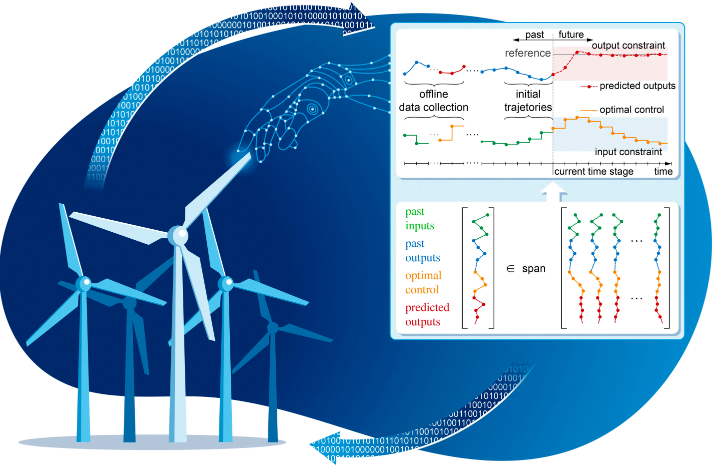
Data-Enabled Predictive Control of Autonomous Energy Systems



Florian Dörfler

ETH Zürich

DTU Summer School 2023

Acknowledgements

Jeremy Coulson

Linbin Huang

John Lygeros

Ivan Markovsky

Further:

Ezzat Elokda,
Paul Beuchat,
Daniele Alpago,
Jianzhe (Trevor) Zhen,
Claudio de Persis,
Pietro Tesi,
Henk van Waarde,
Eduardo Prieto,
Saverio Bolognani,
Andrea Favato,
Paolo Carlet,
Andrea Martin,
Luca Furieri,
Giancarlo Ferrari-Trecate,
Keith Moffat,
...

& many master students

Big, deep, intelligent & so on

- ***unprecedented availability*** of computation, storage, & data
- ***theoretical advances*** in statistics, optimization, & machine learning
- …and ***big-data*** frenzy

→ increasing importance of ***data-centric methods*** in all of science / engineering

> Make up your own opinion, but machine learning works too well to be ignored.

From Pixels to Torques: Policy Learning with Deep Dynamical Models

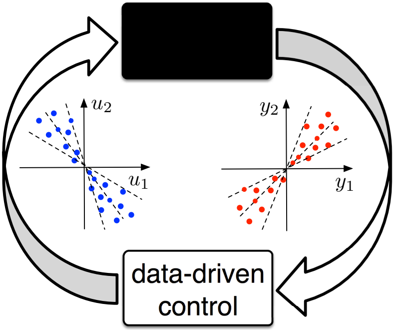
Niklas Wahlström NIKWA@ISY.LIU.SE
Division of Automatic Control, Linköping University, Linköping, Sweden

Thomas B. Schön THOMAS.SCHON@IT.UU.SE
Department of Information Technology, Uppsala University, Sweden

Marc Peter Deisenroth M.DEISENROTH@IMPERIAL.AC.UK
Department of Computing, Imperial College London, United Kingdom

Thoughts on data-driven control

- ***indirect data-driven control*** via models:
  data $\xrightarrow{\text{SysID}}$ model + uncertainty $\rightarrow$ control
- growing trend: ***direct data-driven control***
  by-passing models . . . (again) hyped, why ?



The direct approach is ***viable alternative***

- for some ***applications*** : model-based approach is too complex to be useful

  $\rightarrow$ too complex models, environments, sensing modalities, specifications (e.g., wind farm)

- due to (well-known) ***shortcomings of ID***

  $\rightarrow$ too cumbersome, models not identified for control, incompatible uncertainty estimates, ...

- when ***brute force*** data/compute available

Central promise: *It is often easier to learn a control policy from data rather than a model.*

Example 1973: autotuned PID

Abstraction reveals pros & cons

indirect (model-based) ***data-driven control***

minimize	control cost (u, x)	} outer optimization
subject to	(u, x) satisfy state-space model	
where	x estimated from (u, y) & model	} middle opt.
where	model identified from (u^d, y^d) data	} inner opt.

outer optimization + middle opt. } separation & certainty equivalence (→ LQG case)

inner opt. } **no** separation (→ ID-4-control)

→ nested multi-level optimization problem

direct (black-box) ***data-driven control***

minimize	control cost (u, y)
subject to	(u, y) consistent with (u^d, y^d) data

→ ***trade-offs***

modular vs. end-2-end

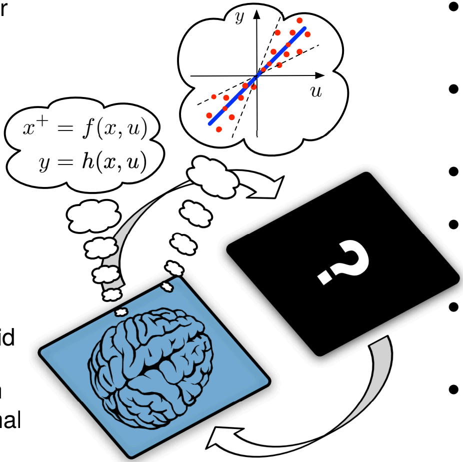
suboptimal (?) vs. optimal

convex vs. non-convex (?)

Additionally: account for ***uncertainty*** (hard to propagate in indirect approach)

Indirect (models) vs. direct (data)

- models are useful for design & beyond
- modular → easy to debug & interpret
- id = noise filtering
- id = projection on model class
- harder to propagate uncertainty through id
- no robust separation principle → suboptimal
- …



- some models too complex to be useful
- end-to-end → suitable for non-experts
- design handles noise
- harder to inject side info but no bias error
- transparent: no unmodeled dynamics
- possibly optimal but often less tractable
- …

there are ***no universal conclusions*** & plenty of counterexamples

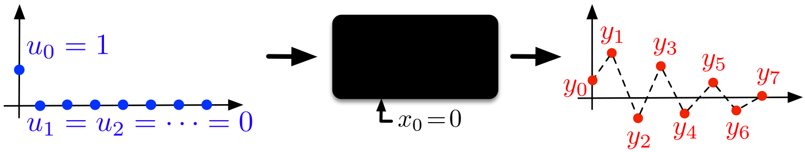
A direct approach: dictionary + MPC

① trajectory ***dictionary learning***

- motion primitives / basis functions
- theory: Koopman & Liouville
  practice: (E)DMD & particles

② ***MPC*** optimizing over dictionary span

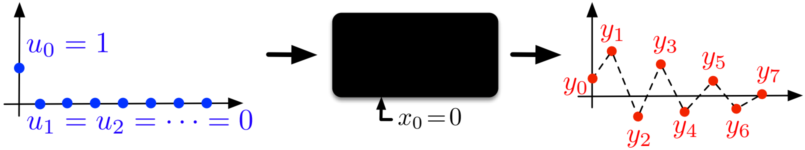
→ huge ***theory vs. practice*** gap
→ back to basics: ***impulse response***



- $\left.\begin{array}{l} u_0 = \delta_0 \\ x_0 = 0 \end{array}\right\} \rightsquigarrow y(t) = \text{impulse response} =: g(t) = \{g_0, g_1, g_2 \ldots\}$
- response to any other input $u(t)$ given by convolution with $g(t)$:

$$y(t) = g(t) * u(t) = \sum_{\tau=0}^{t-1} g(t-\tau)\, u(\tau) = g(0)\,u(t) + g(1)\,u(t-1) + \ldots$$

↑ convolution



Now what if we had the impulse response recorded in our data-library?

$$\begin{bmatrix} g_0 & g_1 & g_2 & \dots \end{bmatrix} = \begin{bmatrix} y_0^d & y_1^d & y_2^d & \dots \end{bmatrix}$$

⇝ can predict any future input $y_{\text{future}}(t)$ to any future input $u_{\text{future}}(t)$ by convolution with $g(t) = y^d(t)$: $y_{\text{future}}(t) = y_0^d u_{\text{future}}(t) + y_1^d u_{\text{future}}(t-1) + \dots$

in vector form

$\longrightarrow$ dynamic matrix control (Shell, 1970s): ***predictive control from raw data***

$$y_{\text{future}}(t) = \begin{bmatrix} y_0^d & y_1^d & y_2^d & \dots \end{bmatrix} \cdot \begin{bmatrix} u_{\text{future}}(t) \\ u_{\text{future}}(t-1) \\ u_{\text{future}}(t-2) \\ \vdots \end{bmatrix}$$

today: arbitrary, finite, & corrupted data, …stochastic & nonlinear ?

Today's menu

1. behavioral system theory: ***fundamental lemma***
2. ***DeePC***: data-enabled predictive control
3. robustification via salient ***regularizations***
4. cases studies from ***wind*** & ***power systems***

blooming literature (2-3 ArXiv / week)

$\rightarrow$ survey & tutorial to get started:

DATA-DRIVEN CONTROL BASED ON BEHAVIORAL APPROACH: FROM THEORY TO APPLICATIONS IN POWER SYSTEMS

Ivan Markovsky, Linbin Huang, and Florian Dörfler
I. Markovsky is with ICREA, Pg. Lluis Companys 23, Barcelona, and CIMNE, Gran Capitàn, Barcelona, Spain (e-mail: imarkovsky@cimne.upc.edu),
L. Huang and F. Dörfler are with the Automatic Control Laboratory, ETH Zürich, 8092 Zürich, Switzerland (e-mails: linhuang@ethz.ch, dorfler@ethz.ch).

Annual Reviews in Control 52 (2021) 42–64
Contents lists available at ScienceDirect
Annual Reviews in Control
journal homepage: www.elsevier.com/locate/arcontrol
ELSEVIER

Review article

Behavioral systems theory in data-driven analysis, signal processing, and control

Ivan Markovsky [a,*], Florian Dörfler [b]
[a] Department ELEC, Vrije Universiteit Brussel, Brussels, 1050, Belgium
[b] Automatic Control Laboratory (IfA), ETH Zürich, Zürich, 8092, Switzerland

[link] to related publications

Organization of the “school”

- I will ***teach the basics*** & provide pointers to more sophisticated research material ⟶ study cutting-edge papers yourself
- it’s a school: so we will spend time on the ***board*** ⟶ take notes

take notes in these boxes: hello world ☺

- We teach this material also in the ETH Zürich bachelor & have plenty of ***background material*** + implementation experience ⟶ please reach out if you need anything
- we will take a ***break*** after 90 minutes ⟶ coffee ☺

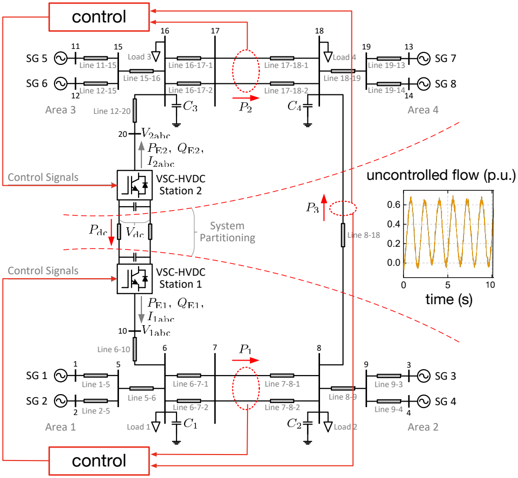
Preview

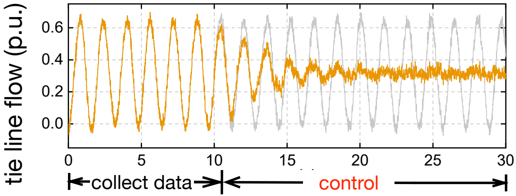
complex 4-area power ***system***:

large ($n = 208$), few sensors (8), nonlinear, noisy, stiff, input constraints, & decentralized control specifications

control objective: oscillation damping without a model

(grid has many owners, models are proprietary, operation in flux, . . .)





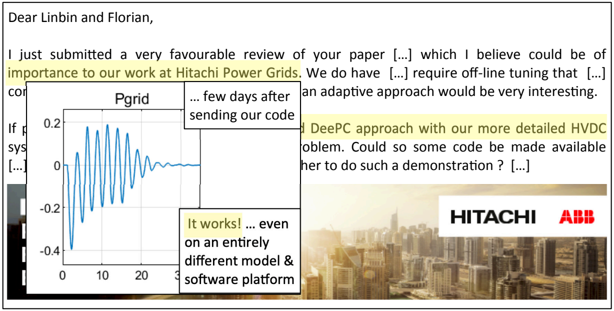
seek a method that ***works reliably***, can be ***efficiently*** implemented, & ***certifiable***

→ automating ourselves

Reality check: black magic or hoax ?

surely, nobody would put apply such a **shaky data-driven method**

- on the **world's most complex engineered system** (the electric grid),
- using the **world's biggest actuators** (Gigawatt-sized HVDC links),
- and subject to **real-time, safety, stability, constraints** . . . right?



at least someone believes that our method is practically useful . . .

LTI system representations

- ARX model (auto regressive, exogenous): $y(t+2) + 2y(t+1) + 3y(t) = 4u(t)$

- ARX to state space: $x(t) = \begin{bmatrix} y(t) \\ y(t+1) \end{bmatrix} \rightsquigarrow \begin{cases} \underbrace{x^+}_{x(t+1)} = \underbrace{\begin{bmatrix} 0 & 1 \\ -3 & -2 \end{bmatrix}}_{=A} x + \underbrace{\begin{bmatrix} 0 \\ 4 \end{bmatrix}}_{=B} u \\ y = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{=C} x \end{cases}$

- ARX to transfer function via z-transform

$y(t) \longrightarrow Y(z)$

$y(t+1) \longrightarrow z \cdot Y(z)$

$\rightsquigarrow Y(z) = \frac{4}{z^2 + 2z + 3} U(z)$

These are all parametric representations of an LTI system.
They are all so-called kernel representations since they nullify signals. E.g., for the ARX model, let z denote the forward time shift, e.g., $z \cdot y(t) = y(t+1)$, then we can write the model as

$$\underbrace{\begin{bmatrix} z^2 + 2z + 3, & 4 \end{bmatrix}}_{\text{"kernel"}} \cdot \underbrace{\begin{bmatrix} y(t) \\ u(t) \end{bmatrix}}_{\text{signal}} = 0$$

Behavioral view on dynamical systems

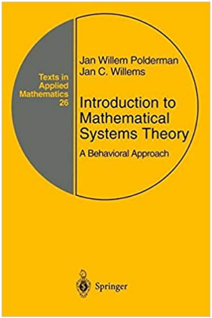
Definition: A discrete-time ***dynamical system*** is a 3-tuple $(\mathbb{Z}_{\geq 0}, \mathbb{W}, \mathscr{B})$ where

(i) $\mathbb{Z}_{\geq 0}$ is the *discrete-time* axis,

(ii) $\mathbb{W}$ is the *signal space*, &

(iii) $\mathscr{B} \subseteq \mathbb{W}^{\mathbb{Z}_{\geq 0}}$ is the *behavior*.

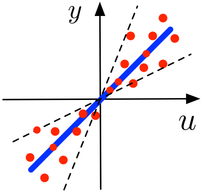
} $\mathscr{B}$ is the set of all trajectories



Definition: The dynamical system $(\mathbb{Z}_{\geq 0}, \mathbb{W}, \mathscr{B})$ is

(i) ***linear*** if $\mathbb{W}$ is a vector space & $\mathscr{B}$ is a subspace of $\mathbb{W}^{\mathbb{Z}_{\geq 0}}$

(ii) & ***time-invariant*** if $\mathscr{B} \subseteq \sigma\mathscr{B}$, where $\sigma w_t = w_{t+1}$.



LTI system $=$ shift-invariant subspace of trajectory space

$\longrightarrow$ abstract perspective suited for ***data-driven control***

Properties of the LTI trajectory space

- solution to $x^+ = A\underbrace{x}_{\in \mathbb{R}^n} + B\underbrace{u}_{\in \mathbb{R}^m}$ with $x(0) = x_{ini}$ is

  $x(0) = x_{ini}$, $x(1) = A x_{ini} + B u(0)$, $x(2) = A^2 x_{ini} + A B u(0) + B u(1)$, ...

  $\rightsquigarrow x(t) = A^t x_{ini} + \sum_{\tau=0}^{t-1} A^\tau B u(t-\tau-1)$

- solution to $x^+ = Ax + B$, $\underbrace{y}_{\in \mathbb{R}^p} = Cx + Du$, $x(0) = x_{ini}$ is

$$y(t) = C A^t x_{ini} + \underbrace{\sum_{\tau=0}^{t-1} C A^\tau B u(t-\tau-1) + D u(t)}_{g(t) * u(t)}$$

- in vector notation for $t \in \{0, 1, \dots, T\}$

$$\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ \vdots \\ y(T) \end{bmatrix} = \underbrace{\begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^T \end{bmatrix}}_{\mathcal{O}_T} x_{ini} + \underbrace{\begin{bmatrix} D & & & \\ CB & D & & \\ CAB & CB & D & \\ \ddots & \ddots & \ddots & \\ CA^{T-1}B \cdots & CAB & CB & D \end{bmatrix}}_{\mathcal{C}_T} \begin{bmatrix} u(0) \\ u(1) \\ u(2) \\ \vdots \\ u(T) \end{bmatrix}$$

$\mathcal{O}_T$: extended observability matrix $\quad$ $\mathcal{C}_T$: convolution (or impulse response) matrix

- compactly: $y = \mathcal{O}_T x_{ini} + \mathcal{C}_T u$ (*)
  $\mathcal{O}_T \in \mathbb{R}^{pT \times n}$

- <u>observability:</u> x_{ini} can be uniquely reconstructed from (*)
  $\Leftrightarrow \; n = \operatorname{rank} \mathcal{O}_T = \operatorname{rank} \begin{bmatrix} C \\ CA \\ \vdots \\ CA^T \end{bmatrix}$

  ⇝ the smallest T so that $\operatorname{rank} \mathcal{O}_T = n$ is called the <u>lag</u> ℓ of the system. In the SISO case ($p=1$), we have $\ell = n$; for $p \geq 1$, we have $\ell \leq n$ (MIMO case).

  ⇝ given past data $y_{ini} \in \mathbb{R}^{pT_{ini}}$, $u_{ini} \in \mathbb{R}^{mT_{ini}}$ of length $T_{ini} \geq \ell$, we can uniquely recover x_{ini} from
  $$y_{ini} = \mathcal{O}_{T_{ini}} x_{ini} + \mathcal{C}_{T_{ini}} x_{ini}$$

- **dimension** of the LTI trajectory space: for any x_{ini} what is the dimension $y = \begin{bmatrix} y(0) \\ \vdots \\ y(T) \end{bmatrix}$ and $u = \begin{bmatrix} u(0) \\ \vdots \\ u(T) \end{bmatrix}$?

$$\underbrace{\begin{bmatrix} u \\ y \end{bmatrix}}_{u \in \mathbb{R}^{mT},\; y \in \mathbb{R}^{pT}} = \begin{bmatrix} 0 & I \\ \mathcal{O}_T & \mathcal{C}_T \end{bmatrix} \begin{bmatrix} x_{ini} \\ u \end{bmatrix} \quad x_{ini} \in \mathbb{R}^n,\; u \in \mathbb{R}^{mT}$$

↑ this column has full rank mT (second column)

↑ this colum has full rank n provided $T \geq \ell$ (first column)

⇝ **dimension $\begin{bmatrix} u \\ y \end{bmatrix} = mT + n$ for $T \geq \ell$**

LTI systems & matrix time series

foundation of state-space subspace system ID & signal recovery algorithms

$\big(u(t), y(t)\big)$ satisfy recursive ***difference equation***

$$b_0 u_t + b_1 u_{t+1} + \ldots + b_n u_{t+n} + a_0 y_t + a_1 y_{t+1} + \ldots + a_n y_{t+n} = 0$$

(ARX / kernel representation)

$$\Rightarrow$$

$\Leftarrow$

under assumptions

$\begin{bmatrix} 0 & b_0 & a_0 & b_1 & a_1 & \ldots & b_n & a_n & 0 \end{bmatrix}$ in left nullspace of ***trajectory matrix*** (collected data)

$$\mathscr{H}\begin{pmatrix} u^d \\ y^d \end{pmatrix} = \begin{bmatrix} \begin{pmatrix} u^d_{1,1} \\ y^d_{1,1} \end{pmatrix} & \begin{pmatrix} u^d_{1,2} \\ y^d_{1,2} \end{pmatrix} & \begin{pmatrix} u^d_{1,3} \\ y^d_{1,3} \end{pmatrix} & \cdots \\ \begin{pmatrix} u^d_{2,1} \\ y^d_{2,1} \end{pmatrix} & \begin{pmatrix} u^d_{2,2} \\ y^d_{2,2} \end{pmatrix} & \begin{pmatrix} u^d_{2,3} \\ y^d_{2,3} \end{pmatrix} & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ \begin{pmatrix} u^d_{T,1} \\ y^d_{T,1} \end{pmatrix} & \begin{pmatrix} u^d_{T,2} \\ y^d_{T,2} \end{pmatrix} & \begin{pmatrix} u^d_{T,3} \\ y^d_{T,3} \end{pmatrix} & \cdots \end{bmatrix}$$

where $y^d_{t,i}$ is tth sample from ith experiment

Fundamental Lemma

Given: data $\begin{pmatrix} u_i^d \\ y_i^d \end{pmatrix} \in \mathbb{R}^{m+p}$ & LTI complexity parameters $\begin{cases} \text{lag } \ell \\ \text{order } n \end{cases}$

$$\text{set of all } T\text{-length trajectories } = \left\{ (u, y) \in \mathbb{R}^{(m+p)T} : \exists x \in \mathbb{R}^{nT} \text{ s.t. } x^+ = Ax + Bu\,,\; y = Cx + Du \right\} = \text{colspan} \begin{bmatrix} \begin{pmatrix} u_{1,1}^d \\ y_{1,1}^d \end{pmatrix} & \begin{pmatrix} u_{1,2}^d \\ y_{1,2}^d \end{pmatrix} & \begin{pmatrix} u_{1,3}^d \\ y_{1,3}^d \end{pmatrix} & \cdots \\ \begin{pmatrix} u_{2,1}^d \\ y_{2,1}^d \end{pmatrix} & \begin{pmatrix} u_{2,2}^d \\ y_{2,2}^d \end{pmatrix} & \begin{pmatrix} u_{2,3}^d \\ y_{2,3}^d \end{pmatrix} & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ \begin{pmatrix} u_{T,1}^d \\ y_{T,1}^d \end{pmatrix} & \begin{pmatrix} u_{T,2}^d \\ y_{T,2}^d \end{pmatrix} & \begin{pmatrix} u_{T,3}^d \\ y_{T,3}^d \end{pmatrix} & \cdots \end{bmatrix}$$

parametric state-space model

raw data (every column is an experiment)

if and only if the trajectory matrix has rank $m \cdot T + n$ for all $T > \ell$

$$\text{set of all } T\text{-length trajectories } = \left\{ (u, y) \in \mathbb{R}^{(m+p)T} : \exists x \in \mathbb{R}^{nT} \text{ s.t. } x^+ = Ax + Bu\,,\; y = Cx + Du \right\} = \operatorname{colspan} \begin{bmatrix} \begin{pmatrix} u^d_{1,1} \\ y^d_{1,1} \end{pmatrix} & \begin{pmatrix} u^d_{1,2} \\ y^d_{1,2} \end{pmatrix} & \begin{pmatrix} u^d_{1,3} \\ y^d_{1,3} \end{pmatrix} & \cdots \\ \begin{pmatrix} u^d_{2,1} \\ y^d_{2,1} \end{pmatrix} & \begin{pmatrix} u^d_{2,2} \\ y^d_{2,2} \end{pmatrix} & \begin{pmatrix} u^d_{2,3} \\ y^d_{2,3} \end{pmatrix} & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ \begin{pmatrix} u^d_{T,1} \\ y^d_{T,1} \end{pmatrix} & \begin{pmatrix} u^d_{T,2} \\ y^d_{T,2} \end{pmatrix} & \begin{pmatrix} u^d_{T,3} \\ y^d_{T,3} \end{pmatrix} & \cdots \end{bmatrix}$$

all trajectories constructible from finitely many previous trajectories

- ***standing on the shoulders of giants***: classic Willems' result was only "if" & required further assumptions: Hankel, persistency of excitation, controllability

A note on persistency of excitation

Jan C. Willems[a], Paolo Rapisarda[b], Ivan Markovsky[a,*], Bart L.M. De Moor[a]

[a]ESAT, SCD/SISTA, K.U. Leuven, Kasteelpark Arenberg 10, B 3001 Leuven, Heverlee, Belgium
[b]Department of Mathematics, University of Maastricht, 6200 MD Maastricht, The Netherlands

Received 3 June 2004; accepted 7 September 2004
Available online 30 November 2004

- terminology ***fundamental*** is justified : motion primitives, subspace SysID, dictionary learning, (E)DMD, . . . all implicitly rely on this equivalence

- many recent ***extensions*** to other ***system classes*** (bi-linear, descriptor, LPV, delay, Volterra series, Wiener-Hammerstein, . . .), other ***matrix data structures*** (mosaic Hankel, Page, . . .), & other ***proof methods***

Input design for Fundamental Lemma

Definition: The data signal $u^d \in \mathbb{R}^{mT_d}$ of length T_d is ***persistently exciting of order*** T if the Hankel matrix $\begin{bmatrix} u_1 & \cdots & u_{T_d-T+1} \\ \vdots & \ddots & \vdots \\ u_T & \cdots & u_{T_d} \end{bmatrix}$ is of full rank.

mT rows

rank $= mT$ only if $T_d - T + 1 \geq mT$ $\Rightarrow$ sufficiently rich & long data

Input design [Willems et al, '05]: Controllable LTI system & persistently exciting input u^d of order $T + n$ $\Longrightarrow$ rank $\left(\mathscr{H}\begin{pmatrix} u^d \\ y^d \end{pmatrix}\right) = mT + n$.

Data matrix structures & preprocessing

Matrix structures

$w = \begin{pmatrix} u^d \\ y^d \end{pmatrix}$

- **trajectory matrix:** $H(w) = \begin{bmatrix} \text{trajectory 1 of length } T & \vdots & \text{trajectory 2 of length } T & \vdots & \cdots \end{bmatrix}$

  requires independent experiments

- **page matrix:** $H(w) = \begin{bmatrix} w_1 & w_{T+1} & w_{2T+1} & \\ w_2 & \vdots & \vdots & \cdots \\ \vdots & & & \\ w_T & w_{2T} & w_{3T} & \end{bmatrix}$

  requires one long trajectory

- **Hankel matrix:** $H(w) = \begin{bmatrix} w_1 & w_2 & w_3 & \\ w_2 & w_3 & w_4 & \cdots \\ \vdots & \vdots & \vdots & \\ w_T & w_{T+1} & w_{T+2} & \end{bmatrix}$ uses shift invariance

  requires one short trajectory

$\rightsquigarrow$ Hankel structure (constant anti-diagonals) due to shift invariance

⇝ all these are valid matrices whose columns span the space of length-T trajectories provided that their rank is $mT+n$. One can also combine these different data structures into block matrices...

Preprocessing: if the data $w = \begin{pmatrix} u^d \\ y^d \end{pmatrix}$ is noisy, then likely all of the above matrices have full rank.

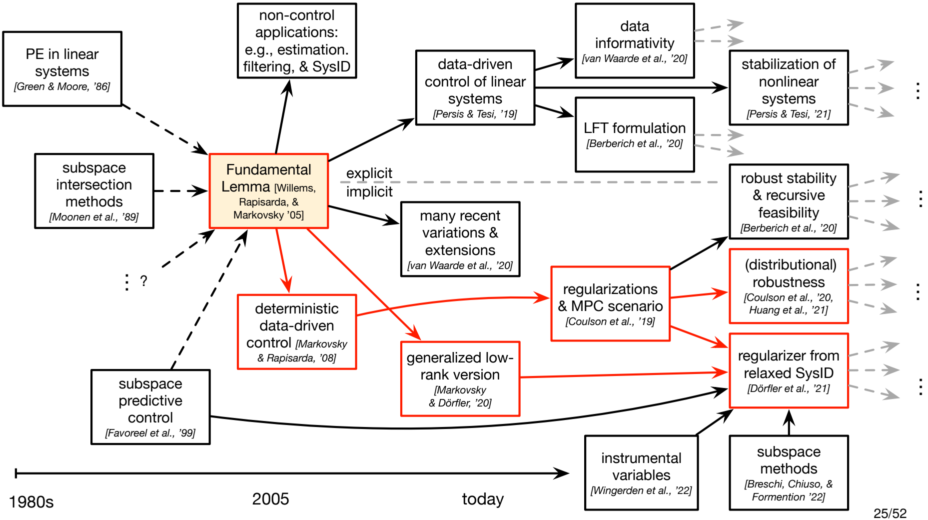
⇝ low-rank approximation: to denoise the data, find the "closest" matrix so that its rank is $mT+n$

for theory: look up the Eckhard Young theorem

⇝ practical solution is singular value thresholding: remove all but the $mT+n$ dominant singular values of the SVD

⇝ hard problem for Hankel matrices if it is desired to keep the Hankel structure (and induce shift invariance)

Bird’s view & today’s sample path through the accelerating literature



Output **M**odel **P**redictive **C**ontrol (**MPC**)

$$
\begin{aligned}
\underset{u,x,y}{\text{minimize}} \quad & \sum_{k=1}^{T_{\text{future}}} \|y_k - r_k\|_Q^2 + \|u_k\|_R^2 \\
\text{subject to} \quad & \left.\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k + Du_k \end{aligned}\right\} \quad \forall k \in \{1, \dots, T_{\text{future}}\} \\
& \left.\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k + Du_k \end{aligned}\right\} \quad \forall k \in \{-T_{\text{ini}} - 1, \dots, 0\} \\
& \left.\begin{aligned} u_k &\in \mathcal{U} \\ y_k &\in \mathcal{Y} \end{aligned}\right\} \quad \forall k \in \{1, \dots, T_{\text{future}}\}
\end{aligned}
$$

quadratic cost with $R \succ 0, Q \succeq 0$ & ref. r

model for **prediction** with $k \in [1, T_{\text{future}}]$

model for **estimation** with $k \in [-T_{\text{ini}} - 1, 0]$ & $T_{\text{ini}} \geq$ lag (many flavors)

hard operational or safety **constraints**

*"[MPC] has perhaps too little system theory and too much **brute force** [...], but MPC is an area where all aspects of the field [...] are in synergy." – Willems '07*

Elegance aside, for a LTI plant, deterministic, & with known model, MPC is the ***gold standard of control***.

Data-enabled Predictive Control (DeePC)

$$\underset{g,u,y}{\text{minimize}} \quad \sum_{k=1}^{T_{\text{future}}} \|y_k - r_k\|_Q^2 + \|u_k\|_R^2$$

$$\text{subject to} \quad \mathscr{H}\begin{pmatrix} u^{\mathsf{d}} \\ y^{\mathsf{d}} \end{pmatrix} \cdot g = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ \hline u \\ y \end{bmatrix}$$

$$\left.\begin{array}{l} u_k \in \mathcal{U} \\ y_k \in \mathcal{Y} \end{array}\right\} \quad \forall k \in \{1, \ldots, T_{\text{future}}\}$$

quadratic cost with $R \succ 0, Q \succeq 0$ & ref. r

non-parametric model for **prediction** and **estimation**

hard operational or safety **constraints**

- real-time measurements $(u_{\text{ini}}, y_{\text{ini}})$ for estimation — updated **online**
- trajectory matrix $\mathscr{H}\begin{pmatrix} u^{\mathsf{d}} \\ y^{\mathsf{d}} \end{pmatrix}$ from past experimental data — collected **offline** (could be adapted online)

→ ***equivalent to MPC*** in deterministic LTI case ...
but needs to be robustified in case of noise / nonlinearity !

Regularizations to make it work

$$\underset{g,u,y,\sigma}{\text{minimize}} \quad \sum_{k=1}^{T_{\text{future}}} \|y_k - r_k\|_Q^2 + \|u_k\|_R^2 + \lambda_y \|\sigma\|_p + \lambda_g h(g)$$

$$\text{subject to} \quad \mathscr{H}\begin{pmatrix} u^{\mathsf{d}} \\ y^{\mathsf{d}} \end{pmatrix} \cdot g = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ \hline u \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ \sigma \\ 0 \\ 0 \end{bmatrix}$$

$$\left.\begin{matrix} u_k \in \mathcal{U} \\ y_k \in \mathcal{Y} \end{matrix}\right\} \quad \forall k \in \{1, \ldots, T_{\text{future}}\}$$

measurement noise
→ infeasible y_{ini} estimate
→ estimation slack σ
→ moving-horizon least-square filter

noisy or nonlinear (offline) **data matrix**
→ any $\binom{u}{y}$ feasible
→ add regularizer $h(g)$

Bayesian intuition: regularization ⇔ prior, e.g., $h(g) = \|g\|_1$ sparsely selects {trajectory matrix columns} = {motion primitives} ∼ low-order basis

triangle inequality is tight since Δx can be collinear with $Ax-b$

Robustness intuition: regularization ⇔ robustifies, e.g., in a simple case

$$\min_x \max_{\Delta : \|\Delta\| \le \rho} \|(A+\Delta)x - b\| \;\underset{=}{\le}\; \min_x \max_{\|\Delta\| \le \rho} \|Ax-b\| + \|\Delta x\| = \min_x \|Ax-b\| + \rho\|x\|$$

Regularization = relaxation of bi-level ID

$$\begin{aligned}
&\text{minimize}_{u,y,g} && \text{control cost}(u,y)\\
&\text{subject to} && \begin{bmatrix} u \\ y \end{bmatrix} = \mathscr{H}\begin{pmatrix} \hat{u} \\ \hat{y} \end{pmatrix} g\\
&\text{where} && \begin{pmatrix} \hat{u} \\ \hat{y} \end{pmatrix} \in \operatorname{argmin} \left\| \begin{pmatrix} \hat{u} \\ \hat{y} \end{pmatrix} - \begin{pmatrix} u^{\mathrm{d}} \\ y^{\mathrm{d}} \end{pmatrix} \right\|\\
& && \text{subject to } \operatorname{rank}\left(\mathscr{H}\begin{pmatrix} \hat{u} \\ \hat{y} \end{pmatrix}\right) = mL + n
\end{aligned}$$

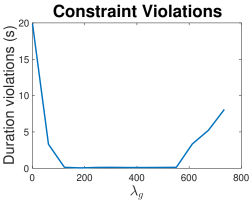
} optimal control

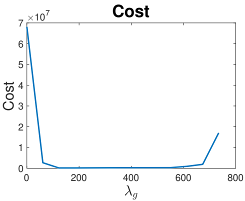
} system identification

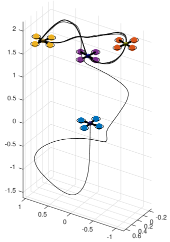
↓ sequence of convex relaxations ↓

$$\begin{aligned}
&\text{minimize}_{u,y,g} && \text{control cost}(u,y) + \lambda_g \cdot \|g\|_1\\
&\text{subject to} && \begin{bmatrix} u \\ y \end{bmatrix} = \mathscr{H}\begin{pmatrix} u^{d} \\ y^{d} \end{pmatrix} g
\end{aligned}$$

ℓ_1-regularization
$=$ relaxation of id
smooth order selection







Certainty-Equivalence Regularizer

ARX representation of predictor:

- recall $y = \mathcal{O}_T x_{ini} + \mathcal{C}_T u$

  where x_{ini} satisfies $y_{ini} = \mathcal{O}_{T_{ini}} x_{ini} + \mathcal{C}_{T_{ini}} u_{ini}$

$\rightsquigarrow y = K \cdot \begin{bmatrix} u_{ini} \\ y_{ini} \\ u \end{bmatrix}$, where K is

learned from data (certainty equiv.):

$$K = \underset{\tilde{K}}{\operatorname{argmin}} \left\| Y_f - \tilde{K} \begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix} \right\| = Y_f \begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix}^{+}$$

(note: K should also satisfy structural constraints, which we dropped)

$$\rightsquigarrow y = Y_f \cdot \begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix}^{+} \cdot \begin{bmatrix} u_{ini} \\ y_{ini} \\ u \end{bmatrix}$$

DeePC representation of predictor:

$$\begin{bmatrix} u_{ini} \\ y_{ini} \\ u \\ y \end{bmatrix} = \begin{bmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{bmatrix} g$$

— $H(w)$ suitably partitioned

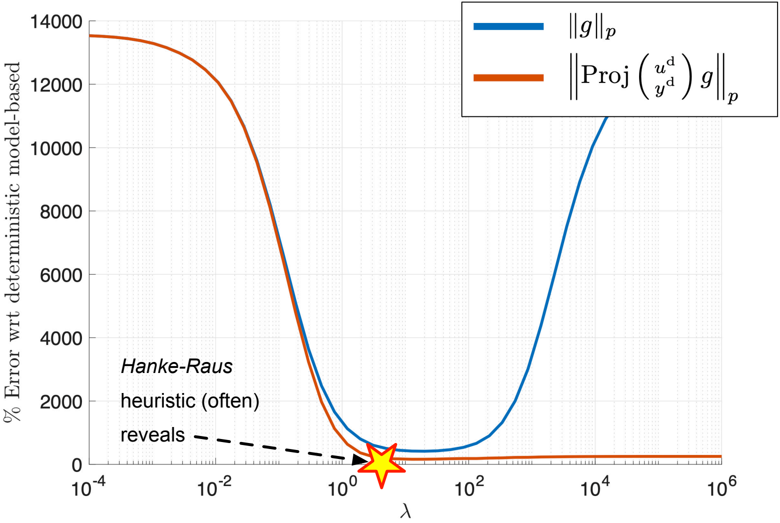
or $y = Y_f\, g$ with g from $\begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix} g = \begin{bmatrix} u_{ini} \\ y_{ini} \\ u \end{bmatrix}$

$$\rightsquigarrow y = \underbrace{Y_f \cdot \begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix}^{+} \begin{bmatrix} u_{ini} \\ y_{ini} \\ u \end{bmatrix}}_{\text{particular solution}} + \underbrace{Y_f \cdot \ker \begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix}}_{\text{homogeneous}}$$

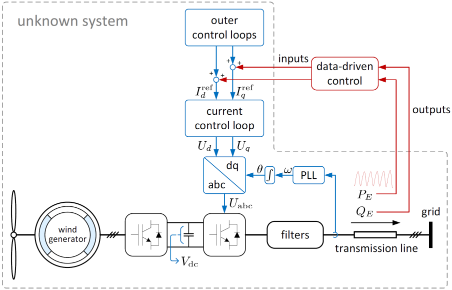
these two coincide if we force that $0 = \left\| \left(I - \begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix}^{+} \begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix} \right) g \right\|$

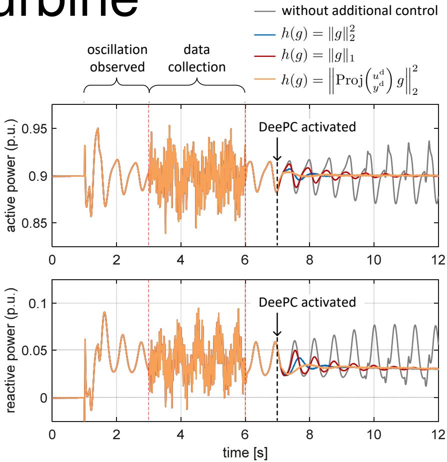
add this regularizer to DeePC → $\text{Proj}\begin{pmatrix} u^d \\ y^d \end{pmatrix}$ on nullspace

Performance of regularizers applied to a stochastic LTI system

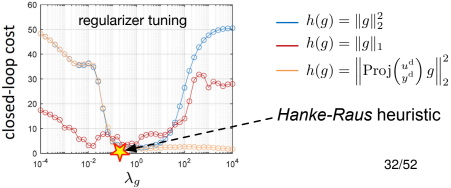


Case study: wind turbine

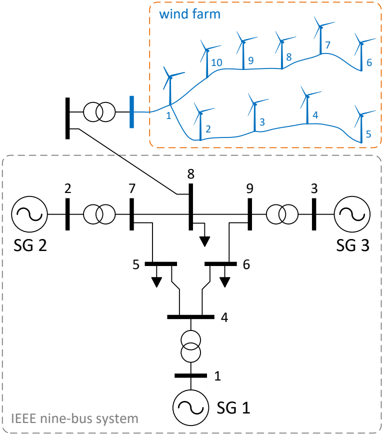


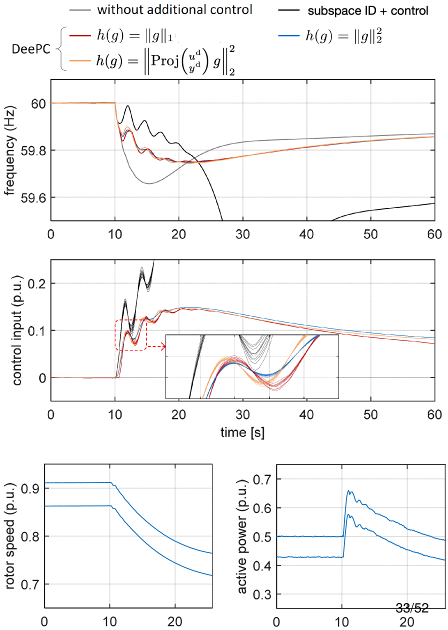


- detailed ***industrial model***: 37 states & highly nonlinear (abc $\leftrightarrow$ dq, MPTT, PLL, power specs, dynamics, etc.)
- turbine & grid model ***unknown*** to commissioning engineer & operator
- weak grid + PLL + fault $\rightarrow$ ***loss of sync***
- disturbance to be rejected by ***DeePC***



Case study ++ : wind farm

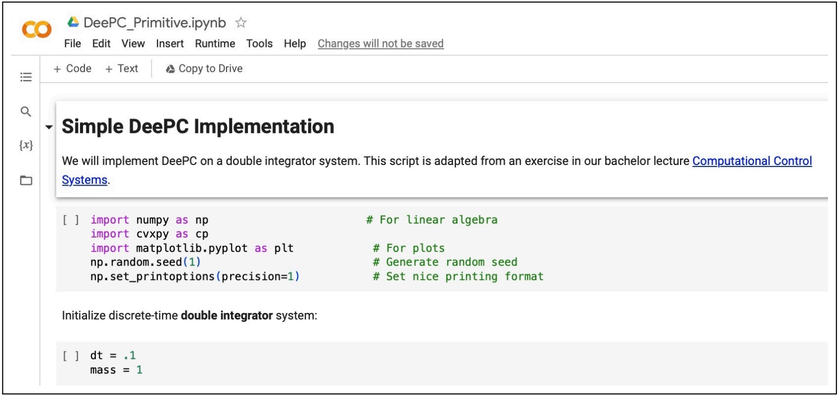




- ***high-fidelity models*** for turbines, machines, & IEEE-9-bus system
- ***fast frequency response*** via ***decentralized DeePC*** at turbines

DeePC is easy to implement → try it !

→ simple script adapted from our ETH Zürich bachelor course on *Computational control* : https://colab.research.google.com/drive/1URdRqr-Up0A6uDMjlU6gwmsoAAPl1GId?usp=sharing



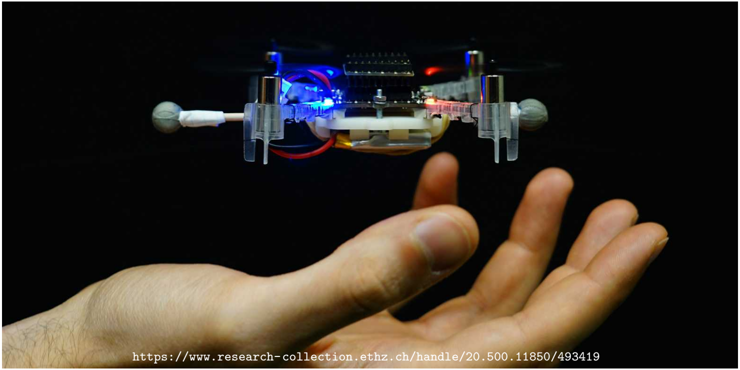
Towards a theory for nonlinear systems

idea : lift nonlinear system to large/∞-dimensional bi-/linear system

$\rightarrow$ Carleman, Volterra, Fliess, Koopman, Sturm-Liouville methods

$\rightarrow$ nonlinear dynamics can be approximated by LTI on finite horizon

regularization singles out relevant features / basis functions in data

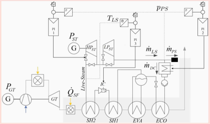


Works very well across case studies

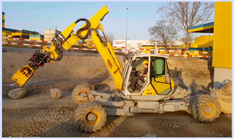
quad coptor fig-8 tracking

quadruped (by Fawcett, Afsari Amers, & Hamed)

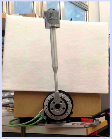
greenhouse automation (by Automatoes)



combined cycle power plant (by P Mahdavipour et. al)

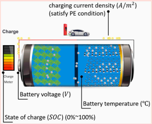


robotic excavator

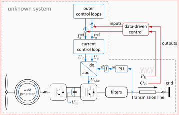


pendulum swing up

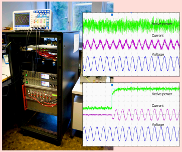
traffic coordination (by J. Wang et al.)



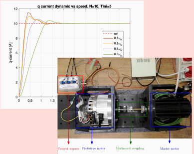
battery charging (by K. Chen et al.)



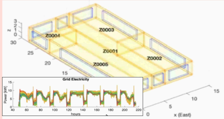
wind turbine control



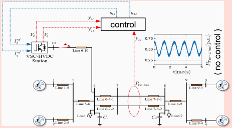
grid-connected converter



synchronous motor drive



energy hub & building automation



power system oscillation damping

Distributional robustification beyond LTI

- ***problem abstraction*** : $\min_{x \in \mathcal{X}} \; c(\widehat{\xi}, x) \;\; = \;\; \min_{x \in \mathcal{X}} \; \mathbb{E}_{\xi \sim \widehat{\mathbb{P}}}\left[c(\xi, x)\right]$

  where $\widehat{\xi}$ denotes *measured* data with *empirical distribution* $\widehat{\mathbb{P}} = \delta_{\widehat{\xi}}$

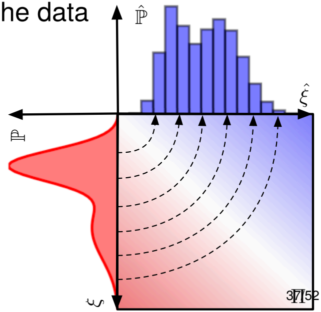
$\Rightarrow$ ***poor out-of-sample performance*** of above sample-average solution $x^\star$
for real problem: $\mathbb{E}_{\xi \sim \mathbb{P}}\left[c(\xi, x^\star)\right]$ where $\mathbb{P}$ is the *unknown distribution* of ξ

- ***distributionally robust*** formulation accounting for all (possibly nonlinear) stochastic processes that could have generated the data

$$\inf_{x \in \mathcal{X}} \; \sup_{\mathbb{Q} \in \mathbb{B}_\epsilon(\widehat{\mathbb{P}})} \; \mathbb{E}_{\xi \sim \mathbb{Q}}\left[c(\xi, x)\right]$$

where $\mathbb{B}_\epsilon(\widehat{\mathbb{P}})$ is an ***ϵ-Wasserstein ball*** centered at empirical sample distribution $\widehat{\mathbb{P}}$:

$$\mathbb{B}_\epsilon(\widehat{\mathbb{P}}) = \left\{ \mathbb{P} \,:\, \inf_{\Pi} \int \left\| \xi - \widehat{\xi} \right\|_p d\Pi \leq \epsilon \right\}$$

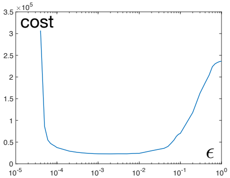


- ***distributionally robustness*** $\equiv$ ***regularization***: under minor conditions

Theorem: $\inf_{x\in\mathcal{X}} \sup_{\mathbb{Q}\in\mathbb{B}_\epsilon(\widehat{\mathbb{P}})} \mathbb{E}_{\xi\sim\mathbb{Q}}\left[c(\xi,x)\right] \equiv \min_{x\in\mathcal{X}} c\left(\widehat{\xi},x\right) + \epsilon\,\text{Lip}(c)\cdot\|x\|_p^\star$

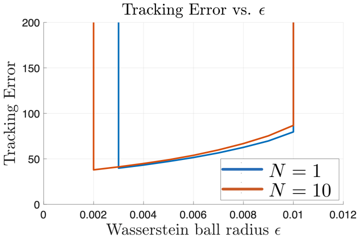
distributional robust formulation — previous regularized DeePC formulation

Cor: ℓ_∞-robustness in trajectory space $\iff$ ℓ_1-regularization of DeePC



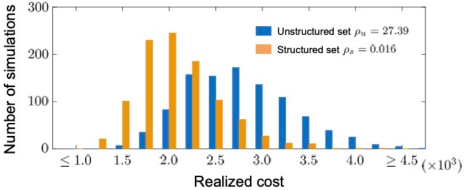
- ***measure concentration***: average matrix $\frac{1}{N}\sum_{i=1}^{N}\mathscr{H}_i(y^{\mathrm{d}})$ from i.i.d. experiments

$\Longrightarrow$ ambiguity set $\mathbb{B}_\epsilon(\widehat{\mathbb{P}})$ includes true $\mathbb{P}$ with high confidence if $\epsilon \sim 1/N^{1/\dim(\xi)}$



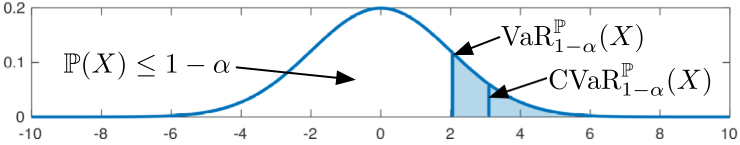
Further ingredients

- more ***structured uncertainty sets***: tractable reformulations (relaxations) & performance guarantees



- ***distributionally robust probabilistic constraints***

$\sup_{\mathbb{Q} \in \mathbb{B}_\epsilon(\widehat{\mathbb{P}})} \text{CVaR}_{1-\alpha}^{\mathbb{Q}} \iff$ averaging + regularization + tightening



- replace (finite) moving horizon estimation via $\binom{u_{\text{ini}}}{y_{\text{ini}}}$ by ***recursive Kalman filtering*** based on optimization solution $g^\star$ as hidden state . . .

how does DeePC relate to sequential SysID + control ?

surprise: **DeePC consistently beats models** across all our real-world case studies !

why ?!?

Comparison: direct vs. indirect control

indirect* ID-based *data-driven control

minimize control cost (u, y)

subject to (u, y) satisfy parametric model

where model $\in$ argmin id cost (u^d, y^d)

subject to model $\in$ LTI(n, ℓ) class } ID

ID projects data on the set of LTI models

- with parameters (n, ℓ)
- removes noise & thus lowers variance error
- suffers bias error if plant is not LTI(n, ℓ)

direct* regularized *data-driven control

minimize control cost (u, y) + $\lambda \cdot$ regularizer

subject to (u, y) consistent with (u^d, y^d) data

- ***regularization robustifies***
  $\rightarrow$ choosing λ makes it work
- ***no projection*** on LTI(n, ℓ)
  $\rightarrow$ no de-noising & no bias

hypothesis: ID wins in stochastic (variance) & DeePC in nonlinear (bias) case

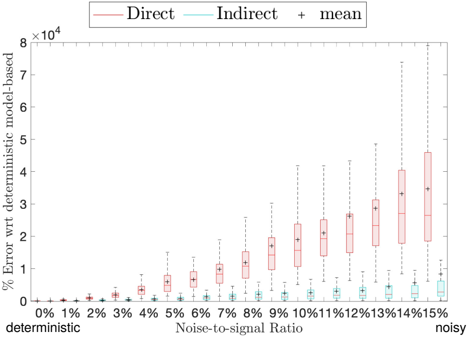
Case study: direct vs. indirect control

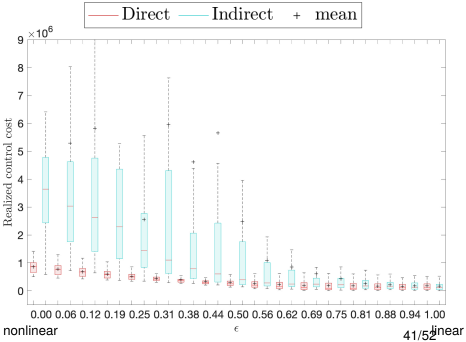
stochastic LTI case $\rightarrow$ indirect ID wins

- LQR control of 5th order LTI system
- Gaussian noise with varying noise to signal ratio (100 rollouts each case)
- ℓ_1-regularized DeePC, SysID via N4SID, & judicious hyper-parameters

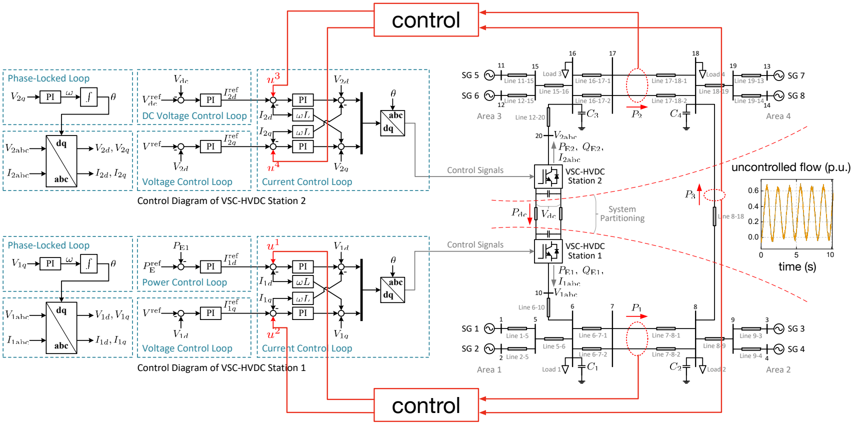
nonlinear case $\rightarrow$ direct DeePC wins

- Lotka-Volterra + control: $x^+ = f(x,u)$
- interpolated system $x^+ = \epsilon \cdot f_{\text{linearized}}(x,u) + (1-\epsilon)\cdot f(x,u)$
- same ID & DeePC as on the left & 100 initial x_0 rollouts for each ϵ



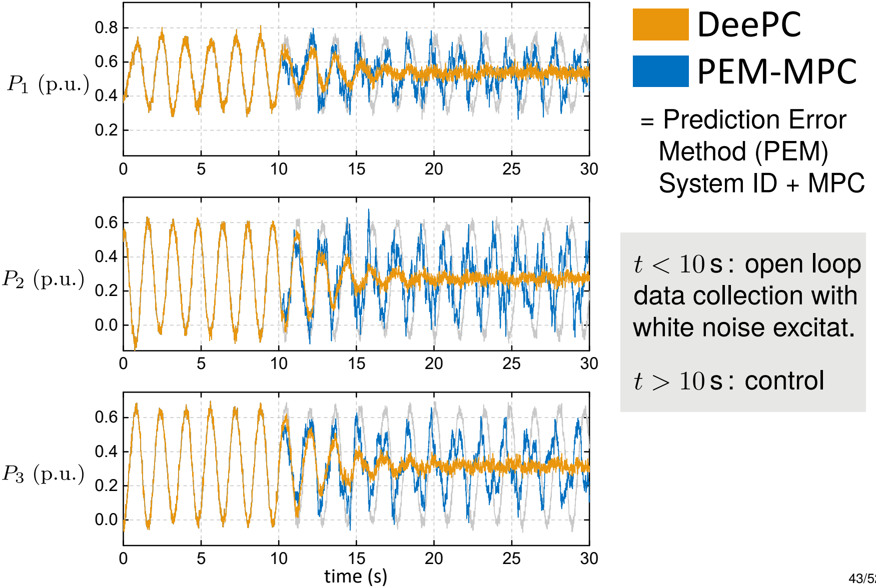


Power system case study revisited



- ***complex*** 4-area power ***system***: large ($n = 208$), few measurements (8), nonlinear, noisy, stiff, input constraints, & decentralized control
- ***control objective:*** damping of inter-area oscillations via HVDC link
- ***real-time*** MPC & DeePC prohibitive → choose T, T_{ini}, & T_{future} wisely

Centralized control

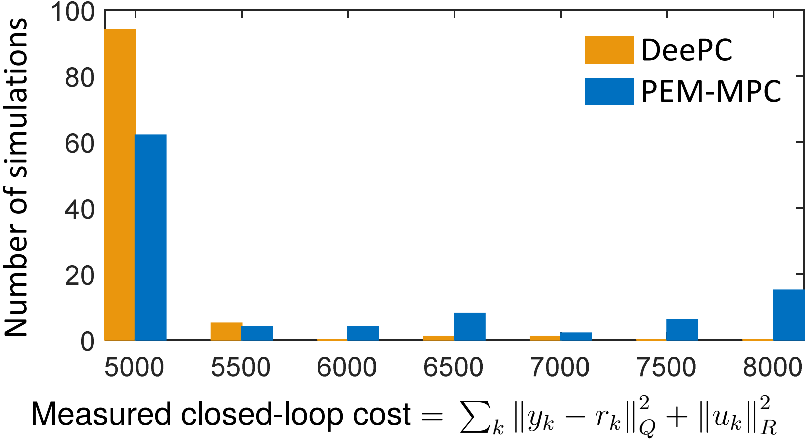


= Prediction Error Method (PEM) System ID + MPC

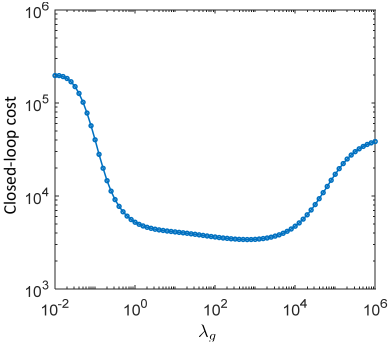
$t < 10\,$s : open loop data collection with white noise excitat.

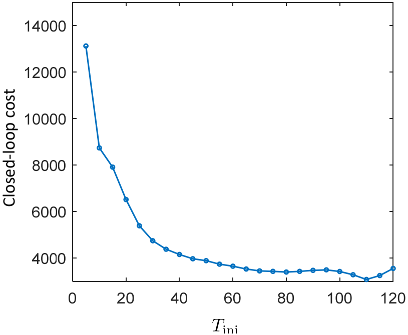
$t > 10\,$s : control

Performance: DeePC wins (clearly!)



DeePC hyper-parameter tuning





regularizer λ_g

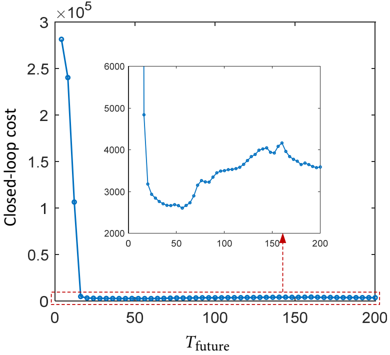
- for distributional robustness $\approx$ radius of Wasserstein ball
- wide range of sweet spots

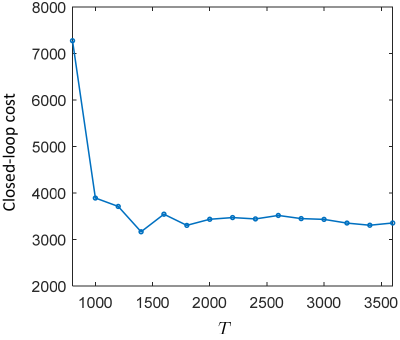
$\rightarrow$ choose $\lambda_g = 20$

estimation horizon T_{ini}

- for model complexity $\approx$ lag
- $T_{\text{ini}} \geq 50$ is sufficient & low computational complexity

$\rightarrow$ choose $T_{\text{ini}} = 60$





prediction horizon T_{future}

- nominal MPC is stable if horizon T_{future} long enough

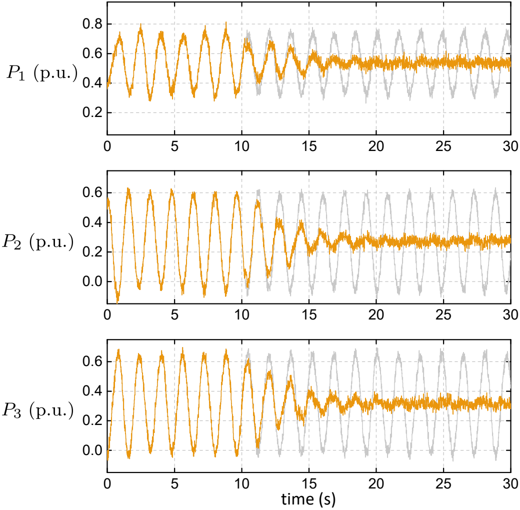
  $\rightarrow$ choose $T_{\text{future}} = 120$ and apply first 60 input steps

data length T

- long enough for low-rank condition but card(g) grows

  $\rightarrow$ choose $T = 1500$
  (data matrix $\approx$ square)

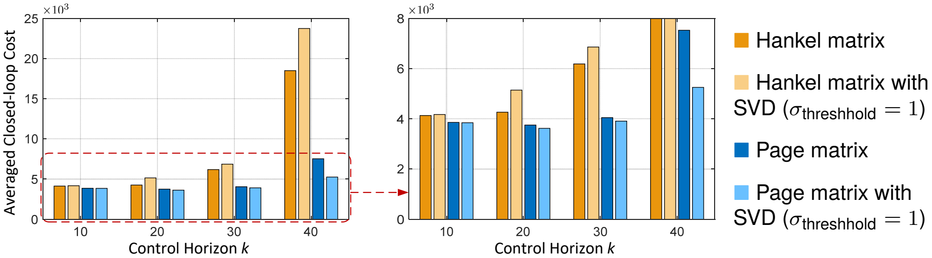
Computational cost



- $T = 1500$
- $\lambda_g = 20$
- $T_{\text{ini}} = 60$
- $T_{\text{future}} = 120$ & apply first 60 input steps
- sampling time = 0.02 s
- solver (OSQP) time = 1 s (on Intel Core i5 7200U)

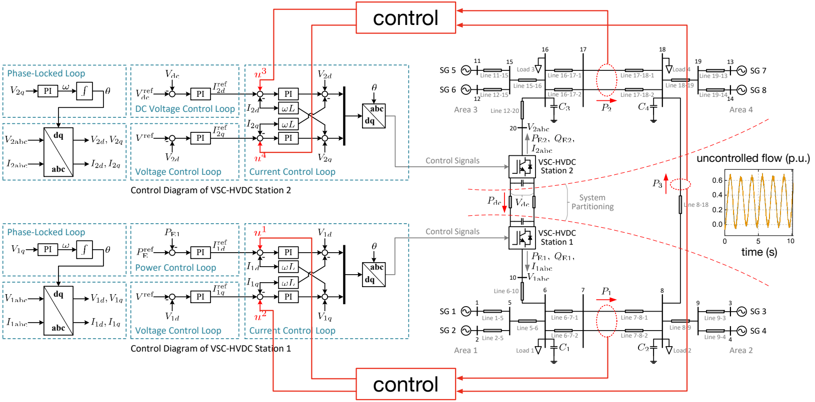
$\Rightarrow$ **implementable**

Comparison: Hankel & Page matrix



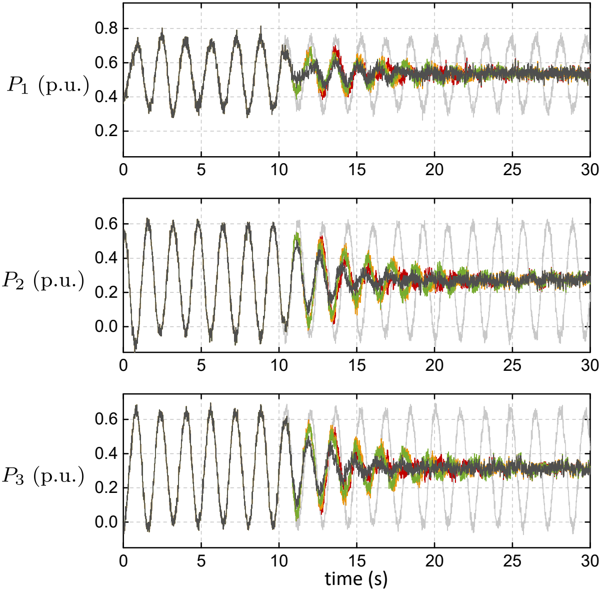
- comparison baseline: Hankel and Page matrices of ***same size***
- ***perfomance***: Page consistency beats Hankel matrix predictors
- offline ***denoising via SVD threshholding*** works wonderfully for Page though obviously not for Hankel (entries are constrained)
- effects very pronounced for ***longer horizon*** (= open-loop time)
- ***price-to-be-paid***: Page matrix predictor requires more data

Decentralized implementation



- ***plug'n'play MPC:*** treat interconnection P_3 as disturbance variable w with past disturbance w_{ini} measurable & future $w_{\text{future}} \in \mathcal{W}$ uncertain
- for each controller ***augment trajectory matrix*** with disturbance data w
- decentralized ***robust min-max DeePC:*** $\min_{g,u,y} \max_{w \in \mathcal{W}}$

Decentralized control performance



- colors correspond to different hyper-parameter settings (not discernible)
- ambiguity set $\mathcal{W}$ is ∞-ball (box)
- for computational efficiency $\mathcal{W}$ is downsampled (piece-wise linear)
- solver time ≈ 2.6 s

$\Rightarrow$ **implementable**

Conclusions

main take-aways

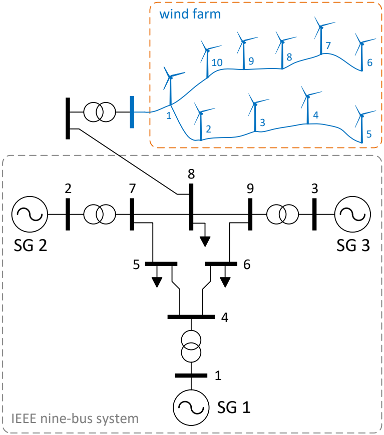
- matrix time series as predictive model
- robustness & side-info by regularization
- method that works in theory & practice

ongoing work

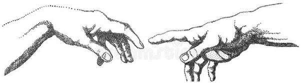
→ certificates for adaptive & nonlinear cases

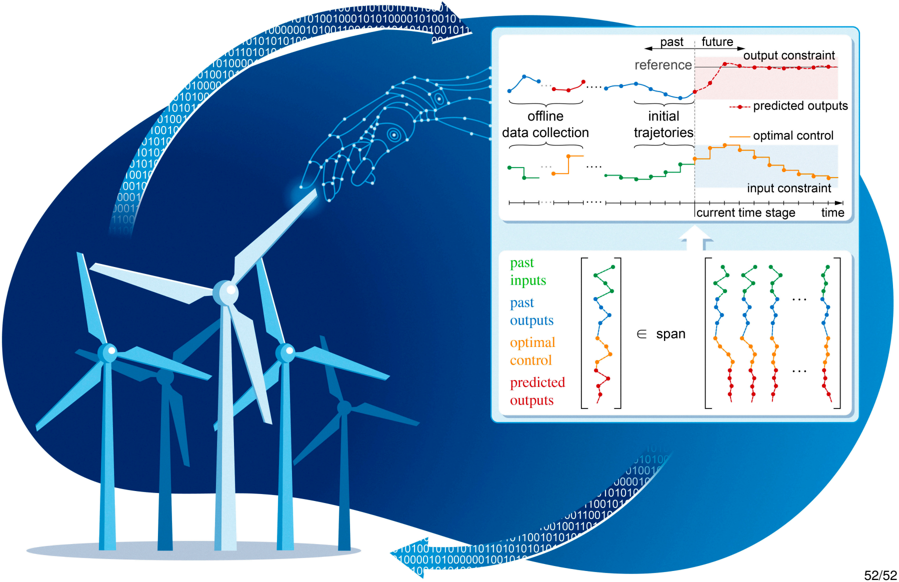
→ propagate optimal transport uncertainties

→ applications with a true “business case”



only catch (no-free-lunch) : optimization problems become large
→ models are compressed, de-noised, & tidied-up representations

Florian's version of 



Thanks !

Florian Dörfler

mail: dorfler@ethz.ch

[link] to homepage

[link] to related publications