Data-driven systems theory, signal processing, and control

Ivan Markovsky

ICREA

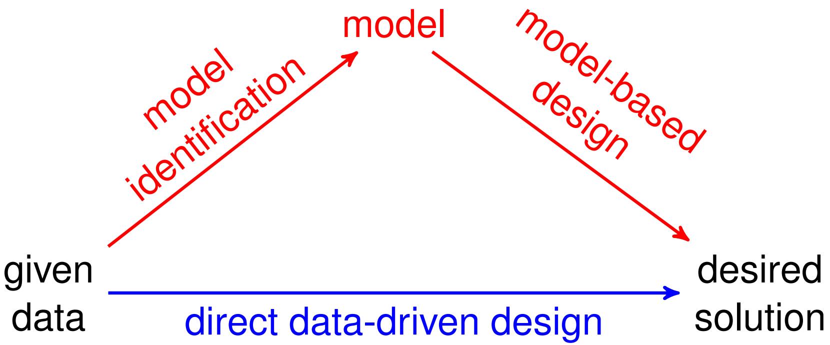
CIMNE
INTERNATIONAL CENTRE FOR NUMERICAL METHODS IN ENGINEERING

The problems and methods reviewed today differ from those you've learned yesterday

yesterday:	object	parametric PDEs
	problem	given $\{\xi_i, u_i\}$, find $\xi \mapsto u$
	approach	neural network

today:	object	linear time-invariant systems
	problem	given u, predict, filter, control
	approach	behavioral systems theory

We are aiming at direct data-driven methods for analysis and design of dynamical systems



the classical approach is “indirect data-driven”

Data-driven does not mean model-free

data-driven methods make model assumptions

but don't use *parametric representations*

they are *non-parametric* using directly the data

Outline

Example: Free fall prediction

Linear time-invariant systems

Data-driven representation

Dealing with noise

The goal is to predict free fall trajectory

object with mass m, falling in gravitational field

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

task: given initial condition, find the trajectory w

- model-based approach:
  1. physics $\mapsto$ parametric model
  2. model parameter estimation
  3. model + ini. conditions $\mapsto w$
- data-driven approach: data $w_{\mathrm{d}}^1, \dots, w_{\mathrm{d}}^N$ + ini. cond. $\mapsto w$

Modeling from first principles yields affine time-invariant dynamical system

second law of Newton + the law of gravity

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

- 9.81 — gravitational constant
- $f = -\gamma\dot{w}$ — force due to friction in the air

1st order equation

$$\dot{x} = Ax, \quad w = Cx, \quad x(0) = x_{\text{ini}}$$

- state $x := (w_1, \dot{w}_1, w_2, \dot{w}_2, -9.81)$
- initial state $x_{\text{ini}} := (w_{\text{ini},1}, v_{\text{ini},1}, w_{\text{ini},2}, v_{\text{ini},2}, -9.81)$
- A, C — model parameters (depend on m and γ)

Data-driven free fall prediction method

data: N, discrete-time trajectories $w_{\rm d}^1, \ldots, w_{\rm d}^N$

$$\operatorname{rank} \begin{bmatrix} w_{\rm d}^1 & \cdots & w_{\rm d}^N \end{bmatrix} = 5 \qquad \text{"informativity" condition}$$

algorithm for data-driven prediction:

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

2. define $w := \begin{bmatrix} w_{\rm d}^1 & \cdots & w_{\rm d}^N \end{bmatrix} g$

Summary: prediction of free fall trajectory

first principles modeling

- use Newton's 2nd law, law of gravity, and friction
- and model parameters m, γ, gravitational constant
- lead to autonomous affine time-invariant system

data-driven approach

- bypasses the knowledge of the physical laws
- and prior knowledge or estimation of model parameters
- no hyper-parameters to tune

The exercises are linked to the lectures, they are an integral part of the course

"I hear, I forget; I see, I remember; I do, I understand."

your task

1. write a function for model-based free fall simulation
2. collect free falls data $w_d^1, \ldots, w_d^N$ using the model
3. implement and try the direct data-driven method

if you have questions

- option 1: use the "raise hand" function
- option 2: post them in the chat

Signals are functions of time

$(\mathbb{R}^q)^{\mathscr{T}}$ — signal space: functions $\mathscr{T} \mapsto (\mathbb{R}^q)$

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

$w(t) \in \mathbb{R}^q$ is the value of w at time $t \in \mathscr{T}$

Signals are classified according to # of variables q and type of time axis $\mathscr{T}$

$q = 1$ — scalar signal
$q > 1$ — vector signal

$\mathscr{T} = \mathbb{R}$ — continuous-time
$\mathscr{T} = \mathbb{Z}$ — discrete-time

$(\mathbb{R}^q)^{\mathbb{R}} \mapsto (\mathbb{R}^q)^{\mathbb{Z}}$ — sampling / time-discretization

Signals are transformed by operators

$(\sigma w)(t) := w(t+1)$ — unit-shift operator

$\underbrace{R_0 + R_1\sigma + \ldots + R_\ell \sigma^\ell}_{R(\sigma)}$ — polynomial operator

$w|_{[t_1,t_2]}$ and $w|_T$ — restriction to interval

The classical view of dynamical system is a "signal processor": an input/output map

accepts input signal and produces output signal

intuition: the input *causes* the output

In the behavioral approach to systems theory, dynamical system is a set of signals

$\mathscr{B} \subset (\mathbb{R}^q)^{\mathbb{Z}}$ — q-variate discrete-time system

- $q = 1$ — scalar system
- $q > 1$ — multivariable system

$w \in \mathscr{B}$ — w is a trajectory of $\mathscr{B}$

- w is allowed/predicted by $\mathscr{B}$
- $\mathscr{B}$ is unfalsified by w

$\mathscr{B}|_T$ — restriction of $\mathscr{B}$ to the interval $1, \ldots, T$

$\mathscr{B} = \{ w \mid f(w) = 0 \}$ is a representation of $\mathscr{B}$

a given $\mathscr{B}$ allows different representations

- parametric vs non-parametric representations
- uniqueness of the parameters
- how to switch from one representation to another?

different representations $\rightsquigarrow$ different methods

problems related to a system $\mathscr{B}$:

- $\mathscr{B} \mapsto w$ — simulation
- $w_d \mapsto \mathscr{B}$ — identification
- noise filtering, prediction, control, ...

Example: free fall in gravitational field

$w \in (\mathbb{R}^2)^{\mathbb{R}_+}$ — object's position ($q = 2$, $\mathscr{T} = \mathbb{R}_+$)

$\mathscr{B} \subset (\mathbb{R}^2)^{\mathbb{R}_+}$ — all possible free-fall trajectories the object may have

representations

$$\mathscr{B} = \left\{ w \in (\mathbb{R}^2)^{\mathbb{R}_+} \mid m\ddot{w} = m\left[\begin{smallmatrix} 0 \\ -9.81 \end{smallmatrix}\right] - \gamma\dot{w},\ \begin{bmatrix} w(0) \\ \dot{w}(0) \end{bmatrix} \in \mathbb{R}^4 \right\}$$

$$= \left\{ w \in (\mathbb{R}^2)^{\mathbb{R}_+} \mid \text{there is } x \in (\mathbb{R}^5)^{\mathbb{R}_+}, \text{ such that } \dot{x} = Ax,\ w = Cx,\ x_5(0) = -9.81 \right\}$$

Linearity, time-invariance, and complexity are defined in terms of the set $\mathscr{B}$

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

$$\mathscr{B} \text{ is time-invariant} \quad :\Longleftrightarrow \quad \sigma\mathscr{B} = \mathscr{B}$$

$\mathscr{L}^q$ linear time-invariant (LTI) model class

- $\mathbf{m}(\mathscr{B})$ — number of inputs
- $\boldsymbol{\ell}(\mathscr{B})$ — lag
- $\mathbf{n}(\mathscr{B})$ — order

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

Kernel representation $\mathscr{B} = \ker R(\sigma)$ is ℓ-th order vector difference equation

$$\left\{\, w \mid R_0 w(t) + R_1 w(t+1) + \cdots + R_\ell w(t+\ell) = 0, \text{ for all } t \in \mathscr{T} \,\right\}$$

$$\Updownarrow$$

$$\left\{\, w \mid \underbrace{R_0 w + R_1 \sigma w + \cdots + R_\ell \sigma^\ell}_{R(\sigma)} w = 0 \,\right\}$$

$$\Updownarrow$$

$$\ker R(\sigma) \qquad \text{(KER)}$$

the parameter is a polynomial matrix $R(z) \in \mathbb{R}^{g \times q}[z]$

Input/state/output representation is 1-st order vector difference equation

$$\left\{ w = \Pi \begin{bmatrix} u \\ y \end{bmatrix} \;\middle|\; \text{there is } x \in (\mathbb{R}^n)^{\mathbb{N}}, \text{ such that } \sigma x = Ax + Bu,\ y = Cx + Du \right\} \quad \text{(I/S/O)}$$

x	—	state	,	n	$:=$	$\dim x$	—	order
u	—	input	,	m	$:=$	$\dim u$	—	# of inputs
y	—	output	,	p	$:=$	$\dim y$	—	# of outputs

the parameters are:

- permutation matrix $\Pi \in \mathbb{R}^{q \times q}$ and
- matrices $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$, $D \in \mathbb{R}^{p \times m}$

Summary: linear time-invariant systems

$w \in (\mathbb{R}^q)^{\mathscr{T}}$ signals are functions of time

$\mathscr{B} \subset (\mathbb{R}^q)^{\mathbb{Z}}$ systems are sets of signals

$\mathscr{B}$ can be represented by different equations

$\mathscr{L}^q$ LTI model class: shift-invariant subspaces

- complexity: (# of inputs, lag, order)
- $\mathscr{B} = \ker R(\sigma)$ kernel representation
- input/state/output representation

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

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

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

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 linear subspace of $(\mathbb{R}^q)^L$

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

collect $N \geq qL$ random trajectories

$$w_d^1, \ldots, w_d^N \in \mathscr{B}|_L$$

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

$$\text{span}\left\{ w_d^1, \ldots, w_d^N \right\} \subseteq \mathscr{B}|_L$$

with probability one equality holds

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

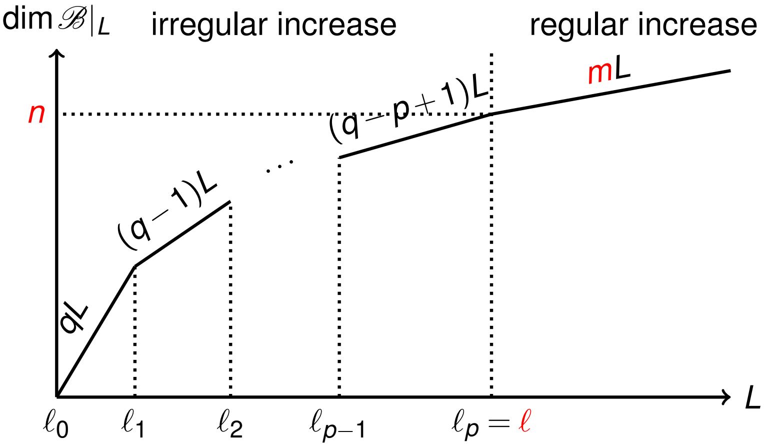
$$\underbrace{\begin{bmatrix} w_\mathrm{d}^1 & \cdots & w_\mathrm{d}^N \end{bmatrix}}_{W} \in \mathbb{R}^{qL \times N}$$ — "trajectory matrix"

$\widehat{\mathscr{B}}|_L = \text{image}\, W$ — data-driven representation

now, we can do explorations, in particular check

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

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



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

Data-driven representation (infinite horizon)

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

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

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

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

$\sigma^0 w_d$	$\sigma^1 w_d$	$\cdots$	$\sigma^{T_d-1} w_d$
$w_d(1)$	$w_d(2)$	$\cdots$	$w_d(T_d)$
$w_d(2)$	$\vdots$	$\iddots$	?
$\vdots$	$w_d(T_d)$	$\iddots$	$\vdots$
$w_d(T_d)$	?	$\cdots$	?

for $w_d = \big(w_d(1), \ldots, w_d(T_d)\big)$ and $1 \leq L \leq T_d$

$$\mathscr{H}_L(w_d) := \Big[(\sigma^0 w_d)|_L \;\; (\sigma^1 w_d)|_L \;\; \cdots \;\; (\sigma^{T_d-L} w_d)|_L \Big]$$

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

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

GPE — generalized persistency of excitation

Identifiability condition verifiable from $w_{\mathrm{d}} \in \mathscr{B}|_{T_{\mathrm{d}}}$ 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 $w_{\mathrm{d}} \in \mathscr{B}|_{T_{\mathrm{d}}}$ if and only if

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

$w_{\mathrm{d}} \mapsto \mathscr{B}$ — system identification

Generic data-driven problem: trajectory interpolation/approximation

given:

"data trajectory"	$w_{\mathrm{d}} \in \mathscr{B}\|_{T_{\mathrm{d}}}$
and elements	$w\|_{I_{\mathrm{given}}}$
of a trajectory	$w \in \mathscr{B}\|_T$

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

aim:

$$\text{minimize} \quad \text{over } \widehat{w} \quad \| w|_{I_{\mathrm{given}}} - \widehat{w}|_{I_{\mathrm{given}}} \| \qquad \text{subject to} \quad \widehat{w} \in \mathscr{B}|_T$$

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

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 $w_d^1, \ldots, w_d^N$

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

w_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 representation

assuming $\quad \operatorname{rank} \mathscr{H}_L(w_\mathrm{d}) = \mathbf{m}(\mathscr{B})L + \mathbf{n}(\mathscr{B})$

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

replaces parametric representations

The data w_d being exact vs inexact / “noisy”

w_d exact and satisfying (GPE)

- “systems 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: $w_\mathrm{d} = \overline{w}_\mathrm{d} + \widetilde{w}_\mathrm{d}$

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

ML problem: given w_d, c, and $w|_{I_\mathrm{given}}$

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

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

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

$$\Updownarrow$$

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

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)

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

Empirical validation on real-life datasets

	data set name	T_d	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

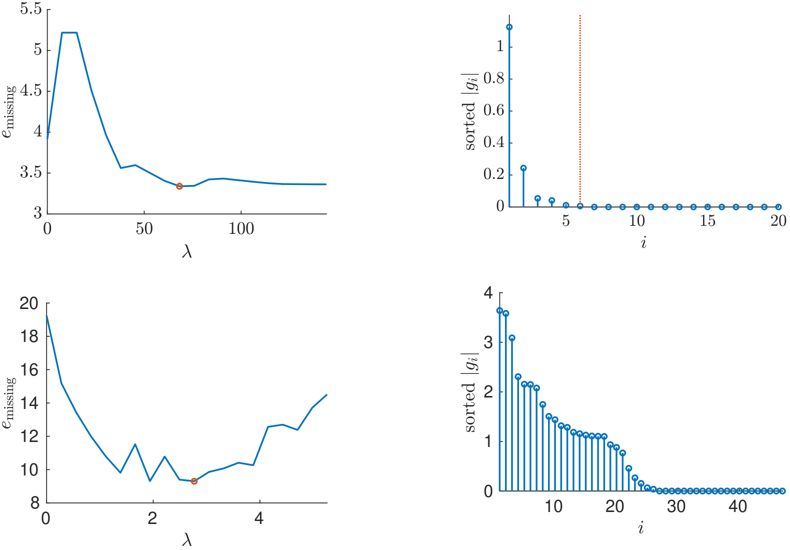
B. De Moor, et al.*DAISY: A database for identification of systems. Journal A, 38:4–5, 1997*

ℓ_1-norm regularization with optimized λ achieves the best performance

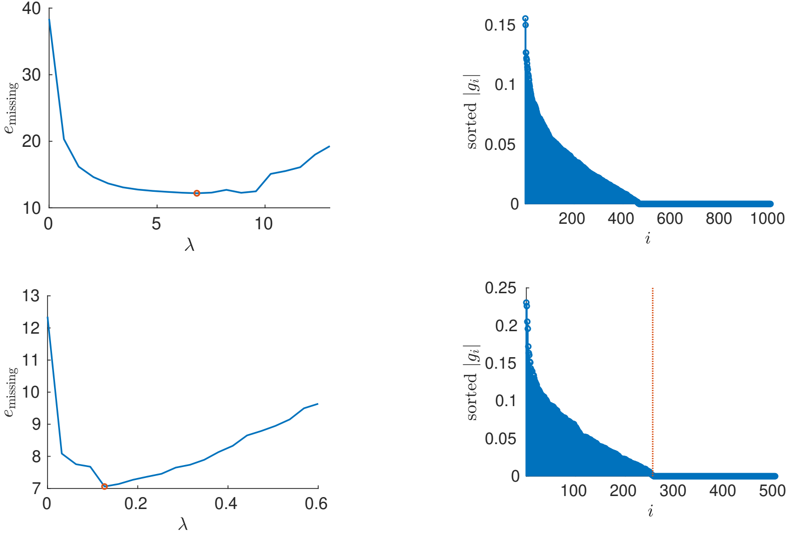
$$e_{\text{missing}} := \frac{\left\| w|_{I_{\text{missing}}} - \widehat{w}|_{I_{\text{missing}}} \right\|}{\left\| w|_{I_{\text{missing}}} \right\|} \, 100\%$$

	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

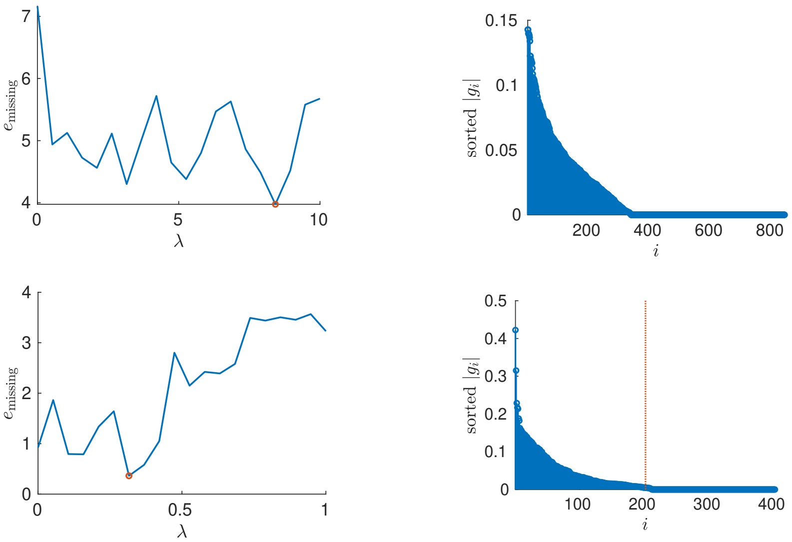
Tuning of λ and sparsity of g (datasets 1, 2)



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 $\rightsquigarrow$ systems theory

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

w_{d} inexact $\rightsquigarrow$ nonconvex optimization

- subspace methods
- local optimization
- convex relaxations

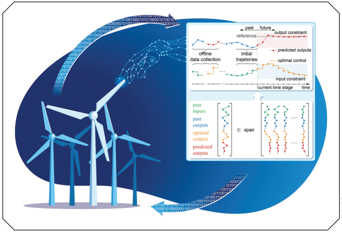
empirical validation

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

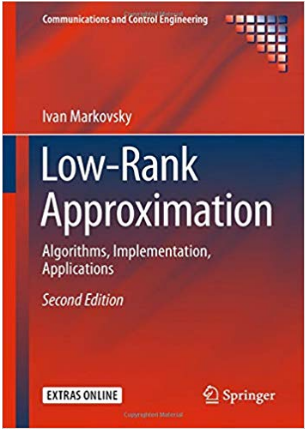
References

Data-Driven Control Based on the Behavioral Approach

FROM THEORY TO APPLICATIONS IN POWER SYSTEMS



IVAN MARKOVSKY, LINBIN HUANG, and FLORIAN DÖRFLER



A textbook problem

D. G. Luenberger, Introduction to Dynamical Systems: Theory, Models and Applications. John Wiley, 1979.

"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

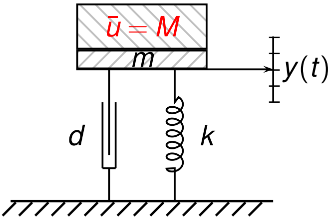
environmental temperature $\bar{u}$	$\xrightarrow{\text{heat transfer}}$	thermometer's reading y

measurement process: Newton's law of cooling

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

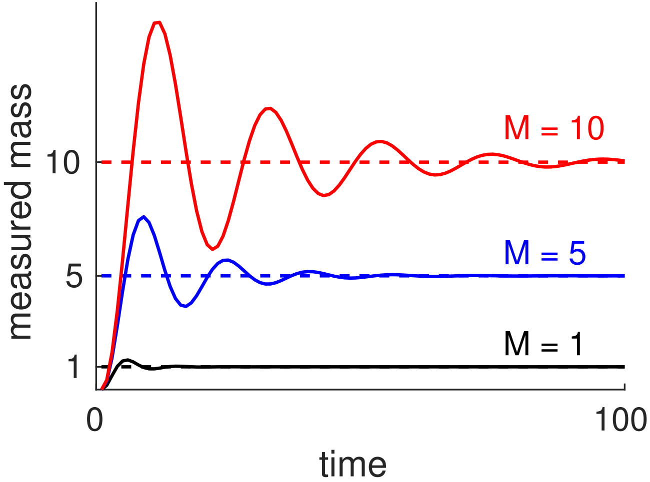
heat transfer coefficient $a > 0$

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

$$\begin{array}{ccc} \text{to-be-measured} & \xrightarrow{\text{measurement process}} & \text{measured} \\ \text{variable } u & & \text{variable } y \end{array}$$

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

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

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

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

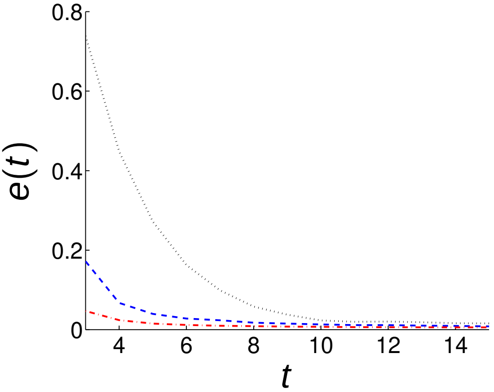
Empirical validation

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)

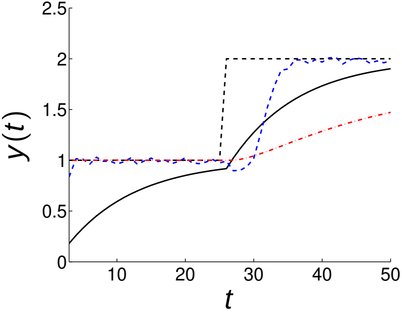
Simulated data of dynamic cooling process

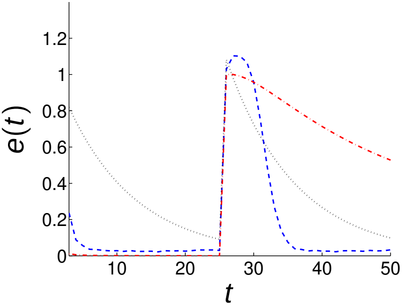


$e(t) \to 0$ as $t \to \infty$ at different rates

best is the Kalman filter (maximum likelihood estimator)

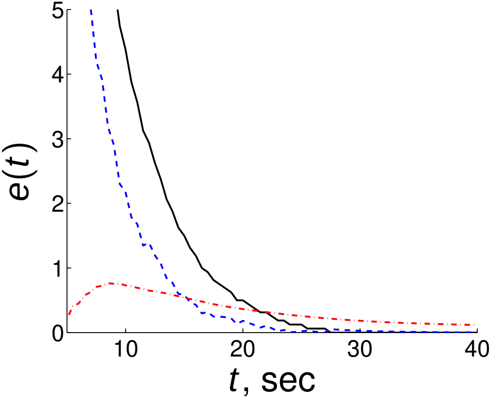
Simulation with time-varying parameter





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

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

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

trivial generalization to

- multivariable systems
- multiple data trajectories $\{ w_{\rm d}^1, \ldots, w_{\rm d}^N \}$
- 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 $\mathscr{H}_L(w_\text{d})$

- 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 $\mathscr{H}_L(\widehat{w_\text{d}})$

- hyper-parameters: L and the regularization parameter

Matlab implementation

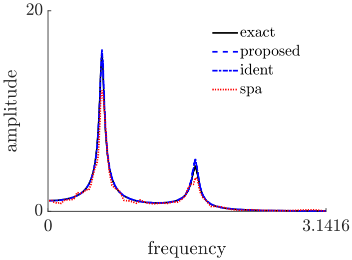
```
function Hh = dd_frest(ud, yd, z, n)
L = n + 1; t = (1:L)';
m = size(ud, 2); p = size(yd, 2);

%% 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
```

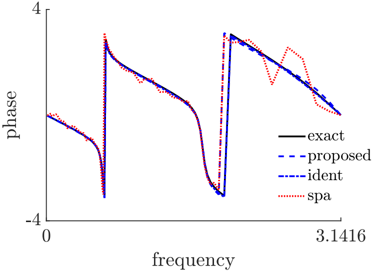
- effectively 5 lines of code
- MIMO case, multiple evaluation points
- $L = n+1$ in order to have a single hyper-parameter

Example: EIV setup with 4th order system

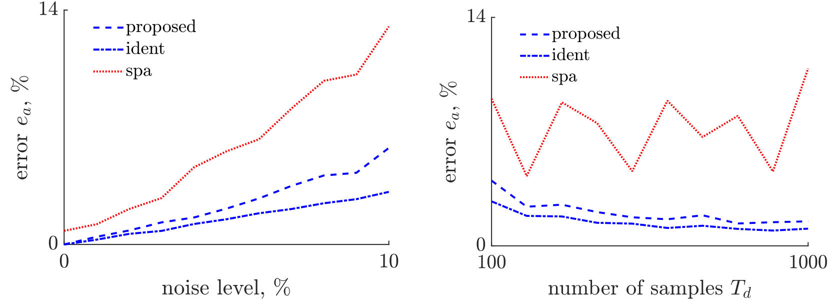
`dd_frest` is compared with

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





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



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

Kernel representation

LTI systems

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

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

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

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

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

FIR Volterra

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

bilinear

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

generalized bilinear

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

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 $w_{\text{d}} = (u_{\text{d}}, y_{\text{d}})$ of lag-ℓ FIR Volterra system $\mathscr{B}$

ϕ_{nl} — system's model structure

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

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

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

proof

$$\begin{bmatrix} \mathscr{H}_\ell(w_\mathrm{d}) \\ \mathscr{H}_L(\sigma^\ell u_\mathrm{d}) \\ \hline \mathscr{H}_\ell\big(\phi_\mathrm{nl}(x_{u_\mathrm{d}})\big) \\ \mathscr{H}_L\big(\sigma^\ell \phi_\mathrm{nl}(x_{u_\mathrm{d}})\big) \\ \hline \mathscr{H}_L(\sigma^\ell y_\mathrm{d}) \end{bmatrix} g = \begin{bmatrix} w_\mathrm{ini} \\ u \\ \hline \phi_\mathrm{nl}(x_{u_\mathrm{ini}}) \\ \phi_\mathrm{nl}(x_u) \\ \hline y \end{bmatrix} \begin{matrix} \left.\vphantom{\begin{matrix} a \\ a \end{matrix}}\right\} \mathrm{B1} \\ \left.\vphantom{\begin{matrix} a \\ a \end{matrix}}\right\} \mathrm{B2} \\ \}\mathrm{B3} \end{matrix}$$

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