Low-Rank Approximation: Theory, Algorithms, and Applications

Ivan Markovsky

ICREA

CIMNE
INTERNATIONAL CENTRE
FOR NUMERICAL METHODS
IN ENGINEERING

Outline

Outline

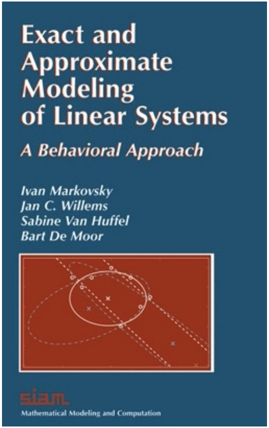
About me

Low-rank approximation in systems and control

Dynamic low-rank approximation

My background is mathematical engineering

2000–2005	PhD @ KUL
2007–2012	University of Southampton
2012–2022	Free Univ. Brussel
2022–	CIMNE, Barcelona

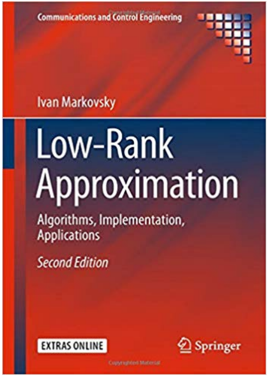


My core research interest is low-rank approx.

control theory

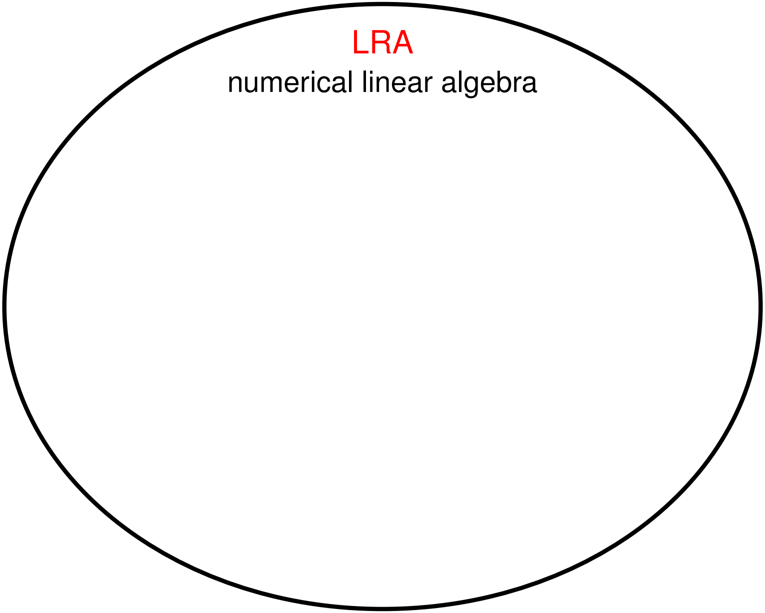
system identification

data-driven signal processing

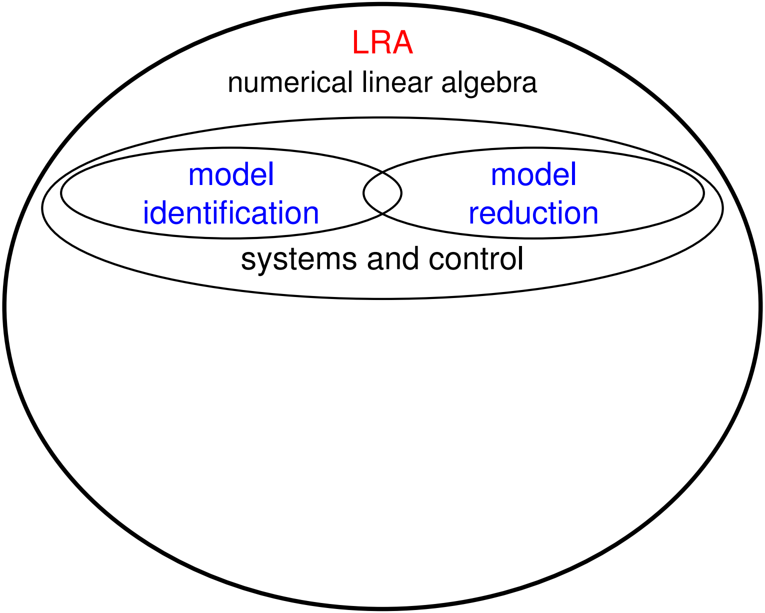


Outline

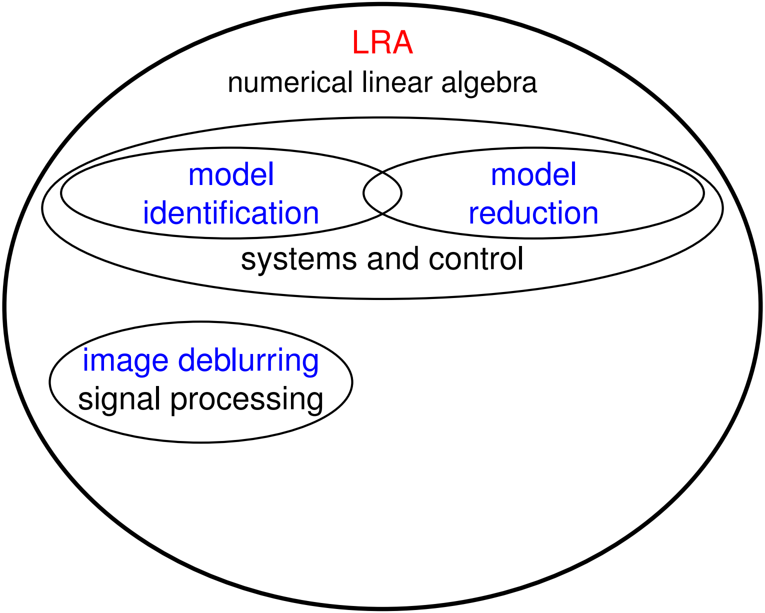
Application area of low-rank approximation



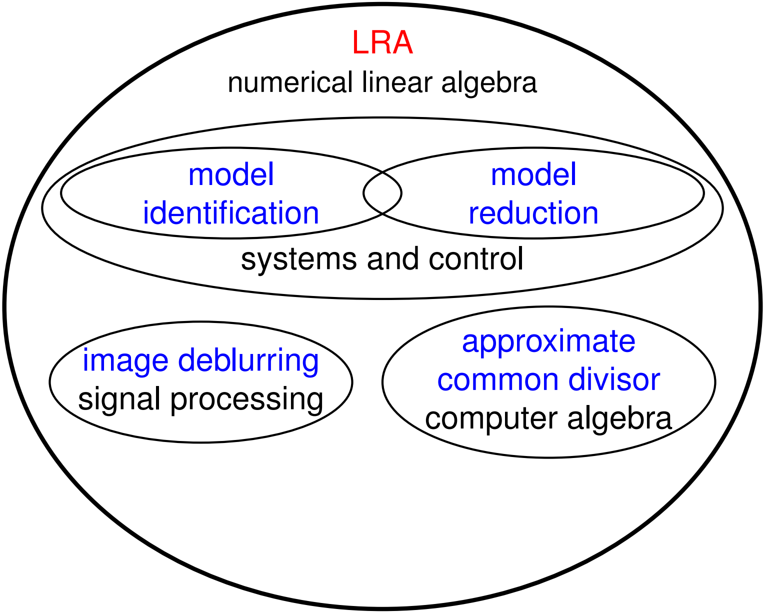
Application area of low-rank approximation



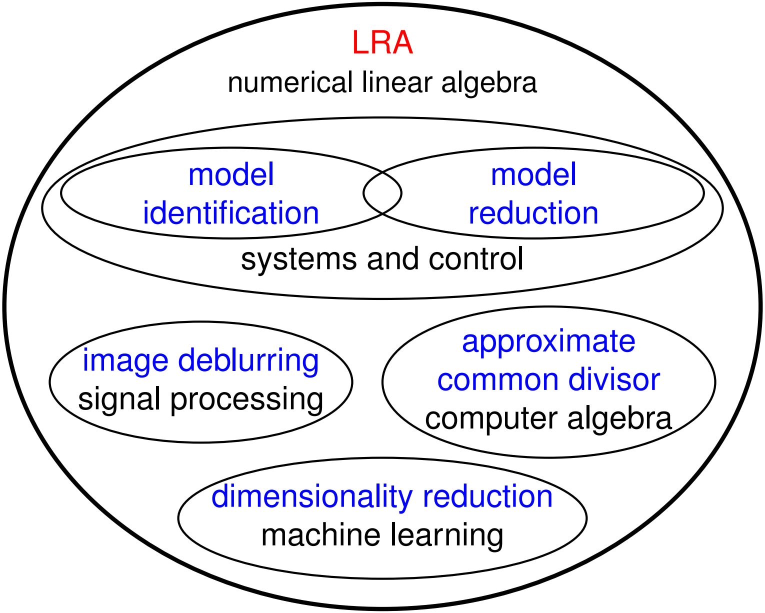
Application area of low-rank approximation



Application area of low-rank approximation



Application area of low-rank approximation



Different applications lead to additional constraints, besides the rank constraint

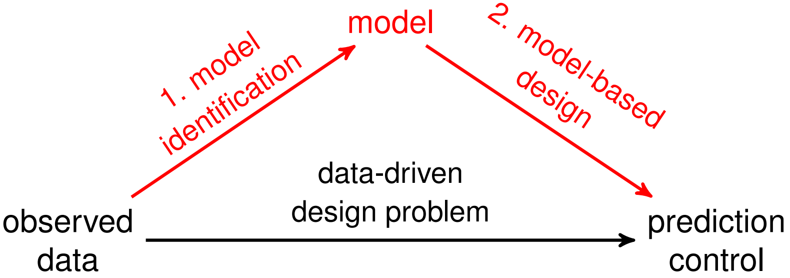
non-negativity of data and approximation

Sylvester structure $\leftrightarrow$ approximate GCD

Hankel structure $\leftrightarrow$ LTI dynamical systems

More recently, my interest is direct data-driven filtering and control

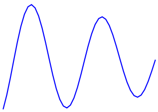
objective: bypass model identification



approach: structured low-rank matrix approximation and completion

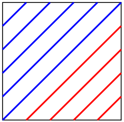
Academic example: time-series forecasting $\rightsquigarrow$ Hankel low-rank matrix completion

given: $\underbrace{y(1),\ldots,y(t)}_{\text{"past" data}}$ find: $\underbrace{y(t+1),\ldots,y(2t)}_{\text{"future" samples}}$



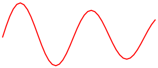
given data

$\mapsto$

rank  $\leq r$

matrix completion problem

$\mapsto$



prediction

Why Hankel structured low-rank matrix?

y is sum of n damped-exponentials

$$\Updownarrow$$

$$\operatorname{rank} \begin{bmatrix} y(1) & y(2) & \cdots & y(T-n) \\ y(2) & y(3) & \cdots & y(T-n+1) \\ \vdots & \vdots & & \vdots \\ y(n+1) & y(n+2) & \cdots & y(T) \end{bmatrix} \leq n$$

Hankel structured matrix $\mathscr{H}_{n+1}(y)$

Sum-of-damped-exponentials signals are solutions of linear constant coefficient ODE

$$y = \alpha_1 \exp_{z_1} + \cdots + \alpha_n \exp_{z_n} \qquad \exp_z(t) := z^t$$

$$\Updownarrow$$

$$p_0 y + p_1 \sigma y + \cdots + p_n \sigma^n y = 0 \quad (\sigma y)(t) := y(t+1)$$

$$\Updownarrow$$

$$y = Cx,\ \sigma x = Ax \qquad x(t) \in \mathbb{R}^n \text{ — state}$$

The solution set of linear constant coefficient ODE is linear time-invariant (LTI) system

n-th order autonomous LTI system

$$\mathscr{B} := \{\, y = Cx \mid \sigma x = Ax,\ x(0) \in \mathbb{R}^n \,\}$$

$\dim \mathscr{B} = n$ — complexity of $\mathscr{B}$

$\mathscr{L}_n$ — LTI systems with order $\leq n$

Model identification is equivalent to Hankel structured low-rank approximation

$$\begin{array}{ll} \text{minimize} & \text{over } \widehat{y} \text{ and } \widehat{\mathscr{B}} \quad \|y - \widehat{y}\| \\ \text{subject to} & \widehat{y} \in \widehat{\mathscr{B}} \in \mathscr{L}_n \end{array}$$

$$\Updownarrow$$

$$\begin{array}{ll} \text{minimize} & \text{over } \widehat{y} \quad \|y - \widehat{y}\| \\ \text{subject to} & \operatorname{rank} \mathscr{H}_{n+1}(\widehat{y}) \leq n \end{array}$$

Three main solution approaches

local optimization

convex relaxations

subspace methods

Outline

About me

Low-rank approximation in systems and control

Dynamic low-rank approximation

Dynamic low-rank approximation

problem formulation

- given $A(t) \in \mathbb{R}^{m \times n}$, for $t \in \mathbb{R}_+$,
- find $\min_X \|A(t) - X(t)\|$, for *all* $t \in \mathbb{R}_+$

comment

- without extra knowledge the problem decouples
- extra knowledge: model defining the evolution of A

example

- $\frac{\mathrm{d}}{\mathrm{d}\,t} A(t) = F\big(A(t)\big)$, $A(0) = A_{\text{ini}}$
- with given $F : \mathbb{R}^{m \times n} \to \mathbb{R}^{m \times n}$

Ideas for collaboration

discrete-time DLRA

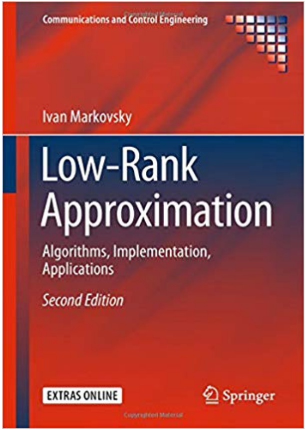
more general model (open system)

$$\frac{\mathrm{d}}{\mathrm{d}\,t} S(t) = F\big(S(t)\big) + G\big(U(t)\big), \quad S(0) = S_{\text{ini}}$$
$$A(t) = C\big(S(t)\big) + D\big(U(t)\big)$$

- U — input or disturbance
- S — state (unobserved)

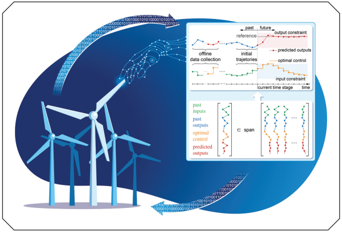
unknown model (F, G, C, D not given)

References



Data-Driven Control Based on the Behavioral Approach

FROM THEORY TO APPLICATIONS IN POWER SYSTEMS



IVAN MARKOVSKY, LINBIN HUANG, and FLORIAN DÖRFLER

Outline

Behavioral approach

Trajectory interpolation and approximation

Generalization for nonlinear systems

Pedagogical Example: Free fall prediction

Case study 1: Dynamic measurement

Case study 2: Direct data-driven fault detection

Case study 3: Frequency response estimation

Outline

We view systems as sets of signals

$w \in (\mathbb{R}^q)^{\mathbb{N}}$ — q-variate discrete-time signal

$\mathscr{B} \subset (\mathbb{R}^q)^{\mathbb{N}}$ — q-variate dynamical model

- linear — $\mathscr{B}$ is a linear subspace of $(\mathbb{R}^q)^{\mathbb{N}}$
- time-invariant — invariant under shifts: $(\sigma w)(t) := w(t+1)$

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

In practice, we deal with finite signals

restriction of w / $\mathscr{B}$ to finite horizon $[1, T]$

$$w|_T := \big(w(1), \dots, w(T)\big), \quad \mathscr{B}|_T := \{\, w|_T \mid w \in \mathscr{B} \,\}$$

for $w_d = \big(w_d(1), \dots, w_d(T_d)\big)$ and $1 \leq T \leq T_d$

$$\mathscr{H}_T(w_d) := \begin{bmatrix} (\sigma^0 w_d)|_T & (\sigma^1 w_d)|_T & \cdots & (\sigma^{T_d - T} w_d)|_T \end{bmatrix}$$

$w_d \in \mathscr{B}|_{T_d}$ — “exact data”

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

m — number of inputs

n — order ($=$ minimal state dimension)

ℓ — lag ($=$ observability index)

$\mathscr{L}_{(m,\ell,n)}$ — bounded complexity LTI systems

Nonparametric representation of LTI system's finite-horizon behavior

assumptions:

- $w_\mathrm{d} \in \mathscr{B}|_{T_\mathrm{d}}$ — exact offline data
- $\mathscr{B} \in \mathscr{L}_{(m,\ell,n)}$ — bounded complexity LTI system
- informative data, for $T \geq \boldsymbol{\ell}(\mathscr{B})$

$$\operatorname{rank} \mathscr{H}_T(w_\mathrm{d}) = mT + n \tag{GPE}$$

then, the data-driven representation holds

$$\operatorname{image} \mathscr{H}_T(w_\mathrm{d}) = \mathscr{B}|_T \tag{DDR}$$

Outline

Trajectory interpolation and approximation

I. Markovsky and F. Dörfler. Data-driven dynamic interpolation and approximation. Automatica, 135:110008, 2022.

Generic data-driven problem: trajectory interpolation/approximation

given:

	"data trajectory"	$w_{\text{d}} \in \mathscr{B}\vert_{T_{\text{d}}}$
given:	and elements	$w\vert_{I_{\text{given}}}$
	of a trajectory	$w \in \mathscr{B}\vert_{T}$

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

Generic data-driven problem: trajectory interpolation/approximation

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

($w|_{I_{\text{given}}}$ 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 $\widehat{w} \in \mathscr{B}|_T$

Generic data-driven problem: trajectory interpolation/approximation

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

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

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

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

Special cases

simulation

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

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

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 $w_{\mathrm{d}}^1, \dots, w_{\mathrm{d}}^N$

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

Generalizations

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

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

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

Generalizations

multiple data trajectories $w_{\mathrm{d}}^1, \dots, w_{\mathrm{d}}^N$

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

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, ...

Outline

Generalization for nonlinear systems

I. Markovsky. Data-driven simulation of generalized bilinear systems via linear time-invariant embedding. IEEE Trans. Automat. Contr., 68:1101–1106, 2023.

I. Markovsky and K. Usevich. Nonlinearly structured low-rank approximation. In Low-Rank and Sparse Modeling for Visual Analysis, pages 1–22. Springer, 2014.

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

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

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 $w_\text{d} = (u_\text{d}, y_\text{d})$ 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

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

- 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

Outline

The goal is to predict free fall trajectory

object with mass m, falling in gravitational field

The goal is to predict free fall trajectory

object with mass m, falling in gravitational field

- w — position

The goal is to predict free fall trajectory

object with mass m, falling in gravitational field

- w — position
- $v := \dot{w}$ — velocity

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

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

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:

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

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

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

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 $w_d^1, \ldots, w_d^N$ + ini. cond. $\mapsto w$

Modeling from first principles yields affine time-invariant dynamical system

second law of Newton + the law of gravity

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

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

Modeling from first principles yields affine time-invariant dynamical system

second law of Newton + the law of gravity

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

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

Data-driven free fall prediction method

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

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

Data-driven free fall prediction method

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

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

algorithm for data-driven prediction:

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

Data-driven free fall prediction method

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

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

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

Summary: prediction of free fall trajectory

first principles modeling

- use Newton's 2nd law, law of gravity, and friction

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

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

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

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

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

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

Outline

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?”

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

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

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

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

Thermometer: first order dynamical system

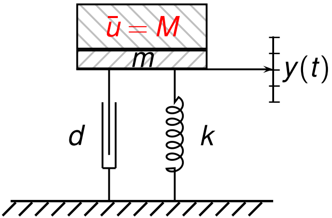
environmental temperature $\bar{u}$ $\xrightarrow{\text{heat transfer}}$ thermometer's reading y

measurement process: Newton's law of cooling

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

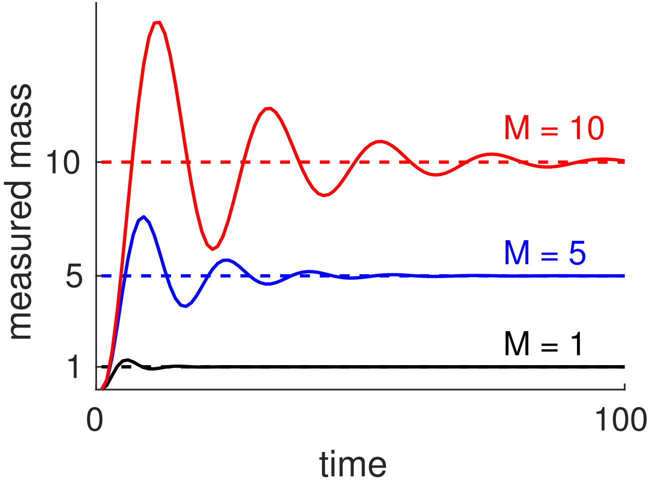
heat transfer coefficient $a > 0$

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

$$\text{to-be-measured variable } u \xrightarrow{\text{measurement process}} \text{measured variable } y$$

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

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

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

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

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

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

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

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

subspace method

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

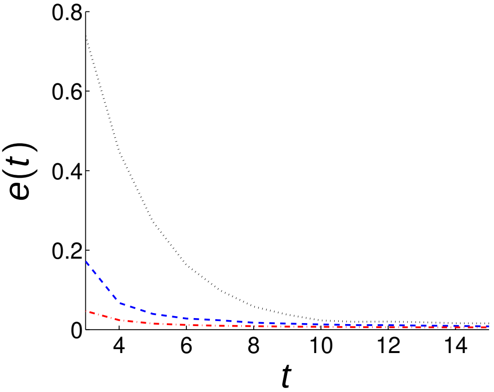
Empirical validation

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)

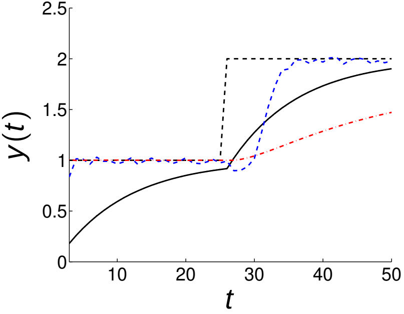
Simulated data of dynamic cooling process

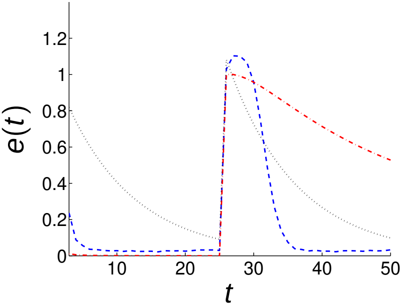


$e(t) \to 0$ as $t \to \infty$ at different rates

best is the Kalman filter (maximum likelihood estimator)

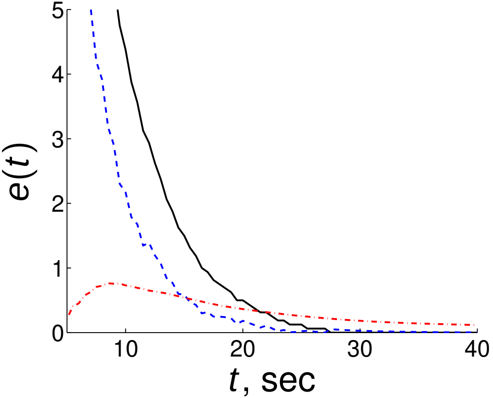
Simulation with time-varying parameter





Proof of concept prototype

Results in real-life experiment



Summary

dynamic measurement

steady-state value prediction

Summary

dynamic measurement

steady-state value prediction

the subspace method is applicable for

- high order dynamics
- noisy vector observations
- online computation

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

The problem considered is to detect abnormal operation based on observed data

prior information about data-generating system

model-based vs direct data-driven methods

observed data collected offline and online

- dedicated experiment — known excitation signal
- “normal” operation — unknown excitation signal

We consider three data collection scenarios

free response / transient data

forced response with known excitation

forced response with unknown excitation

Recall the nonparametric representation of an LTI system's finite-horizon behavior

assumptions:

- $w_\mathrm{d} \in \mathscr{B}|_{T_\mathrm{d}}$ — exact offline data
- $\mathscr{B} \in \mathscr{L}_{(m,\ell,n)}$ — bounded complexity LTI system
- for $T \geq \boldsymbol{\ell}(\mathscr{B})$, $\operatorname{rank} \mathscr{H}_T(w_\mathrm{d}) = mT + n$ — informative data

then, the data-driven representation holds

$$\operatorname{image} \mathscr{H}_T(w_\mathrm{d}) = \mathscr{B}|_T \qquad \text{(DDR)}$$

The fault detection criterion is the distance from online data w to system's behavior $\mathscr{B}$

$$\text{dist}(w, \mathscr{B}) := \min_{\widehat{w} \in \mathscr{B}|_T} \|w - \widehat{w}\|$$

under the assumptions, using (DDR), we have

$$\text{dist}(w, \mathscr{B}) = \|w - \mathscr{H}_T(w_\text{d})\mathscr{H}_T^+(w_\text{d})w\|$$

direct data-driven computation of the distance

The fault detection method has offline and online steps

offline: using w_{d}, find orthonormal basis B for $\mathscr{B}|_T$

online: compute and threshold

$$\mathrm{dist}(w, \mathscr{B}) = \left\|(I - BB^\top)w\right\|$$

with noisy data w_{d}, the offline step is

- SVD truncation of $\mathscr{H}_T(w_{\mathrm{d}})$
- structured low-rank approximation of $\mathscr{H}_T(w_{\mathrm{d}})$
- model identification, using w_{d}

With unobserved excitation signal e, prior knowledge about e is needed

zero-mean white Gaussian (disturbance)

deterministic signal $\rightsquigarrow$ input estimation problem

the model describes $w_{\text{ext}} := \left[\begin{smallmatrix} e \\ w \end{smallmatrix}\right]$

- e — unobserved signal
- w — observed signal

Finding e is a linear least-norm problem

given a model $\mathscr{B}_{\text{ext}}$ that describes $w_{\text{ext}} := \left[\begin{smallmatrix} e \\ w \end{smallmatrix}\right]$

$\widehat{e}_{\text{ln}} := \arg\min_{(\widehat{e},w)\in\mathscr{B}_{\text{ext}}|_T} \ \|\widehat{e}\|$

exact recovery $\widehat{e}_{\text{ln}} = e$ is not possible

Deterministic input estimation is linear least-squares problem

Π_e / Π_w — projection of $w_{\text{ext}} := \left[\begin{smallmatrix} e \\ w \end{smallmatrix}\right]$ on e / w

given, $\widehat{\mathscr{B}}_{\text{ext}}|_T = \text{image}\, B_{\text{ext}}$ (basis for $\widehat{\mathscr{B}}_{\text{ext}}|_T$)

$\widehat{e} := \Pi_e B_{\text{ext}} (\Pi_w B_{\text{ext}})^+ w$

Fault detection method with unobserved input

generalized distance measure:

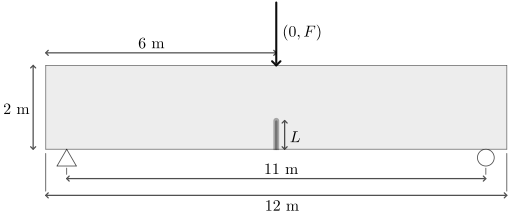
$$\operatorname{dist}(w, \mathscr{B}_{\mathrm{ext}}) := \min_{(\widehat{e},\widehat{w}) \in \mathscr{B}|_T} \|w - \widehat{w}\|$$

offline: using $(e_{\mathrm{d}}, w_{\mathrm{d}})$, find basis B_{ext} for $\mathscr{B}_{\mathrm{ext}}|_T$
and let $B_w := \Pi_w B_{\mathrm{ext}}$

online: compute and threshold

$$\operatorname{dist}(w, \mathscr{B}_{\mathrm{ext}}) = \left\|(I - B_w B_w^\top) w\right\|$$

Validation on vibrating beam with crack subject to unobserved disturbance force



data w_d^k	crack length	loss of stiffness	type of damage
0	0.0m	0%	none
1	0.7m	100%	severe
2	0.7m	36%	medium
3	0.2m	100%	medium
4	0.2m	36%	mild

observed displacements left / right of the crack

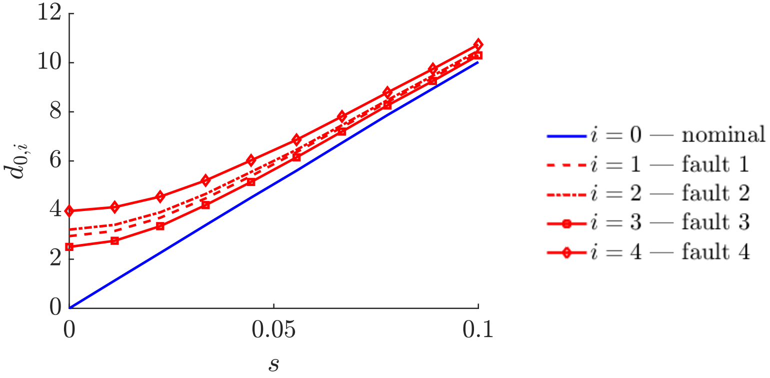
hyper-parameters: $T = 100$, $\ell = 2$, $n = 6$

offline computation: $\mathscr{B}^k$ using w_d^k

online computation: $d_{0,k} := \mathrm{dist}(w^0, \mathscr{B}^k)$

noise with standard deviation s added to w^0

Distances from nominal data to models as function of noise level



Comments

the beam behaves like 6th order LTI system

most severe crack is not hardest to detect

effect of the sensor location

Outlook

assumptions:

- bounded complexity LTI system
- hyper-parameters: horizon T and lag ℓ
- different ways to deal with noise in offline data w_d

advantages:

- representation invariant distance measure
- can deal with unobserved disturbance signal
- cheap to compute online and simple to implement

other applications

Outline

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 $\mathscr{B}$

I. Markovsky and H. Ossareh. Finite-data nonparametric frequency response evaluation without leakage. Automatica, 159:111351, 2024.

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 $\widehat{H}$

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

without prior knowledge of ℓ

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

trivial generalization to

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

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

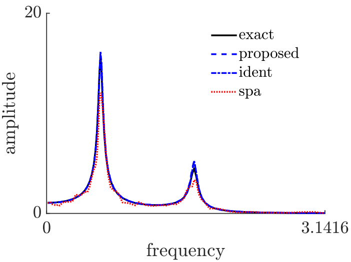
```
function Hh = dd_frest(ud, yd, z, n)
L = n + 1; t = (1:L)';
m = size(ud, 2); p = size(yd, 2);

%% 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
```

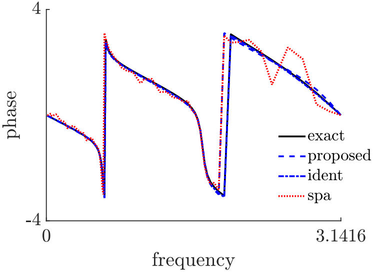
- effectively 5 lines of code
- MIMO case, multiple evaluation points
- $L = n + 1$ in order to have a single hyper-parameter

Example: EIV setup with 4th order system

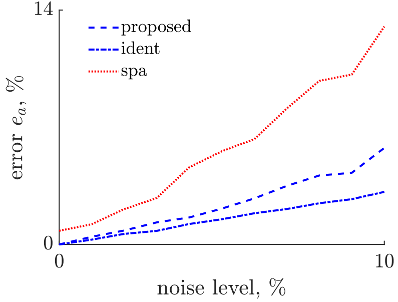
`dd_frest` is compared with

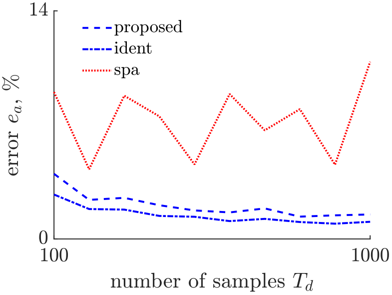
- `ident` — parametric maximum-likelihood estimator
- `spa` — nonparameteric estimator with Welch filter





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





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