant systems
Part II. Exact modelling
Part III. Approximate modelling

From Time Series to Linear System—
Part I. Finite Dimensional Linear Time Invariant Systems*

JAN C. WILLEMS†

Dynamical systems are defined in terms of their behaviour, and input/output systems appear as particular representations. Finite dimensional linear time invariant systems are characterized by the fact that their behaviour is a linear shift invariant complete (equivalently closed) subspace of $(\mathbb{R}^q)^{\mathbb{Z}}$ *or* $(\mathbb{R}^q)^{\mathbb{Z}_+}$.

Jan C. Willems (1939–2013)

Critical revision of the input/output thinking

simple idea: the system is set of trajectories

- variables not partitioned into inputs and outputs
- the system is separated from its representations

the input/output approach is a special case

relevant for the emerging data-driven paradigm

The behavior is all that matters

"The operations allowed to bring model equations in a more convenient form are exactly those that do not change the behavior. Dynamic modeling and system identification aim at coming up with a specification of the behavior. Control comes down to restricting the behavior."

J. C. Willems, "The behavioral approach to open and interconnected systems: Modeling by tearing, zooming, and linking," Control Systems Magazine, vol. 27, pp. 46–99, 2007.

Analogy with solution of systems of equations

Q: what operations are allowed?

A: the ones that don't change the solution set
(for linear systems, the "elementary operations")

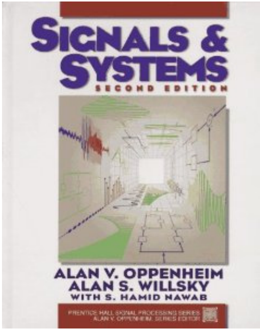
the solution set is all that matters

Classical definition of linear system

$S : u \mapsto y$ is linear $\iff$ S is linear function

for all u, v and $\alpha, \beta \in \mathbb{R}$,

$$S : \alpha u + \beta v \mapsto \alpha S(u) + \beta S(v)$$



The classical definition is deficient

(silently) assumes

- zero initial condition
- controllability

doesn't apply to autonomous systems

relaxing the assumptions requires state-space

Behavioral definition of linear system

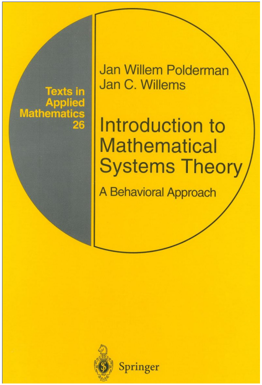
$\mathscr{B}$ is linear $\iff$ $\mathscr{B}$ is subspace

for all $w, v \in \mathscr{B}$ and $\alpha, \beta \in \mathbb{R}$

$$\alpha w + \beta v \in \mathscr{B}$$

fixes the issues with

- nonzero initial condition
- autonomous systems
- controllable systems



Separating problems from solution methods

different representations $\rightsquigarrow$ different methods

- with different properties (efficiency, robustness, ...)
- their common feature is that they solve the same problem

clarifies links among methods

leads to new methods

Summary: behavioral approach

detach the system from its representations

- define properties and problems in terms of the behavior
- lead to new, more general, definitions and problems
- avoid inconsistencies of the classical approach

separate problem from solution methods

- different representations lead to different methods
- show links among different methods
- lead to new solutions

naturally suited for the “data-driven paradigm”

Paradigms shifts

1940–1960	classical	SISO transfer function
1960–1980	modern	MIMO state-space
1980–2000	behavioral	the system as a set
2000–now	data-driven	using directly the data

Outline

Introduction: the need

Basics: notation and conventions

Data-driven interpolation and approximation

We will use the notation $(\mathbb{R}^q)^{\mathscr{T}}$ for the space of signals $w : \mathscr{T} \to \mathbb{R}^q$

$\mathscr{T}$ — time axis

- $\mathbb{R}$ or $\mathbb{R}_+$ or $[0, T]$ — continuous-time
- $\mathbb{Z}$ or $\mathbb{N}$ or $\{1, \dots, T\}$ — discrete-time

$(\mathbb{R}^q)^{\mathscr{T}}$ — real-valued q-variate signals

example: $w \in (\mathbb{R}^2)^{\mathbb{N}}$ means

$$w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \left(\begin{bmatrix} w_1(1) \\ w_2(1) \end{bmatrix}, \dots, \begin{bmatrix} w_1(t) \\ w_2(t) \end{bmatrix}, \dots \right)$$

It's a mistake to say "the signal $w(t)$"

let $w \in (\mathbb{R}^q)^{\mathbb{N}}$ and $t \in \mathbb{N}$

then, $w(t) \in \mathbb{R}^q$ is the *value* of w at time t

$w(t)$ is not signal (in $(\mathbb{R}^q)^{\mathbb{N}}$), but vector (in $\mathbb{R}^q$)

$w(\cdot)$ — specifies explicitly the time dependence of w

Use short, unambiguous, consistent notation

“$w = v$” means

$$\text{"}w(t) = v(t), \text{ for all } t \in \mathscr{T}\text{"}$$

shift operator σ

$$(\sigma w)(t) := w(t+1), \text{ for all } t \in \mathscr{T}$$

For example

ℓ-th order vector difference equation

$$R_0 w + R_1 \sigma w + \cdots + R_\ell \sigma^\ell w = 0$$

$$\Updownarrow$$

$$R_0 w(t) + R_1 w(t+1) + \cdots + R_\ell w(t+\ell) = 0, \text{ for all } t \in \mathbb{N}$$

first order state equation

$$\sigma x = Ax + Bu$$

$$\Updownarrow$$

$$x(t+1) = Ax(t) + Bu(t), \text{ for all } t \in \mathbb{N}$$

Compact notation for difference equation

$$R_0 w + R_1 \sigma w + \cdots + R_\ell \sigma^\ell w = 0$$

$$\Updownarrow$$

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

polynomial operator

$$R(\sigma) = R_0 + R_1 \sigma + \cdots + R_\ell \sigma^\ell$$

kernel of polynomial operator

$$\ker R(\sigma) := \{\, w \mid R(\sigma) w = 0 \,\}$$

We identify a dynamical system with its behavior, *i.e.*, the set of trajectories

real-valued system $\mathscr{B}$ with q variables
and time-axis $\mathscr{T}$ is a subset of $(\mathbb{R}^q)^{\mathscr{T}}$

in particular, we use set theoretic notation

$$
\begin{aligned}
w \in \mathscr{B} \quad &\iff \quad w \text{ is a trajectory of } \mathscr{B} \\
&\iff \quad \mathscr{B} \text{ is an exact model of } w
\end{aligned}
$$

… and specify $\mathscr{B}$ by representations

representation of the system $\mathscr{B} \subseteq (\mathbb{R}^q)^{\mathscr{T}}$

$$\mathscr{B} = \{\, w \in (\mathbb{R}^q)^{\mathscr{T}} \mid \text{"constraints on } w\text{"} \,\}$$

for example

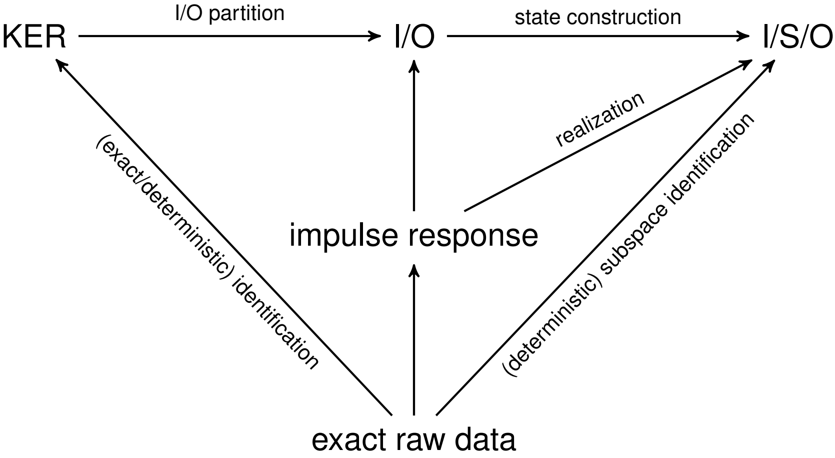
- kernel (KER) representation

$$\mathscr{B} = \ker R(\sigma) := \{\, w \mid R_0 w + R_1 \sigma w + \cdots + R_\ell \sigma^\ell w = 0 \,\}$$

- input/state/output (I/S/O) representation

$$\mathscr{B} = \left\{ w = \Pi \begin{bmatrix} u \\ y \end{bmatrix} \;\middle|\; \exists\, x \in (\mathbb{R}^n)^{\mathbb{N}},\ \begin{bmatrix} \sigma x \\ y \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} \right\}$$

Equivalence of representations and transformations among them



How to check if $w \in \mathscr{B}$?

depends on what representation of $\mathscr{B}$ is used

different repr. leads to different methods

for example

- if $\mathscr{B}$ is specified by vector difference equation

$$w \in \mathscr{B} \quad \iff \quad R_0 w + R_1 \sigma w + \cdots + R_\ell \sigma^\ell w = 0$$

- if $\mathscr{B}$ is specified by input/state/output representation

$$w \in \mathscr{B} \quad \iff \quad \exists\, x \in (\mathbb{R}^n)^{\mathbb{N}},\ \begin{bmatrix} \sigma x \\ y \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix}$$

Linearity and time-invariance are naturally defined in terms of $\mathscr{B}$

$$\mathscr{B} \text{ is linear system} \quad \iff \quad \mathscr{B} \text{ is subspace}$$

$$\mathscr{B} \text{ is time-invariant} \quad \iff \quad \sigma^{\tau}\mathscr{B} := \mathscr{B} \text{ for all } \tau$$

$$\sigma\mathscr{B} = \{\, \sigma w \mid w \in \mathscr{B} \,\}$$

$\mathscr{L}^{q}$ — set of LTI systems with q variables

The finite-horizon behavior $\mathscr{B}|_L$ is used for both analysis and computations

restriction of w to finite interval $[1, L]$

$$w|_L := \big(w(1), \dots, w(L)\big) \in (\mathbb{R}^q)^L$$

restriction of $\mathscr{B}$ to $[1, L]$

$$\mathscr{B}|_L := \{\, w|_L \mid w \in \mathscr{B} \,\} \subset (\mathbb{R}^q)^L$$

if $\mathscr{B}$ is linear, $\mathscr{B}|_L$ is a subspace of $(\mathbb{R}^q)^L$

$\mathscr{B}|_L$ can be obtained experimentally by collecting "informative" data

collect $N \geq qL$ random trajectories

$$w_{\mathrm{d}}^1, \ldots, w_{\mathrm{d}}^N \in \mathscr{B}|_L$$

by the linearity of $\mathscr{B}$, we have

$$\mathrm{span}\{\, w_{\mathrm{d}}^1, \ldots, w_{\mathrm{d}}^N \,\} \subseteq \mathscr{B}|_L$$

with probability one equality holds

Discrete-time LTI systems over finite horizon can be studied using linear algebra only

$$\underbrace{\begin{bmatrix} w_\mathrm{d}^1 & \cdots & w_\mathrm{d}^N \end{bmatrix}}_{W} \in \mathbb{R}^{qL\times N} \text{ — “trajectory matrix”}$$

$\widehat{\mathscr{B}}|_L = \text{image}\, W$ — “data-driven model” of $\mathscr{B}|_L$

now, we can do explorations using Matlab

What is the dimension of $\mathscr{B}|_L$?

take a random LTI system

```
m = 2; p = 5; n = 20; B = drss(n, p, m);
```

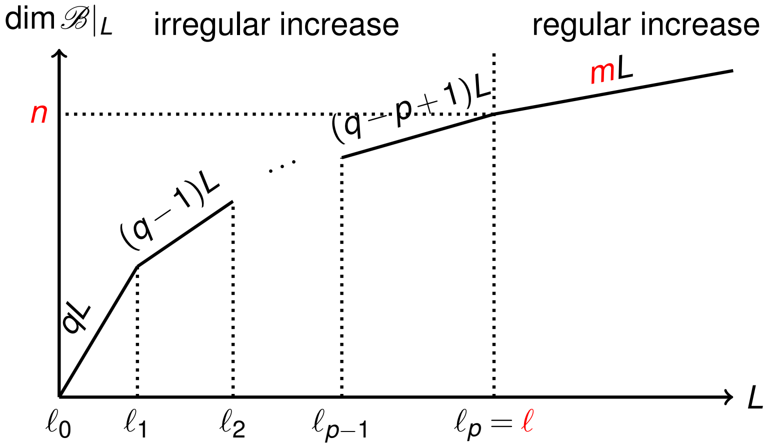
generate qL random trajectories of length L

```
L = 100; q = m + p; W = []; vec = @(a) a(:);
for i = 1:q*L
  u = rand(L, m); xini = rand(n, 1);
  y = lsim(B, u, [], xini);
  w = [u y]; W = [W vec(w')];
end
```

assuming that image $\mathtt{W} = \mathscr{B}|_L$, find $\dim \mathscr{B}|_L$

```
for t = 1:L, d(t) = rank(W(1:q*t, :)); end
stem(d)
```

$\dim \mathscr{B}|_L$ is a piecewise affine function of L



in particular, $\dim \mathscr{B}|_L = mL + n$, for all $L \geq \ell$

The set of linear time-invariant systems $\mathscr{L}$ has structure characterized by set of integers

the dimension of $\mathscr{B} \in \mathscr{L}$ is determined by

$\mathbf{m}(\mathscr{B})$ — number of inputs

$\mathbf{n}(\mathscr{B})$ — order ($=$ minimal state dimension)

$\boldsymbol{\ell}(\mathscr{B})$ — lag ($=$ observability index)

J.C. Willems, From time series to linear systems.
Part I, Finite dimensional linear time invariant systems, Automatica, 22(561–580), 1986

$\mathscr{B}_1$ less complex than $\mathscr{B}_2 \quad \iff \quad \mathscr{B}_1 \subset \mathscr{B}_2$

in the LTI case, complexity $\leftrightarrow$ dimension

complexity: (# inputs, order, lag)

$$\mathbf{c}(\mathscr{B}) := \big(\mathbf{m}(\mathscr{B}), \mathbf{n}(\mathscr{B}), \boldsymbol{\ell}(\mathscr{B})\big)$$

$\mathscr{L}_c$ — bounded complexity LTI model class

$$\mathscr{L}_c^q := \{\mathscr{B} \in \mathscr{L}^q \mid \mathbf{c}(\mathscr{B}) \leq c\}$$

Finite vs infinite dimensional LTI systems

$$\mathscr{B} \in \mathscr{L}^q \text{ finite-dimensional} \quad :\Longleftrightarrow \quad \begin{array}{l} \mathbf{m}(\mathscr{B}) < q \\ \mathbf{n}(\mathscr{B}) < \infty \end{array}$$

equivalently

- $\mathscr{B}$ has *bounded complexity* $\mathbf{c}(\mathscr{B})$
- $\mathscr{B}$ admits KER and I/S/O representations
- $\mathscr{B}$ admits rational transfer function representation

parametric representations of $\mathscr{B} \in \mathscr{L}_c^q$

Summary

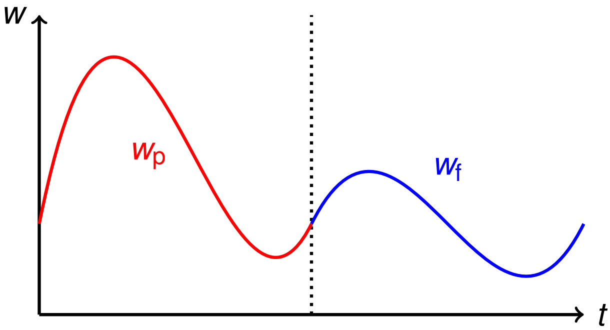
$w \in (\mathbb{R}^q)^{\mathscr{T}}$ — q-variate signal

$\mathscr{B} \in \mathscr{L}^q$ — q-variate LTI system

$$\dim \mathscr{B}|_L = \mathbf{m}(\mathscr{B})L + \mathbf{n}(\mathscr{B}), \quad \text{for all } L \geq \boldsymbol{\ell}(\mathscr{B})$$

Initial conditions specified by “past” trajectory

$$w = w_p \wedge w_f$$



How long should $w_{\rm p}$ be in order to specify the initial conditions for $w_{\rm f}$?

answer: at least $\boldsymbol{\ell}(\mathscr{B})$ samples

in general, there are infinitely many $w_{\rm p}$'s that specify the same initial condition

$w_{\rm p}$ is a non-minimal state vector

Input/output partitioning of the variables

$w =: \Pi \begin{bmatrix} u \\ y \end{bmatrix}$, with Π permutation, such that

u is input $:=$ free variable
y is output $:=$ uniquely defined by $\mathscr{B}$, w_{ini}, and u

simulation problem: $(\mathscr{B}, w_{\text{ini}}, u) \mapsto y$

parametrization of w by u and w_{ini}

Initial conditions recovery (observer)

given $\mathscr{B}$ and $w_{\mathrm{f}} \in \mathscr{B}|_{T_{\mathrm{f}}}$, find $w_{\mathrm{p}} \in (\mathbb{R}^q)^{T_{\mathrm{p}}}$, s.t.

$$w_{\mathrm{p}} \wedge w_{\mathrm{f}} \in \mathscr{B}|_{T_{\mathrm{p}}+T_{\mathrm{f}}}$$

feasibility problem, solution always exists (why?)

in general, it is not unique (is this an issue?)

Initial conditions estimation (smoothing)

given $\mathscr{B}$ and $w_\text{f} \in (\mathbb{R}^q)^{T_\text{f}}$, find $w_\text{p} \in (\mathbb{R}^q)^{T_\text{p}}$ that

$$\begin{array}{ll} \text{minimize} & \text{over } \widehat{w}_\text{p}, \widehat{w}_\text{f} \quad \| w_\text{f} - \widehat{w}_\text{f} \| \\ \text{subject to} & \widehat{w}_\text{p} \wedge \widehat{w}_\text{f} \in \mathscr{B}|_{T_\text{p}+T_\text{f}} \end{array}$$

as byproduct we find "smoothed" trajectory $\widehat{w}_\text{f}$

errors-in-variables (EIV) smoother

Projection on $\mathscr{B}$

given $\mathscr{B}$ and $w \in (\mathbb{R}^q)^T$, find $\widehat{w} \in (\mathbb{R}^q)^T$ that

$$\begin{array}{ll} \text{minimize} & \text{over } \widehat{w} \quad \|w - \widehat{w}\| \\ \text{subject to} & \widehat{w} \in \mathscr{B}|_T \end{array}$$

equivalent to the EIV smoothing problem

prior knowledge about the initial conditions

- completely unknown
- uncertain (mean value and covariance are given)
- given exactly

Most powerful unfalsified model of $\mathscr{B}_{\text{mpum}}(w_{\text{d}})$

exact identification problem

$$\mathscr{B}_{\text{mpum}}(w_{\text{d}}) := \arg \underbrace{\min_{\widehat{\mathscr{B}} \in \mathscr{L}} \; c(\widehat{\mathscr{B}})}_{\text{most powerful}} \quad \text{subject to} \quad \underbrace{w_{\text{d}} \in \widehat{\mathscr{B}}}_{\text{unfalsified model}}$$

multi-objective optimization problem

- complexities are compared in the lexicographic order
- more inputs imply higher complexity irrespective of order

feasibility and uniqueness are guaranteed

$$\mathscr{B}_{\text{mpum}}(w_{\text{d}}) := \operatorname{span}\{\, w_{\text{d}}, \sigma w_{\text{d}}, \sigma^2 w_{\text{d}}, \ldots \,\}$$

There is a problem with $\mathscr{B}_{\text{mpum}}(w_{\text{d}})$ in case of finite data

$\widehat{\mathscr{B}} := \mathscr{B}_{\text{mpum}}(w_{\text{d}})$ is autonomous

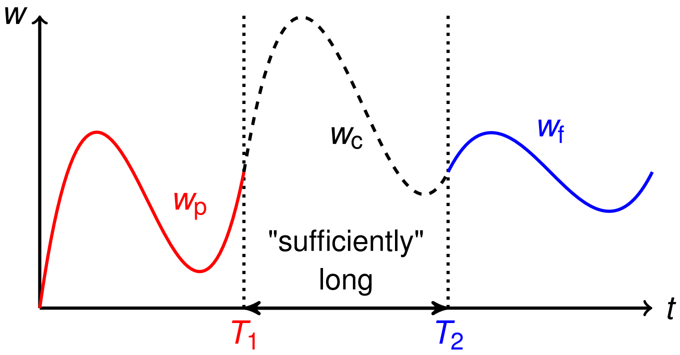
solution: impose the upper bound

$$\boldsymbol{\ell}(\widehat{\mathscr{B}}) \leq \ell_{\max} := \left\lfloor \frac{T+1}{q+1} \right\rfloor$$

What means that $\mathscr{B}$ is controllable?

controllability is the property of “patching”
any past trajectory with any future trajectory

$$w_\mathrm{p} \wedge w_\mathrm{c} \wedge w_\mathrm{f} \in \mathscr{B}$$



Compare with the classical definition: transfer from any initial to any terminal state

property of a state-space representation of $\mathscr{B}$

- is lack of controllability due to a “bad” choice of the state or due to an intrinsic issue with the system?
- in the LTI case, does it make sense to talk about controllability of a transfer function representation?
- how to quantify the “distance” to uncontrollability?

does not apply to infinite dimensional system

Methods for checking controllability

how to check controllability of an LTI system?

using state-space representation:

1. ensure minimality (in the behavioral sense)
2. perform rank test for the controllability matrix

using matrix fraction representation:

$$\mathscr{B} = \left\{ w = \Pi \left[\begin{smallmatrix} u \\ y \end{smallmatrix}\right] \in (\mathbb{R}^q)^{\mathbb{N}} \mid N(\sigma)u = D(\sigma)y \right\}$$

- facts: $\mathscr{B}$ is controllable $\iff$ N and D are co-prime
- $\rightsquigarrow$ rank test for the (generalized) Sylvester matrix

Summary

"past" trajectory — specifies initial conditions

simulation: with $w =: \Pi \left[\begin{smallmatrix} u \\ y \end{smallmatrix} \right]$, $(\mathscr{B}, w_{\text{ini}}, u) \mapsto y$

inverse problem: $w_{\text{d}} \mapsto \mathscr{B}_{\text{mpum}}(w_{\text{d}})$

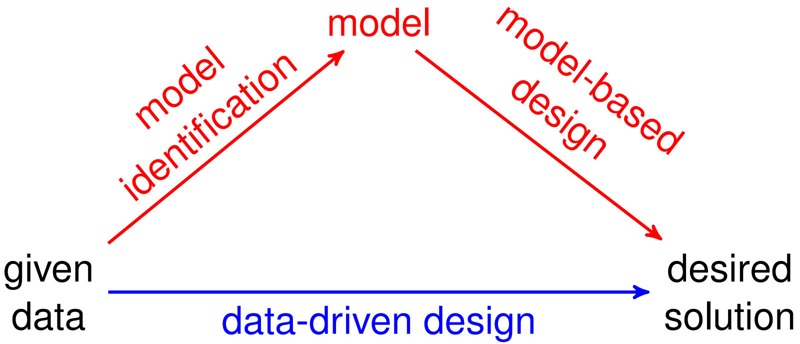
Outline

Introduction: the need

Basics: notation and conventions

Data-driven interpolation and approximation

The new "data-driven" paradigm obtains desired solution directly from given data



Data-driven does not mean model-free

data-driven problems do assume model

however, specific representation is not fixed

the methods we review are non-parametric

Data-driven representation (infinite horizon)

data: exact infinite trajectory w_d of $\mathscr{B} \in \mathscr{L}$

$\widehat{\mathscr{B}} = \mathscr{B}_{\mathrm{mpum}}(w_\mathrm{d}) = \mathrm{span}\{\, w_\mathrm{d}, \sigma w_\mathrm{d}, \sigma^2 w_\mathrm{d}, \dots \}$

identifiability condition: $\quad \mathscr{B} = \widehat{\mathscr{B}}$

Consecutive application of σ on finite w_d results in Hankel matrix with missing values

$$
\begin{array}{cccc}
\sigma^0 w_\mathrm{d} & \sigma^1 w_\mathrm{d} & \cdots & \sigma^{T_\mathrm{d}-1} w_\mathrm{d} \\
\hline
w_\mathrm{d}(1) & w_\mathrm{d}(2) & \cdots & w_\mathrm{d}(T_\mathrm{d}) \\
w_\mathrm{d}(2) & \vdots & \iddots & ? \\
\vdots & w_\mathrm{d}(T_\mathrm{d}) & \iddots & \vdots \\
w_\mathrm{d}(T_\mathrm{d}) & ? & \cdots & ?
\end{array}
$$

for $w_\mathrm{d} = \big(w_\mathrm{d}(1), \dots, w_\mathrm{d}(T)\big)$ and $1 \leq L \leq T$

$$\mathscr{H}_L(w_\mathrm{d}) := \Big[(\sigma^0 w_\mathrm{d})|_L \;\; (\sigma^1 w_\mathrm{d})|_L \;\; \cdots \;\; (\sigma^{T-L} w_\mathrm{d})|_L\Big]$$

Data-driven representation (finite horizon)

the finite horizon data-driven representation

$$\mathscr{B}|_L = \widehat{\mathscr{B}}|_L := \text{image}\, \mathscr{H}_L(w_\text{d}) \qquad \text{(DD-REPR)}$$

holds if and only if

$$\text{rank}\, \mathscr{H}_L(w_\text{d}) = L\mathbf{m}(\mathscr{B}) + \mathbf{n}(\mathscr{B}) \qquad \text{(GPE)}$$

I. Markovsky and F. Dörfler, Identifiability in the Behavioral Setting, 2020

Identifiability condition verifiable from $w_\mathrm{d} \in \mathscr{B}|_T$ and (m, ℓ, n)

fact: $\mathscr{B} = \mathscr{B}' \iff \mathscr{B}|_{\ell+1} = \mathscr{B}'|_{\ell+1}$, then

$$\begin{aligned} \widehat{\mathscr{B}} = \mathscr{B} \quad &\iff \quad \widehat{\mathscr{B}}|_{\ell+1} = \mathscr{B}|_{\ell+1} \\ &\iff \quad \dim \widehat{\mathscr{B}}|_{\ell+1} = \dim \mathscr{B}|_{\ell+1} \end{aligned}$$

$\mathscr{B}$ is identifiable from $w_\mathrm{d} \in \mathscr{B}|_T$ if and only if

$$\operatorname{rank} \mathscr{H}_{\ell+1}(w_\mathrm{d}) = (\ell+1)m + n$$

The "fundamental lemma" is an input design result

sufficient conditions for (DD-REPR)

1. $w_\mathrm{d} = \left[\begin{smallmatrix} u_\mathrm{d} \\ y_\mathrm{d} \end{smallmatrix}\right]$

2. $\mathscr{B}$ controllable

3. $\mathscr{H}_{L+n}(u_\mathrm{d})$ full row rank (PE condition)

J.C. Willems et al., A note on persistency of excitation
Systems & Control Letters, (54)325–329, 2005

PE — persistency of excitation, GPE — generalized PE

Generic data-driven problem: trajectory interpolation/approximation

given: “data” trajectory $w_\mathrm{d} \in \mathscr{B}|_T$

partially specified trajectory $w|_{I_\text{given}}$

($w|_{I_\text{given}}$ selects the elements of w, specified by I_given)

aim: minimize over $\widehat{w}$ $\|w|_{I_\text{given}} - \widehat{w}|_{I_\text{given}}\|$

subject to $\widehat{w} \in \mathscr{B}|_L$

$$\widehat{w} = \mathscr{H}_L(w_\mathrm{d})\big(\mathscr{H}_L(w_\mathrm{d})|_{I_\text{given}}\big)^+ w|_{I_\text{given}} \qquad \text{(SOL)}$$

Special cases

simulation

- given data: initial condition and input
- to-be-found: output (exact interpolation)

smoothing

- given data: noisy trajectory
- to-be-found: ℓ_2-optimal approximation

tracking control

- given data: to-be-tracked trajectory
- to-be-found: ℓ_2-optimal approximation

Generalizations

multiple data trajectories $w_{\rm d}^1, \ldots, w_{\rm d}^N$

$$\mathscr{B} = \text{image} \underbrace{\begin{bmatrix} \mathscr{H}_L(w_{\rm d}^1) & \cdots & \mathscr{H}_L(w_{\rm d}^N) \end{bmatrix}}_{\text{mosaic-Hankel matrix}}$$

$w_{\rm d}$ not exact / noisy

maximum-likelihood estimation
$\rightsquigarrow$ Hankel structured low-rank approximation/completion
nuclear norm and ℓ_1-norm relaxations
$\rightsquigarrow$ nonparametric, convex optimization problems

nonlinear systems

results for special classes of nonlinear systems:
Volterra, Wiener-Hammerstein, bilinear, ...

Summary: data-driven signal processing

data-driven representation

leads to general, simple, practical methods

interpolation/approximation of trajectories

simulation, filtering and control are special cases
assumes only LTI dynamics; no hyper parameters

dealing with noise and nonlinearities

nonlinear optimization
convex relaxations

The data w_{d} being exact vs inexact / "noisy"

w_{d} exact and satisfying (GPE)

- "system theory" problems
- image $\mathscr{H}_L(w_{\mathrm{d}})$ is nonparametric finite-horizon model
- data-driven solution $=$ model-based solution

w_{d} inexact, due to noise and/or nonlinearities

- naive approach: apply the solution (SOL) for exact data
- rigorous: assume noise model $\rightsquigarrow$ ML estimation problem
- heuristics: convex relaxations of the ML estimator

The maximum-likelihood estimation problem in the errors-in-variables setup is nonconvex

errors-in-variables setup: $w_{\text{d}} = \overline{w}_{\text{d}} + \widetilde{w}_{\text{d}}$

- $\overline{w}_{\text{d}}$ — true data, $\overline{w}_{\text{d}} \in \mathscr{B}|_T$, $\mathscr{B} \in \mathscr{L}_c^q$
- $\widetilde{w}_{\text{d}}$ — zero mean, white, Gaussian measurement noise

ML problem: given w_{d}, c, and $w|_{I_{\text{given}}}$

$$\begin{aligned} \underset{g}{\text{minimize}} \quad & \| w|_{I_{\text{given}}} - \mathscr{H}_L(\widehat{w}_{\text{d}}^*)|_{I_{\text{given}}} g \| \\ \text{subject to} \quad & \widehat{w}_{\text{d}}^* = \arg\min_{\widehat{w}_{\text{d}}, \widehat{\mathscr{B}}} \quad \| w_{\text{d}} - \widehat{w}_{\text{d}} \| \\ & \qquad \text{subject to} \quad \widehat{w}_{\text{d}} \in \widehat{\mathscr{B}}|_T \text{ and } \widehat{\mathscr{B}} \in \mathscr{L}_c^q \end{aligned}$$

The ML estimation problem is equivalent to Hankel structured low-rank approximation

$$\begin{aligned}
\underset{g}{\text{minimize}} \quad & \|w|_{I_{\text{given}}} - \mathscr{H}_L(\widehat{w}_{\text{d}}^*)|_{I_{\text{given}}} g\| \\
\text{subject to} \quad & \widehat{w}_{\text{d}}^* = \arg\min_{\widehat{w}_{\text{d}}, \widehat{\mathscr{B}}} \quad \|w_{\text{d}} - \widehat{w}_{\text{d}}\| \\
& \qquad \text{subject to} \quad \widehat{w}_{\text{d}} \in \widehat{\mathscr{B}}|_T \text{ and } \widehat{\mathscr{B}} \in \mathscr{L}_c^q
\end{aligned}$$

$$\Updownarrow$$

$$\begin{aligned}
\underset{g}{\text{minimize}} \quad & \|w|_{I_{\text{given}}} - \mathscr{H}_L(\widehat{w}_{\text{d}}^*)|_{I_{\text{given}}} g\| \\
\text{subject to} \quad & \widehat{w}_{\text{d}}^* = \arg\min_{\widehat{w}_{\text{d}}} \quad \|w_{\text{d}} - \widehat{w}_{\text{d}}\| \\
& \qquad \text{subject to} \quad \operatorname{rank} \mathscr{H}_{\ell+1}(\widehat{w}_{\text{d}}) \leq (\ell+1)m + n
\end{aligned}$$

Solution methods

local optimization

- choose a parametric representation of $\widehat{\mathscr{B}}(\theta)$
- optimize over $\widehat{w}$, $\widehat{w_{\mathrm{d}}}$, and θ
- depends on the initial guess

convex relaxation based on the nuclear norm

$$\text{minimize} \quad \text{over } \widehat{w}_{\mathrm{d}} \text{ and } \widehat{w} \quad \|w|_{I_{\text{given}}} - \widehat{w}|_{I_{\text{given}}}\| + \|w_{\mathrm{d}} - \widehat{w}_{\mathrm{d}}\| \\ + \gamma \cdot \left\| \begin{bmatrix} \mathscr{H}_{\Delta}(\widehat{w}_{\mathrm{d}}) & \mathscr{H}_{\Delta}(\widehat{w}) \end{bmatrix} \right\|_*$$

convex relaxation based on ℓ_1-norm (LASSO)

$$\text{minimize} \quad \text{over } g \quad \|w|_{I_{\text{given}}} - \mathscr{H}_L(w_{\mathrm{d}})|_{I_{\text{given}}} g\| + \lambda \|g\|_1$$

Empirical validation on real-life datasets

	data set name	T	m	p
1	Air passengers data	144	0	1
2	Distillation column	90	5	3
3	pH process	2001	2	1
4	Hair dryer	1000	1	1
5	Heat flow density	1680	2	1
6	Heating system	801	1	1

G. Box, and G. Jenkins. Time Series Analysis: Forecasting and Control, Holden-Day, 1976

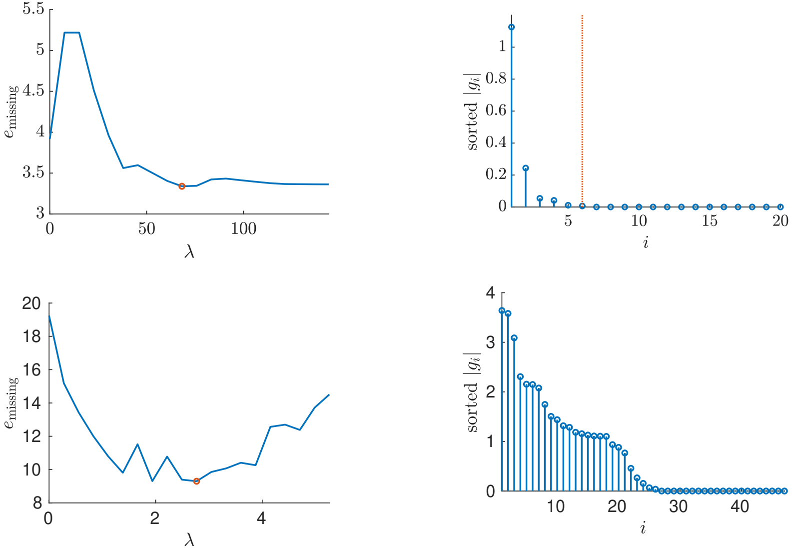
B. De Moor, et al.*DAISY: A database for identification of systems. Journal A, 38:4–5, 1997*

ℓ_1-norm regularization with optimized λ achieves the best performance

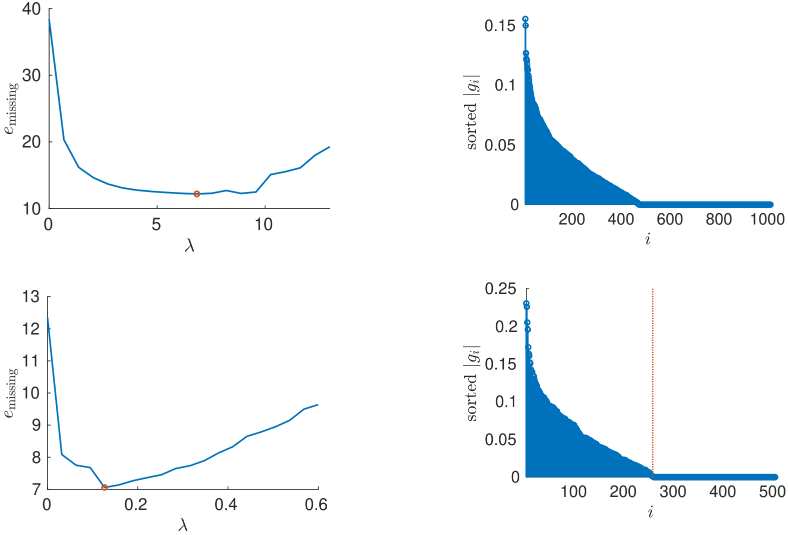
$$e_{\text{missing}} := \frac{\left\| w|_{I_{\text{missing}}} - \widehat{w}|_{I_{\text{missing}}} \right\|}{\left\| w|_{I_{\text{missing}}} \right\|} \, 100\%$$

	data set name	naive	ML	LASSO
1	Air passengers data	3.9	fail	3.3
2	Distillation column	19.24	17.44	9.30
3	pH process	38.38	85.71	12.19
4	Hair dryer	12.35	8.96	7.06
5	Heat flow density	7.16	44.10	3.98
6	Heating system	0.92	1.35	0.36

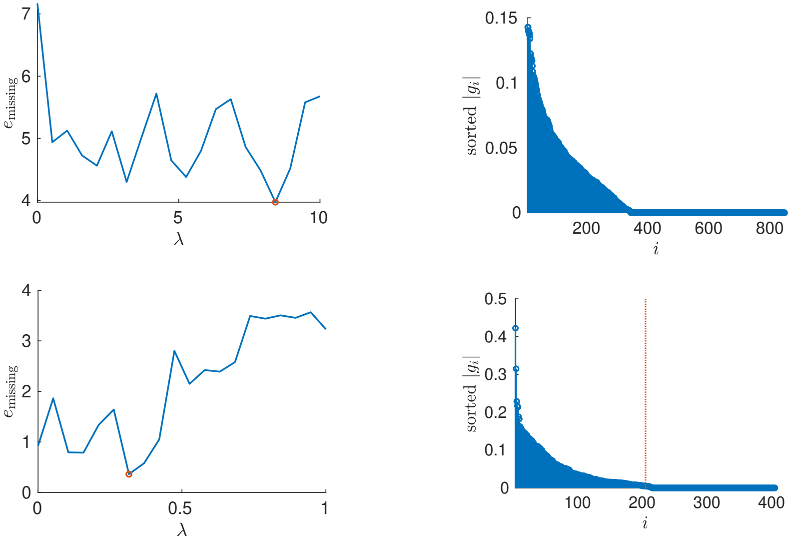
Tuning of λ and sparsity of g (datasets 1, 2)



Tuning of λ and sparsity of g (datasets 3, 4)



Tuning of λ and sparsity of g (datasets 5, 6)



Summary: convex relaxations

w_d exact $\rightsquigarrow$ system theory

- exact analytical solution
- current work: efficient real-time algorithms

w_d inexact $\rightsquigarrow$ nonconvex optimization

- subspace methods
- local optimization
- convex relaxations

empirical validation

- the naive approach works (surprisingly) well
- parametric local optimization is not robust
- ℓ_1-norm regularization gives the best results

Extras

Constructive proof of the fundamental lemma

Pedagogical example: Free fall prediction

Case study: Dynamic measurement

Nonparametric frequency response estimation

Generalization for nonlinear systems

Outline

The fundamental lemma gives data-driven finite horizon representation of LTI system $\mathscr{B}$

$$\mathscr{B}|_L = \operatorname{image} \mathscr{H}_L(w_\mathrm{d}) \qquad \text{(DD-REPR)}$$

assumptions:

- A0 $w_\mathrm{d} = \left[\begin{smallmatrix} u_\mathrm{d} \\ y_\mathrm{d} \end{smallmatrix}\right]$ is a trajectory of an LTI system $\mathscr{B}$
- A1 $\mathscr{B}$ is controllable
- A2 u_d is persistently exciting of order $L+n$

Decoding the notation $\mathscr{B}|_L = \operatorname{image} \mathscr{H}_L(w_\mathrm{d})$

$\mathscr{B}$ — system's behavior, *i.e.*, set of trajectories

$\mathscr{B}|_L$ — restriction of $\mathscr{B}$ to the interval $[1, L]$

$w_\mathrm{d} := \big(w_\mathrm{d}(1), \dots, w_\mathrm{d}(T_\mathrm{d})\big)$ — "data" trajectory

$$\mathscr{H}_L(w_\mathrm{d}) := \begin{bmatrix} w_\mathrm{d}(1) & w_\mathrm{d}(2) & \cdots & w_\mathrm{d}(T_\mathrm{d} - L + 1) \\ \vdots & \vdots & & \vdots \\ w_\mathrm{d}(L) & w_\mathrm{d}(L+1) & \cdots & w_\mathrm{d}(T_\mathrm{d}) \end{bmatrix}$$

$\mathrm{PE}(u_\mathrm{d}) :=$ max L, such that $\mathscr{H}_L(u_\mathrm{d})$ is f.r.r.

We address the following issues/questions

proof by contradiction

What is the meaning/interpretation of the conditions?

sufficiency of the conditions

How conservative are they? Can they be improved?

conjecture

The extra PE of order n is generically not needed.
What are the nongeneric cases when it is needed?

Answers

constructive proof in the single-input case

$$\mathrm{PE}(u_\mathrm{d}) = n_u \iff u_\mathrm{d} \in \mathscr{B}_u|_{T_\mathrm{d}}, \text{ where } \mathscr{B}_u \text{ is autonomous LTI of order } n_u$$

shows that the FL is nonconservative

conjecture: it is conservative in the multi-input case

characterizes the nongeneric cases

they correspond to special initial conditions

Necessary and sufficient condition for the data-driven representation

$$\operatorname{rank} \mathscr{H}_L(w_{\mathrm{d}}) = mL + n, \qquad \text{(GPE)}$$

nonconservative (necessary and sufficient)

general no I/O partitioning and controllability

verifiable from w_{d} with prior knowledge of (m, n)

I. Markovsky and F. Dörfler, Identifiability in the Behavioral Setting, 2020

The fundamental lemma is input design result

input design problem

choose u_d, so that (DD-REPR) holds for any initial cond.

refined problem statement

find nonconservative conditions on u_{d} and $\mathscr{B}$, under which

$$\text{for } \forall\, w_{\text{d,ini}},\ w_{\text{d,ini}} \wedge w_{\text{d}} \in \mathscr{B}|_{T_{\text{ini}}+T_{\text{d}}} \text{ satisfies (GPE)} \qquad \text{(GOAL)}$$

subproblem: find w_{ini} that minimize rank $\mathscr{H}_L(w_{\text{d}})$

Obvious necessary conditions

A0: exact representation requires exact data and input design requires input/output partition

A1: for uncontrollable $\mathscr{B} = \mathscr{B}_{\text{ctr}} \oplus \mathscr{B}_{\text{aut}}$

- $w_{\text{d}} \in \mathscr{B} \implies w_{\text{d}} = w_{\text{d,ctr}} + w_{\text{d,aut}}$, $w_{\text{d,ctr}} \in \mathscr{B}_{\text{ctr}}$, $w_{\text{d,aut}} \in \mathscr{B}_{\text{aut}}$
- $w_{\text{d,aut}}$ is completely determined by $w_{\text{d,ini}}$
- there is $w_{\text{d,ini}}$, such that $w_{\text{d,aut}} = 0 \implies$ (GPE) doesn't hold

A2′: u_{d} is persistently exciting of order L

- since u is an input, $\Pi_u \mathscr{B}|_L = \mathbb{R}^{\mathbf{m}(\mathscr{B})L}$
- for (GPE) to hold true, image $\mathscr{H}_L(u_{\text{d}}) = \mathbb{R}^{\mathbf{m}(\mathscr{B})L}$
- equivalently, $\mathscr{H}_L(u_{\text{d}})$ must be full row-rank

Find the minimal k, such that (GOAL) holds under A0, A1, and $\mathrm{PE}(u_\mathrm{d}) = L + k$

first, we solve the subproblem

find w_{ini}^* that minimize rank $\mathscr{H}_L(w_d)$

then, we check (GPE) for w_ini^*

$\rightsquigarrow$ minimal $k \implies$ nonconservative PE condition

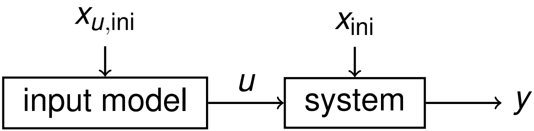
The PE condition is equivalent to existence of an LTI input model

$$u_{\text{d}} \in (\mathbb{R})^{T_{\text{d}}} \quad \text{and} \quad \text{PE}(u_{\text{d}}) = n_u$$

$$\Updownarrow$$

$$u_{\text{d}} \in \mathscr{B}_u|_{T_{\text{d}}} \text{ — autonomous LTI,} \quad T_{\text{d}} \geq 2n_u - 1$$

$$\mathscr{B}_u = \mathscr{B}_{\text{ss}}(A_u, C_u) \text{ with } \big(A_u, x_{u,\text{ini}}\big) \text{ controllable}$$



Augmented system with the input model

$\mathscr{B}_{\text{ext}} = \mathscr{B}_{\text{ss}}(A_{\text{ext}}, C_{\text{ext}})$, with $x_{\text{ext}} = \left[\begin{smallmatrix} x_u \\ x \end{smallmatrix}\right]$

$$A_{\text{ext}} = \begin{bmatrix} A_u & 0 \\ BC_u & A \end{bmatrix} \quad C_{\text{ext}} = \begin{bmatrix} C_u & 0 \\ DC_u & C \end{bmatrix}$$

$\mathscr{B}_{\text{ext}} = \mathscr{B}_{\text{ss}}\left(A'_{\text{ext}}, C'_{\text{ext}}\right)$, where $x'_{\text{ext}} = \left[\begin{smallmatrix} x_u \\ Vx_u + x \end{smallmatrix}\right]$

$$A'_{\text{ext}} = \begin{bmatrix} A_u & 0 \\ 0 & A \end{bmatrix}, \quad C'_{\text{ext}} = \begin{bmatrix} C_u & 0 \\ C' & C \end{bmatrix}, \quad C' := DC_u - CV$$

V is solution of the Sylvester equation $AV - VA_u = BC_u$

The nongeneric cases correspond to special initial conditions $x_{\text{ini}} = -Vx_{u,\text{ini}}$

which eliminates from w_{d} the transient due to $\mathscr{B}$

then, $\operatorname{rank} \mathscr{H}_L(w_{\text{d}}) \leq \text{PE}(u_{\text{d}}) = n_u$

next, we show that $\operatorname{rank} \mathscr{H}_L(w_{\text{d}}) = n_u$

assume simple eigenvalues $\lambda_{u,1}, \dots, \lambda_{u,n_u}$ of $\mathscr{B}_u$

$$u_\mathrm{d} = \sum_{i=1}^{n_u} a_i \exp_{\lambda_{u,i}}$$

assume simple eigenvalues $\lambda_1, \dots, \lambda_n$ of $\mathscr{B}$

$$y_\mathrm{d} = \sum_{i=1}^{n_u} b_i \exp_{\lambda_{u,i}} + \underbrace{\sum_{j=1}^{n} c_j \exp_{\lambda_j}}_{\text{transient}}$$

- $b_i = H(e^{\mathbf{i}\lambda_{u,i}})a_i$, where $H(z) := C(Iz - A)^{-1}B + D$
- $w_\mathrm{ini} = w_\mathrm{ini}^* \implies c_j = 0$

using Vandermonde matrix, we rewrite $(u_{\mathrm{d}}, y_{\mathrm{d}})$

$$u_{\mathrm{d}} = \underbrace{\begin{bmatrix} \lambda_{u,1}^{1} & \cdots & \lambda_{u,n_u}^{1} \\ \vdots & & \vdots \\ \lambda_{u,1}^{T} & \cdots & \lambda_{u,n_u}^{T} \end{bmatrix}}_{V_T(\lambda_u)} \underbrace{\begin{bmatrix} a_1 \\ \vdots \\ a_{n_u} \end{bmatrix}}_{a} = V_T(\lambda_u)a$$

and

$$y_{\mathrm{d}} = V_T(\lambda_u) \underbrace{\begin{bmatrix} H(e^{\mathbf{i}\lambda_{u,1}}) & & \\ & \ddots & \\ & & H(e^{\mathbf{i}\lambda_{u,n_u}}) \end{bmatrix}}_{H(\lambda_u)} \begin{bmatrix} a_1 \\ \vdots \\ a_{n_u} \end{bmatrix}$$

$$= V_T(\lambda_u) \underbrace{H(\lambda_u)a}_{b} = V_T(\lambda_u)b$$

then, for w_d, we obtain

$$w_\text{d} = \Pi_T \begin{bmatrix} V_T(\lambda_u) \\ V_T(\lambda_u) H(\lambda_u) \end{bmatrix} a$$

$\Pi_T \in \mathbb{R}^{2T\times 2T}$ permutation, such that $w_\text{d} = \Pi_T \left[\begin{smallmatrix} u_\text{d} \\ y_\text{d} \end{smallmatrix}\right]$

finally, the Hankel matrix is expressed as

$$\mathscr{H}_L(w_\text{d}) = \underbrace{\Pi_L \begin{bmatrix} V_L(\lambda_u) \\ V_L(\lambda_u) H(\lambda_u) \end{bmatrix}}_{W_L} \underbrace{\begin{bmatrix} a & \Lambda_u a & \Lambda_u^2 a & \cdots & \Lambda_u^{T-L} a \end{bmatrix}}_{\text{controllability matrix of } (\Lambda_u, a)}$$

$$\Lambda_u := \operatorname{diag}(\lambda_{u,1}, \ldots, \lambda_{u,n_u})$$

(Λ_u, a) is controllable because $\mathrm{PE}(u_\mathrm{d}) = n_u$

1. $a_i \neq 0$ for all i
2. $\lambda_{u,i} \neq \lambda_{u,j}$ for all $i \neq j$

for $k \leq n$, W_L is full column rank

- with $W_L = \begin{bmatrix} w^1 & \dots & w^{n_u} \end{bmatrix}$, w^i are trajectories ($w^i \in \mathscr{B}|_L$)
- $\lambda_{u,i} \neq \lambda_{u,j}$ for all $i \neq j \quad \Longrightarrow \quad$ independent responses

$$\operatorname{rank} \mathscr{H}_L(w_\mathrm{d}) = \begin{cases} L + k, & \text{for } k = 1, \dots, n \\ L + n, & \text{for } k = n+1, \dots \end{cases}$$

$k = n$ is the minimal value for (GPE) to hold

Comments

the zeros of $\mathscr{B}$ don't play role in the analysis

simple eigenvalues assumptions can be relaxed

"robustifying" the conditions

exact condition:	robust version:
$a_i \neq 0$, for all i	$a_i > \varepsilon$
$\lambda_{u,i} \neq \lambda_{u,j}$, for all $i \neq j$	the $\lambda_{u,i}$'s are "well spread"

conjecture: in multi-input case, A2 can be tightened, $\mathrm{PE}(u_\mathrm{d}) = n +$ controllability index $\mathscr{B}$

Outline

Constructive proof of the fundamental lemma

Pedagogical example: Free fall prediction

Case study: Dynamic measurement

Nonparametric frequency response estimation

Generalization for nonlinear systems

The goal is to predict free fall trajectory without knowing the laws of physics

object with mass m, falling in gravitational field

- y — position
- $v := \dot{y}$ — velocity
- $y(0), v(0)$ — initial condition

task: given initial condition, find the trajectory y

- model-based approach:
  1. physics $\mapsto$ model
  2. model + ini. cond. $\mapsto y$
- data-driven approach: data $y_{\mathrm{d}}^1, \ldots, y_{\mathrm{d}}^N$ + ini. cond. $\mapsto y$

Modeling from first principles leads to affine time-invariant state-space model

second law of Newton $+$ the law of gravity

$$m\ddot{y} = m\left[\begin{smallmatrix} 0 \\ 9.81 \end{smallmatrix}\right] + f, \quad \text{where} \quad y(0) = y_{\text{ini}} \text{ and } \dot{y}(0) = v_{\text{ini}}$$

- 9.81 — gravitational constant
- $f = -\gamma v$ — force due to friction in the air

state $x := (y_1, \dot{y}_1, y_2, \dot{y}_2, x_5)$, where $x_5 = -9.81$

initial state $x_{\text{ini}} := (y_{\text{ini},1}, v_{\text{ini},1}, y_{\text{ini},2}, v_{\text{ini},2}, -9.81)$

Modeling from first principles leads to affine time-invariant state-space model

$$\dot{x} = \begin{bmatrix} 0 & 1 & & & \\ 0 & -\gamma/m & & & \\ & & 0 & 1 & \\ & & 0 & -\gamma/m & 1 \\ & & & & 0 \end{bmatrix} x, \qquad x(0) = \begin{bmatrix} y_{\text{ini},1} \\ v_{\text{ini},1} \\ y_{\text{ini},2} \\ v_{\text{ini},2} \\ -9.81 \end{bmatrix}$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} x$$

data: N, T-samples long discretized trajectories

Simulation setup and data

write a function `fall` that simulates free fall

```
y = fall(y0, v0, t, m, gamma)
```

simulate `N=10`, `T=100`-samples long trajectories

```
m = 1; gamma = 0.5;
N = 10; T = 100; t = linspace(0, 1, T);
for i = 1:N,
  y{i} = fall(rand(2,1), rand(2,1), t,gamma,m);
end
```

and to-be-predicted trajectory

```
y_new = fall(rand(2,1), rand(2,1), t,gamma,m);
```

Data-driven free fall prediction method

data "informativity" condition:

$$\operatorname{rank} \underbrace{\begin{bmatrix} y_\mathrm{d}^1 & \cdots & y_\mathrm{d}^N \end{bmatrix}}_{D} = 5$$

algorithm for data-driven prediction:

1. solve $$\begin{bmatrix} y_\mathrm{d}^1(1) & \cdots & y_\mathrm{d}^N(1) \\ y_\mathrm{d}^1(2) & \cdots & y_\mathrm{d}^N(2) \\ y_\mathrm{d}^1(3) & \cdots & y_\mathrm{d}^N(3) \end{bmatrix} g = \underbrace{\begin{bmatrix} y(1) \\ y(2) \\ y(3) \end{bmatrix}}_{\text{ini. cond.}}$$

2. define $y := Dg$

Verify that the data-driven prediction “works”

check the data “informativity” condition

```
[rank(D) rank([vec(y_new') D])] % -> [ 5 5 ]
```

implement the data-driven computation method

verify the computed solution

Summary: prediction of free fall trajectory

first principles modeling

- use the second law of Newton and the law of gravity
- in particular, the Earth's gravitational constant is used
- lead to an autonomous affine time-invariant system

data-driven methods

- bypass the knowledge of the physical laws
- automatically infer and use them
- no hyper-parameters to tune

Outline

My interest in dynamic measurement started from a textbook problem

"A thermometer reading $21^\circ C$*, which has been inside a house for a long time, is taken outside. After one minute the thermometer reads* $15^\circ C$*; after two minutes it reads* $11^\circ C$*. What is the outside temperature?"*

According to Newton's law of cooling, an object of higher temperature than its environment cools at a rate that is proportional to the difference in temperature.

Main idea: predict the steady-state value from the first few samples of the transient

textbook problem:

- 1st order dynamics
- 3 noise-free samples
- batch solution

generalizations:

- $n \geq 1$ order dynamics
- $T \geq 3$ noisy (vector) samples
- recursive computation

implementation and practical validation

Thermometer: first order dynamical system

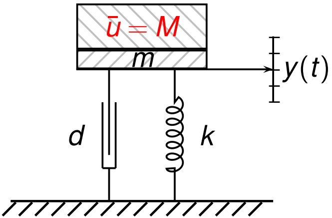
$$\text{environmental temperature } \bar{u} \xrightarrow{\text{heat transfer}} \text{thermometer's reading } y$$

measurement process: Newton's law of cooling

$$y = a(\bar{u} - y)$$

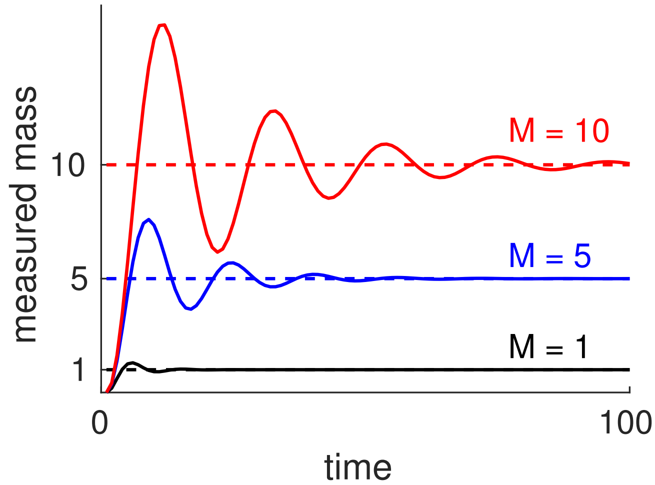
heat transfer coefficient $a > 0$

Scale: second order dynamical system



$$(M+m)\frac{\mathrm{d}}{\mathrm{d}\,t}y + dy + ky = g\bar{u}$$

The measurement process dynamics depends on the to-be-measured mass



Dynamic measurement: take into account the dynamical properties of the sensor

$$\begin{matrix}\text{to-be-measured} \\ \text{variable } u\end{matrix} \quad \xrightarrow{\text{measurement process}} \quad \begin{matrix}\text{measured} \\ \text{variable } y\end{matrix}$$

assumption 1: measured variable is constant $u(t) = \bar{u}$

assumption 2: the sensor is stable LTI system

assumption 3: sensor's DC-gain $= 1$ (calibrated sensor)

The data is generated from LTI system with output noise and constant input

$$\underbrace{y_{\mathrm{d}}}_{\substack{\text{measured}\\ \text{data}}} = \underbrace{y}_{\substack{\text{true}\\ \text{value}}} + \underbrace{e}_{\substack{\text{measurement}\\ \text{noise}}}$$

$$\underbrace{y}_{\substack{\text{true}\\ \text{value}}} = \underbrace{\bar{u}}_{\substack{\text{steady-state}\\ \text{value}}} + \underbrace{y_0}_{\substack{\text{transient}\\ \text{response}}}$$

assumption 4: e is a zero mean, white, Gaussian noise using a state space representation of the sensor

$$
\begin{aligned}
x(t+1) &= Ax(t), \qquad x(0) = x_0 \\
y_0(t) &= cx(t)
\end{aligned}
$$

we obtain

$$
\underbrace{\begin{bmatrix} y_{\mathrm{d}}(1) \\ y_{\mathrm{d}}(2) \\ \vdots \\ y_{\mathrm{d}}(T) \end{bmatrix}}_{y_{\mathrm{d}}} = \underbrace{\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}}_{\mathbf{1}_T} \bar{u} + \underbrace{\begin{bmatrix} c \\ cA \\ \vdots \\ cA^{T-1} \end{bmatrix}}_{\mathscr{O}_T} x_0 + \underbrace{\begin{bmatrix} e(1) \\ e(2) \\ \vdots \\ e(T) \end{bmatrix}}_{e}
$$

Maximum-likelihood model-based estimator

solve approximately

$$\begin{bmatrix} \mathbf{1}_T & \mathscr{O}_T \end{bmatrix} \begin{bmatrix} \widehat{u} \\ \widehat{x}_0 \end{bmatrix} \approx y_\mathrm{d}$$

standard least-squares problem

$$\begin{aligned} &\text{minimize} \quad \text{over } \widehat{y}, \widehat{u}, \widehat{x}_0 \quad \|y_\mathrm{d} - \widehat{y}\| \\ &\text{subject to} \quad \begin{bmatrix} \mathbf{1}_T & \mathscr{O}_T \end{bmatrix} \begin{bmatrix} \widehat{u} \\ \widehat{x}_0 \end{bmatrix} = \widehat{y} \end{aligned}$$

recursive implementation $\rightsquigarrow$ Kalman filter

Subspace model-free method

goal: avoid using the model parameters (A, C, $\mathscr{O}_T$)

in the noise-free case, due to the LTI assumption,

$$\Delta y(t) := y(t) - y(t-1) = y_0(t) - y_0(t-1)$$

satisfies the same dynamics as y_0, *i.e.*,

$$\begin{aligned} x(t+1) &= Ax(t), \qquad x(0) = \Delta x \\ \Delta y(t) &= cx(t) \end{aligned}$$

Hankel matrix—construction of multiple "short" trajectories from one "long" trajectory

$$\mathscr{H}(\Delta y) := \begin{bmatrix} \Delta y(1) & \Delta y(2) & \cdots & \Delta y(\mathtt{n}) \\ \Delta y(2) & \Delta y(3) & \cdots & \Delta y(\mathtt{n}+1) \\ \Delta y(3) & \Delta y(4) & \cdots & \Delta y(\mathtt{n}+2) \\ \vdots & \vdots & & \vdots \\ \Delta y(T-\mathtt{n}) & \Delta y(T-\mathtt{n}) & \cdots & \Delta y(T-1) \end{bmatrix}$$

fact: if $\operatorname{rank} \mathscr{H}(\Delta y) = \mathtt{n}$, then

$$\operatorname{image} \mathscr{O}_{T-\mathtt{n}} = \operatorname{image} \mathscr{H}(\Delta y)$$

model-based equation

$$\begin{bmatrix} \mathbf{1}_T & \mathscr{O}_T \end{bmatrix} \begin{bmatrix} \bar{u} \\ \widehat{x}_0 \end{bmatrix} = y$$

data-driven equation

$$\begin{bmatrix} \mathbf{1}_{T-\mathtt{n}} & \mathscr{H}(\Delta y) \end{bmatrix} \begin{bmatrix} \bar{u} \\ \ell \end{bmatrix} = y|_{T-\mathtt{n}} \qquad (*)$$

subspace method

solve $(*)$ by (recursive) least squares

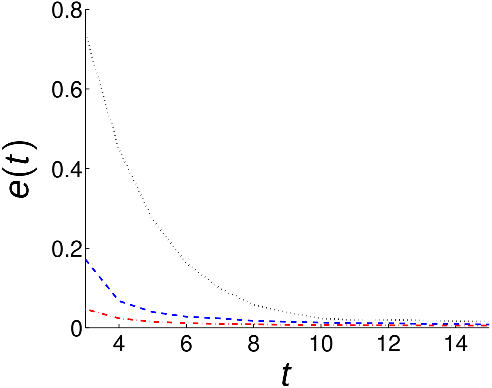
Empirical validation

dashed	—	true parameter value $\bar{u}$
solid	—	true output trajectory y_0
dotted	—	naive estimate $\widehat{u} = G^{+} y$
dashed	—	model-based Kalman filter
bashed-dotted	—	data-driven method

estimation error: $e := \frac{1}{N} \sum_{i=1}^{N} \|\bar{u} - \widehat{u}^{(i)}\|$

(for $N = 100$ Monte-Carlo repetitions)

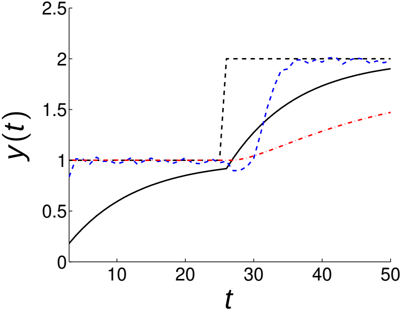
Simulated data of dynamic cooling process

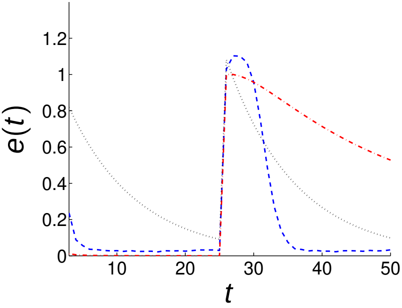


$e(t) \to 0$ as $t \to \infty$ at different rates

best is the Kalman filter (maximum likelihood estimator)

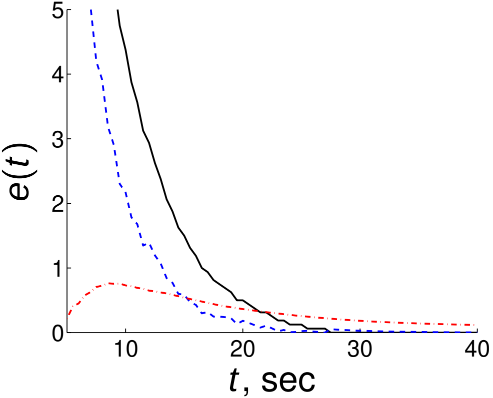
Simulation with time-varying parameter





Proof of concept prototype

Results in real-life experiment



Summary

dynamic measurement

steady-state value prediction

the subspace method is applicable for

- high order dynamics
- noisy vector observations
- online computation

future work / open problems

- numerical efficiency
- real-time uncertainty quantification
- generalization to nonlinear systems

Outline

Constructive proof of the fundamental lemma

Pedagogical example: Free fall prediction

Case study: Dynamic measurement

Nonparametric frequency response estimation

Generalization for nonlinear systems

Problem formulation

given: "data" trajectory $(u_\mathrm{d}, y_\mathrm{d}) \in \mathscr{B}|_{T_\mathrm{d}}$ and $z \in \mathbb{C}$

find: $H(z)$, where H is the transfer function of $\mathscr{B}$

Direct data-driven solution

we are interested in trajectory

$$w = \begin{bmatrix} u \\ y \end{bmatrix} = \begin{bmatrix} \exp_z \\ \widehat{H} \exp_z \end{bmatrix} \in \mathscr{B}, \quad \text{where } \exp_z(t) := z^t$$

using the data-driven representation, we have

$$\begin{bmatrix} \mathscr{H}_L(u_\mathrm{d}) \\ \mathscr{H}_L(y_\mathrm{d}) \end{bmatrix} g = \begin{bmatrix} \mathbf{z} \\ \widehat{H}\mathbf{z} \end{bmatrix}, \quad \text{where } \mathbf{z} := \begin{bmatrix} z^1 \\ \vdots \\ z^L \end{bmatrix}$$

which leads to the system

$$\begin{bmatrix} 0 & \mathscr{H}_L(u_\mathrm{d}) \\ -\mathbf{z} & \mathscr{H}_L(y_\mathrm{d}) \end{bmatrix} \begin{bmatrix} \widehat{H} \\ g \end{bmatrix} = \begin{bmatrix} \mathbf{z} \\ 0 \end{bmatrix} \qquad \text{(SYS)}$$

Solution method: solve (SYS) for $\widehat{H}$

under (GPE) with $L \geq \ell + 1$, $\widehat{H} = H(z)$

without prior knowledge of ℓ

$$L = L_{\max} := \lfloor (T_{\mathrm{d}} + 1)/3 \rfloor$$

trivial generalization to

- multivariable systems
- multiple data trajectories $\{ w_{\mathrm{d}}^1, \ldots, w_{\mathrm{d}}^N \}$
- evaluation of $H(z)$ at multiple points in $\{ z_1, \ldots, z_K \} \in \mathbb{C}^K$

Comparison with classical nonparametric frequency response estimation methods

ignored initial/terminal conditions $\rightsquigarrow$ *leakage*

DFT grid $\rightsquigarrow$ limited *frequency resolution*

improvements by windowing and interpolation

- the leakage is not eliminated
- the methods involve *hyper-parameters*

Generalization of (SYS) to noisy data

preprocessing: rank-$mL+n$ approx. of $\mathscr{H}_L(w_\mathrm{d})$

- hyper-parameters $L \geq \ell + 1$ and n
- if the approximation preserves the Hankel structure, the method is maximum-likelihood in the EIV setting

regularization with $\|g\|_1$

- hyper-parameter: the 1-norm regularization parameter

regularization with the nuclear norm of $\mathscr{H}_L(\widehat{w}_\mathrm{d})$

- hyper-parameters: L and the regularization parameter

Matlab implementation

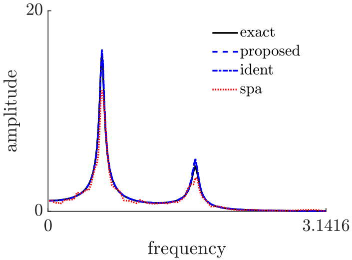
```
function Hh = dd_frest(ud, yd, z, n)
L = n + 1; t = (1:L)';
m = size(ud, 2); p = size(yd, 2);

%% preprocessing by low-rank approximation
H = [moshank(ud, L); moshank(yd, L)];
[U, ~, ~] = svd(H); P = U(:, 1:m * L + n);

%% form and solve the system of equations
for k = 1:length(z)
  A  = [[zeros(m*L, p); -kron(z(k).^t, eye(p))] P];
  hg = A \ [kron(z(k).^t, eye(m)); zeros(p*L, m)];
  Hh(:, :, k) = hg(1:p, :);
end
```

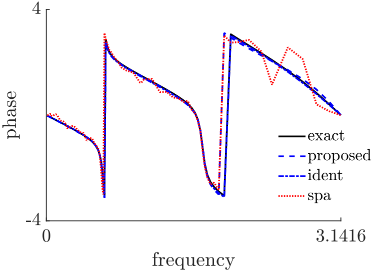
- effectively 5 lines of code
- MIMO case, multiple evaluation points
- $L = n + 1$ in order to have a single hyper-parameter

Example: EIV setup with 4th order system

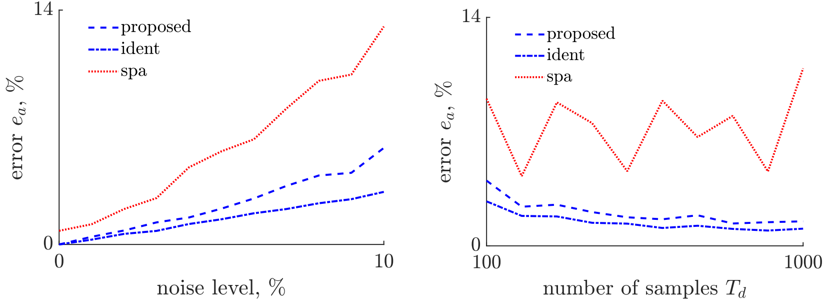
`dd_frest` is compared with

- `ident` — parametric maximum-likelihood estimator
- `spa` — nonparameteric estimator with Welch filter





Monte-Carlo simulation over different noise levels and number of samples



$$e_a := 100\% \cdot |(|\overline{H}_z| - |\widehat{H}_z|)| \,/\, |\overline{H}_z|$$

Outline

Kernel representation

LTI systems

$$\begin{aligned}\mathscr{B} &= \ker R(\sigma) := \{\, w \mid R(\sigma)w = 0 \,\}\\ &= \{\, w \mid R_0 w + R_1 \sigma w + \cdots + R_\ell \sigma^\ell w = 0 \,\}\end{aligned}$$

nonlinear time-invariant system

$$\mathscr{B} = \Big\{\, w \mid R(\underbrace{w, \sigma w, \ldots, \sigma^\ell w}_{x}) = 0 \,\Big\}$$

linearly parameterized R

$$R(x) = \sum \theta_i \phi_i(x) = \theta^\top \phi(x),$$

ϕ	—	model structure
θ	—	parameter vector

Polynomial SISO NARX system

$$\mathscr{B}(\theta) = \left\{ w = \left[\begin{smallmatrix} u \\ y \end{smallmatrix}\right] \mid y = f\big(u, \sigma w, \dots, \sigma^{\ell} w\big) \right\}$$

split f into 1st order (linear) and other (nonlinear) terms

$$f(x) = \theta_{\text{li}}^{\top} x + \theta_{\text{nl}}^{\top} \phi_{\text{nl}}(x)$$

ϕ_{nl} — vector of monomials

Special cases

Hammerstein

$$\phi_{\mathrm{nl}}(x) = \begin{bmatrix} \phi_u(u) & \phi_u(\sigma u) & \cdots & \phi_u(\sigma^\ell u) \end{bmatrix}^\top$$

FIR Volterra

$$\phi_{\mathrm{nl}}(x) = \phi_{\mathrm{nl}}(x_u), \quad \text{where } x_u := \mathrm{vec}(u, \sigma u, \ldots, \sigma^\ell u).$$

bilinear

$$\phi_{\mathrm{nl}}(x) = x_u \otimes x_y, \quad \text{where } x_y := \mathrm{vec}(y, \sigma y, \ldots, \sigma^{\ell-1} y)$$

generalized bilinear

$$\phi_{\mathrm{nl}}(x) = \phi_{u,\mathrm{nl}}(x_u) \otimes x_y$$

LTI embedding of polynomial NARX system

$$\mathscr{B}_{\text{ext}}(\theta) := \left\{ w_{\text{ext}} = \begin{bmatrix} u \\ u_{\text{nl}} \\ y \end{bmatrix} \;\middle|\; \sigma^{\ell} y = \theta_{\text{li}}^{\top} x + \theta_{\text{nl}}^{\top} u_{\text{nl}} \right\}$$

define: $\Pi_w w_{\text{ext}} := w$ and $\Pi_{u_{\text{nl}}} w_{\text{ext}} := u_{\text{nl}}$

fact: $\mathscr{B}(\theta) \subseteq \Pi_w \mathscr{B}_{\text{ext}}(\theta)$, moreover

$$\mathscr{B}(\theta) = \Pi_w \left\{ w_{\text{ext}} \in \mathscr{B}_{\text{ext}}(\theta) \mid \Pi_{u_{\text{nl}}} w_{\text{ext}} = \phi_{\text{nl}}(x) \right\}$$

FIR Volterra data-driven simulation

given

data $w_{\mathrm{d}} = (u_{\mathrm{d}}, y_{\mathrm{d}})$ of lag-ℓ FIR Volterra system $\mathscr{B}$
ϕ_{nl} — system's model structure

assume ID conditions for $\mathscr{B}_{\mathrm{ext}}$ hold

then, $\mathscr{B}|_L = \operatorname{image} M$, where

$$M(w_{\mathrm{ini}}, u) := \mathscr{H}_L(\sigma^{\ell} y_{\mathrm{d}}) \underbrace{\begin{bmatrix} \mathscr{H}_{\ell}(w_{\mathrm{d}}) \\ \mathscr{H}_L(\sigma^{\ell} u_{\mathrm{d}}) \\ \hline \mathscr{H}_{\ell}\big(\phi_{\mathrm{nl}}(x_{u_{\mathrm{d}}})\big) \\ \mathscr{H}_L\big(\sigma^{\ell} \phi_{\mathrm{nl}}(x_{u_{\mathrm{d}}})\big) \end{bmatrix}^{\dagger} \begin{bmatrix} w_{\mathrm{ini}} \\ u \\ \hline \phi_{\mathrm{nl}}(x_{u_{\mathrm{ini}}}) \\ \phi_{\mathrm{nl}}(x_u) \end{bmatrix}}_{g}$$

proof

$$
\left[\begin{array}{c}
\mathscr{H}_\ell(w_\mathrm{d}) \\
\mathscr{H}_L(\sigma^\ell u_\mathrm{d}) \\ \hline
\mathscr{H}_\ell\big(\phi_\mathrm{nl}(x_{u_\mathrm{d}})\big) \\
\mathscr{H}_L\big(\sigma^\ell \phi_\mathrm{nl}(x_{u_\mathrm{d}})\big) \\ \hline
\color{red}{\mathscr{H}_L(\sigma^\ell y_\mathrm{d})}
\end{array}\right] g =
\left[\begin{array}{c}
w_\mathrm{ini} \\
u \\ \hline
\phi_\mathrm{nl}(x_{u_\mathrm{ini}}) \\
\phi_\mathrm{nl}(x_u) \\ \hline
\color{red}{y}
\end{array}\right]
\begin{array}{l}
\left.\vphantom{\begin{array}{c} a \\ a \end{array}}\right\} \text{B1} \\
\left.\vphantom{\begin{array}{c} a \\ a \end{array}}\right\} \text{B2} \\
\left.\vphantom{a}\right\} \text{B3}
\end{array}
$$

B1 constraint on g, such that $w_\mathrm{ini} \wedge \big(u, \mathscr{H}_L(\sigma^\ell y_\mathrm{d}) g\big) \in \mathscr{B}_\mathrm{ext}$

B2 constraint $u_\mathrm{nl} = \phi_\mathrm{nl}(x) \iff \mathscr{B}_\mathrm{ext} = \mathscr{B}(\theta)$

B3 defines the to-be-computed output y

generalized bilinear models

also tractable because B2: $u_\mathrm{nl} = \phi_\mathrm{nl}(x)$ is still linear in y