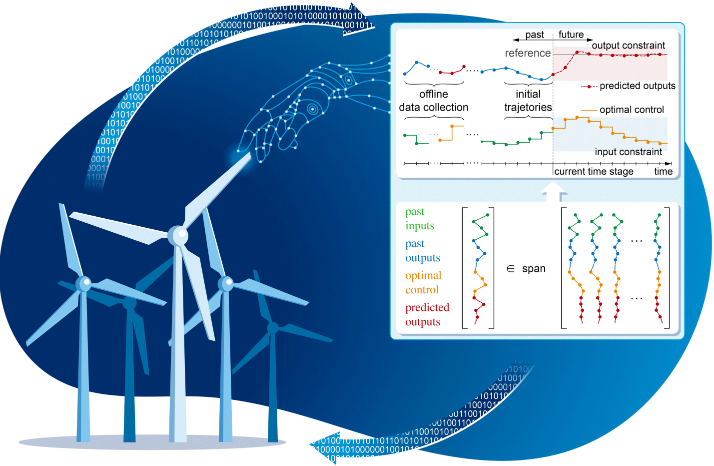
Data-Driven Control Based on Behavioral Systems Theory



Florian Dörfler

ETH Zürich

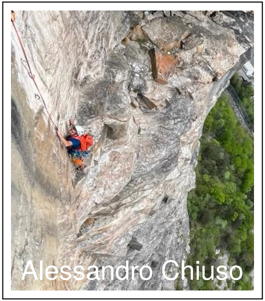
ERNSI 2023

Stockholm

Science & rope partners

Jeremy Coulson

Linbin Huang



Alessandro Chiuso

Ivan Markovsky & Alberto Padoan

John Lygeros

\+ many others

Roy Smith

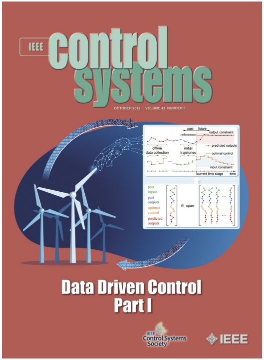
Thoughts on data in control systems

increasing role of ***data-centric methods*** in science / engineering / industry due to

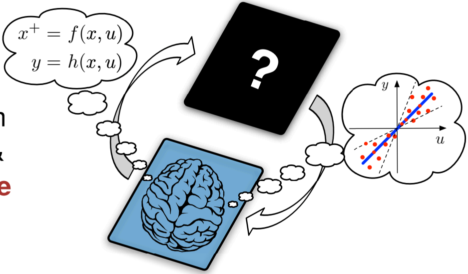
- ***methodological advances*** in statistics, optimization, & machine learning (ML)
- unprecedented availability of ***brute force***: deluge of data & computational power
- …and ***frenzy*** surrounding big data & ML

Make up your own opinion, but ML works too well to be ignored – also in control **?!?**

“One of the major developments in control over the past decade – & one of the most important moving forward – is the interaction of ML & control systems.” [CSS roadmap]



Scientific landscape



long & rich history (auto-tuning, system identification, adaptive control, RL, ...) & **vast & fragmented research landscape**

⟶ useful direct / indirect classification

direct data-driven control

minimize control cost (u, y)

subject to trajectory (u, y) compatible with data (u^d, y^d)

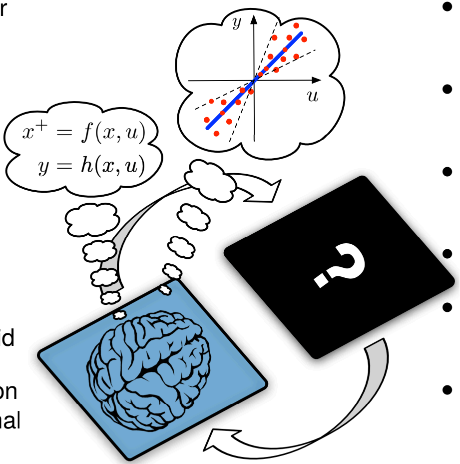
indirect (model-based) ***data-driven control***

model-based design {
minimize control cost (u, y)
subject to trajectory (u, y) compatible with the model

system identification {
where model $\in$ argmin fitting criterion (u^d, y^d)
subject to model belongs to certain class

Indirect vs. direct

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



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

lots of pros, cons, counterexamples, & ***no universal conclusions*** [`discussion`]

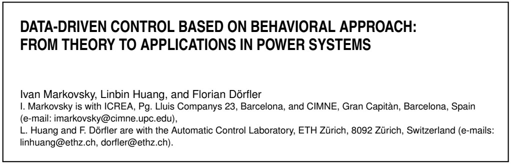
Today's menu

1. {behavioral systems} ∩ {subspace ID}: ***fundamental lemma***
2. potent direct method: data-enabled predictive control ***DeePC***
3. salient ***regularizations*** for robustification & inject side info
4. ***case studies*** from robotics & energy domain + tomatoes ☺

blooming literature (2-3 ArXiv / week)

→ tutorial [`link`] to get started

- [`link`] to graduate school material
- [`link`] to survey
- [`link`] to related bachelor lecture
- [`link`] to related publications



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).

Behavioral view on dynamical systems

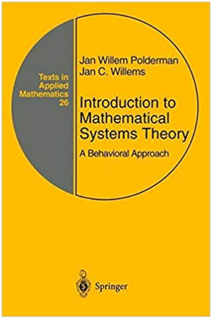
Definition: A discrete-time ***dynamical system*** is a 3-tuple $(\mathbb{Z}_{\geq 0}, \mathbb{W}, \mathscr{B})$ where

(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



Texts in Applied Mathematics 26

Jan Willem Polderman
Jan C. Willems

Introduction to Mathematical Systems Theory

A Behavioral Approach

Springer

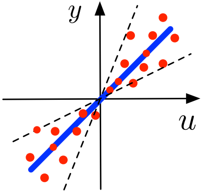
Definition: The dynamical system $(\mathbb{Z}_{\geq 0}, \mathbb{W}, \mathscr{B})$ is

(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***



LTI systems & matrix time series

foundation of subspace system identification & signal recovery algorithms

$\big(u(t), y(t)\big)$ satisfy LTI ***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)

$$\mathscr{H}\begin{pmatrix} u^d \\ y^d \end{pmatrix} = \begin{array}{c} \begin{bmatrix} \begin{pmatrix} u^d_{1,1} \\ y^d_{1,1} \end{pmatrix} & \begin{pmatrix} u^d_{1,2} \\ y^d_{1,2} \end{pmatrix} & \begin{pmatrix} u^d_{1,3} \\ y^d_{1,3} \end{pmatrix} & \cdots \\ \begin{pmatrix} u^d_{2,1} \\ y^d_{2,1} \end{pmatrix} & \begin{pmatrix} u^d_{2,2} \\ y^d_{2,2} \end{pmatrix} & \begin{pmatrix} u^d_{2,3} \\ y^d_{2,3} \end{pmatrix} & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ \begin{pmatrix} u^d_{T,1} \\ y^d_{T,1} \end{pmatrix} & \begin{pmatrix} u^d_{T,2} \\ y^d_{T,2} \end{pmatrix} & \begin{pmatrix} u^d_{T,3} \\ y^d_{T,3} \end{pmatrix} & \cdots \end{bmatrix} \\ \begin{matrix} \underbrace{\qquad\qquad}_{\text{1st experiment}} & \underbrace{\qquad\qquad}_{\text{2nd}} & \underbrace{\qquad\qquad}_{\text{3rd} \ldots} \end{matrix} \end{array}$$

Fundamental Lemma

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

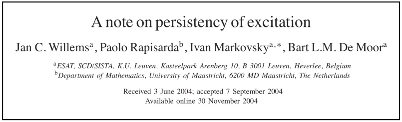
$$\underbrace{\text{set of all } T\text{-length trajectories} = \left\{ (u, y) \in \mathbb{R}^{(m+p)T} : \exists x \in \mathbb{R}^{nT} \text{ s.t. } x^+ = Ax + Bu\,,\ y = Cx + Du \right\}}_{\text{parametric state-space model}} = \underbrace{\text{colspan} \begin{bmatrix} \begin{pmatrix} u_{1,1}^d \\ y_{1,1}^d \end{pmatrix} & \begin{pmatrix} u_{1,2}^d \\ y_{1,2}^d \end{pmatrix} & \begin{pmatrix} u_{1,3}^d \\ y_{1,3}^d \end{pmatrix} & \cdots \\ \begin{pmatrix} u_{2,1}^d \\ y_{2,1}^d \end{pmatrix} & \begin{pmatrix} u_{2,2}^d \\ y_{2,2}^d \end{pmatrix} & \begin{pmatrix} u_{2,3}^d \\ y_{2,3}^d \end{pmatrix} & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ \begin{pmatrix} u_{T,1}^d \\ y_{T,1}^d \end{pmatrix} & \begin{pmatrix} u_{T,2}^d \\ y_{T,2}^d \end{pmatrix} & \begin{pmatrix} u_{T,3}^d \\ y_{T,3}^d \end{pmatrix} & \cdots \end{bmatrix}}_{\text{raw data (every column is an experiment)}}$$

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

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

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]

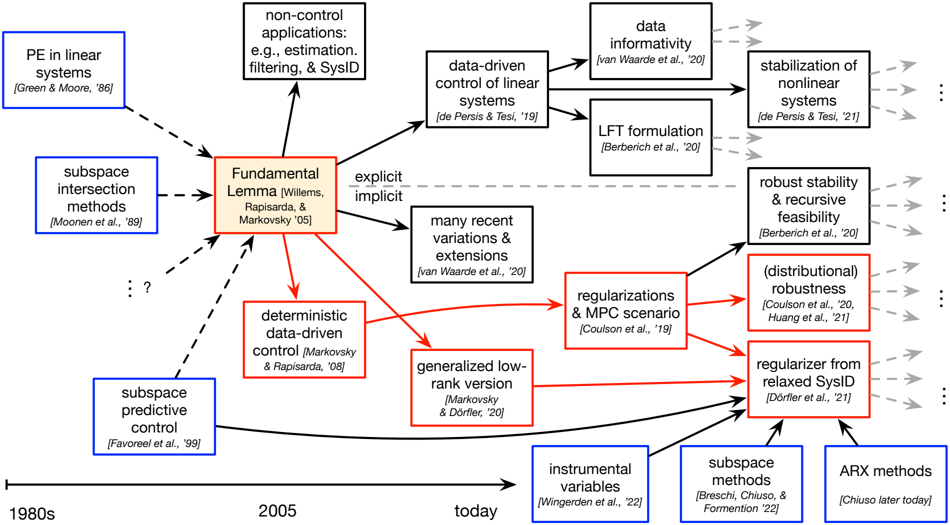
[a]*ESAT, SCD/SISTA, K.U. Leuven, Kasteelpark Arenberg 10, B 3001 Leuven, Heverlee, Belgium*
[b]*Department of Mathematics, University of Maastricht, 6200 MD Maastricht, The Netherlands*

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***

Bird’s view: SysID & today’s path



PE in linear systems *[Green & Moore, ’86]*

subspace intersection methods *[Moonen et al., ’89]*

⋮ ?

subspace predictive control *[Favoreel et al., ’99]*

non-control applications: e.g., estimation. filtering, & SysID

Fundamental Lemma [Willems, Rapisarda, & Markovsky ’05]

explicit

implicit

deterministic data-driven control *[Markovsky & Rapisarda, ’08]*

data-driven control of linear systems *[de Persis & Tesi, ’19]*

many recent variations & extensions *[van Waarde et al., ’20]*

generalized low-rank version *[Markovsky & Dörfler, ’20]*

data informativity *[van Waarde et al., ’20]*

LFT formulation *[Berberich et al., ’20]*

regularizations & MPC scenario *[Coulson et al., ’19]*

stabilization of nonlinear systems *[de Persis & Tesi, ’21]*

robust stability & recursive feasibility *[Berberich et al., ’20]*

(distributional) robustness *[Coulson et al., ’20, Huang et al., ’21]*

regularizer from relaxed SysID *[Dörfler et al., ’21]*

⋮

⋮

⋮

⋮

instrumental variables *[Wingerden et al., ’22]*

subspace methods *[Breschi, Chiuso, & Formention ’22]*

ARX methods *[Chiuso later today]*

1980s

2005

today

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

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

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 an LTI plant, deterministic, & with known model, MPC is the ***gold standard of control***.

Data-**e**nabl**e**d **P**redictive **C**ontrol (**DeePC**)

$$\begin{aligned}
\underset{g,u,y}{\text{minimize}} \quad & \sum_{k=1}^{T_{\text{future}}} \|y_k - r_k\|_Q^2 + \|u_k\|_R^2 \\
\text{subject to} \quad & \mathscr{H}\begin{pmatrix} u^{\mathsf{d}} \\ y^{\mathsf{d}} \end{pmatrix} \cdot g = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ \hline u \\ y \end{bmatrix} \\
& \left.\begin{array}{l} u_k \in \mathcal{U} \\ y_k \in \mathcal{Y} \end{array}\right\} \quad \forall k \in \{1, \dots, T_{\text{future}}\}
\end{aligned}$$

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 make it work

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

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

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

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} ∼ low-order basis ∼ low-rank surrogate

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

$$\min_x \max_{\|\Delta\| \leq \rho} \|(A+\Delta)x - b\| \underset{\text{tight}}{\leq} \min_x \max_{\|\Delta\| \leq \rho} \|Ax - b\| + \|\Delta x\| = \min_x \|Ax - b\| + \rho\|x\|$$

regularization

$\Updownarrow$

incorporating priors
+ implicit SysID

Regularization = relaxing low-rank approximation in pre-processing

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

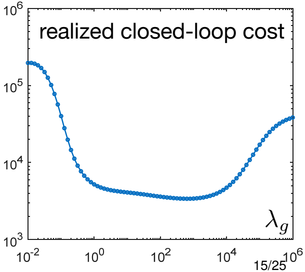
} optimal control

} low-rank approximation

↓ sequence of convex relaxations ↓

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

ℓ_1-regularization $=$ relaxation of low-rank approximation & smoothened order selection



Regularization ⇔ reformulate subspace ID

partition data as in subspace ID:

$$\mathscr{H}\begin{pmatrix}u^{\mathsf{d}}\\ y^{\mathsf{d}}\end{pmatrix} \sim \begin{bmatrix}U_{\mathsf{p}}\\ Y_{\mathsf{p}}\\ U_{\mathsf{f}}\\ Y_{\mathsf{f}}\end{bmatrix} \begin{array}{l}\left.\vphantom{\begin{matrix}U\\ Y\end{matrix}}\right\}(m+p)T_{\mathsf{ini}}\\ \left.\vphantom{\begin{matrix}U\\ Y\end{matrix}}\right\}(m+p)T_{\mathsf{future}}\end{array}$$

ID of optimal multi-step predictor

as in SPC: $K^{\star} = Y_F \begin{bmatrix}U_{\mathsf{p}}\\ Y_{\mathsf{p}}\\ U_{\mathsf{f}}\end{bmatrix}^{\dagger}$

→ ***indirect SysID + control*** problem

$$\begin{aligned} &\underset{u,y}{\text{minimize}} && \text{control cost}(u,y)\\ &\text{subject to} && y = K^{\star}\begin{bmatrix}u_{\mathsf{ini}}\\ y_{\mathsf{ini}}\\ u\end{bmatrix}\\ &\text{where} && K^{\star} = \underset{K}{\text{argmin}}\left\| Y_F - K\begin{bmatrix}U_{\mathsf{p}}\\ Y_{\mathsf{p}}\\ U_{\mathsf{f}}\end{bmatrix}\right\| \end{aligned}$$

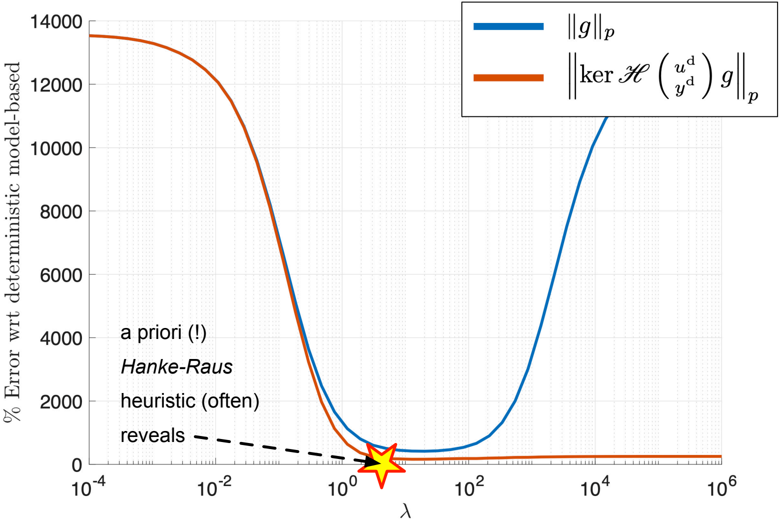
The above is ***equivalent to regularized DeePC***

where $\text{Proj}\begin{pmatrix}u^{\mathsf{d}}\\ y^{\mathsf{d}}\end{pmatrix}$ projects

orthogonal to $\ker\begin{bmatrix}U_{\mathsf{p}}\\ Y_{\mathsf{p}}\\ U_{\mathsf{f}}\end{bmatrix}$

$$\begin{aligned} &\underset{g,u,y}{\text{minimize}} && \text{control cost}(u,y) + \lambda_g \left\|\text{Proj}\begin{pmatrix}u^{\mathsf{d}}\\ y^{\mathsf{d}}\end{pmatrix} g\right\|_p\\ &\text{subject to} && \mathscr{H}\begin{pmatrix}u^{\mathsf{d}}\\ y^{\mathsf{d}}\end{pmatrix}\cdot g = \begin{bmatrix}u_{\mathsf{ini}}\\ y_{\mathsf{ini}}\\ u\\ y\end{bmatrix} \end{aligned}$$

Regularizations applied to stochastic LTI system & hyper-parameter selection



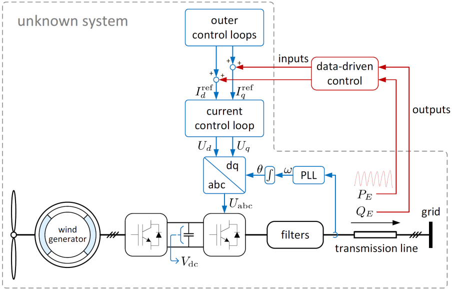
a priori (!)

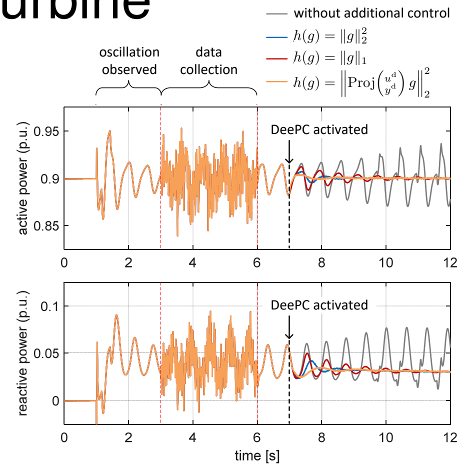
Hanke-Raus

heuristic (often)

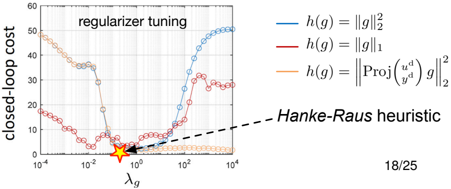
reveals

Case study: wind turbine

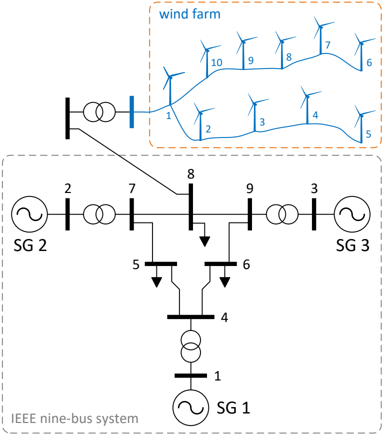


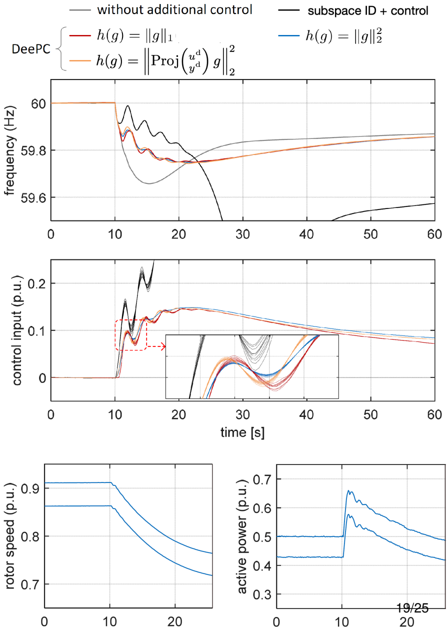


- turbine & grid model ***unknown*** to commissioning engineer & operator
- detailed ***industrial model***: 37 states & highly nonlinear (abc ↔ dq, MPTT, PLL, power specs, dynamics, etc.)
- weak grid → ***oscillations + sync loss***
- disturbance to be rejected by ***DeePC***



Case study +++ : wind farm





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

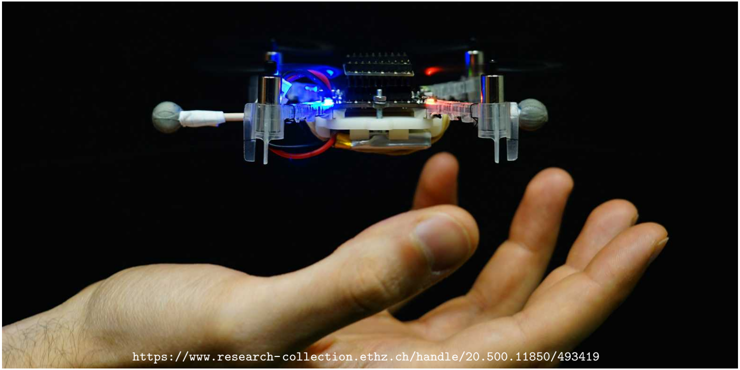
Towards a theory for nonlinear systems

naive idea: lift nonlinear system to large/∞-dim. bi-/linear system

→ Carleman, Volterra, Fliess, Koopman, Sturm-Liouville methods

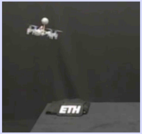
→ nonlinear dynamics can be approximated by LTI on finite horizon

regularization singles out relevant features / basis functions in data



https://www.research-collection.ethz.ch/handle/20.500.11850/493419

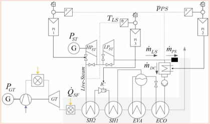
Works very well across case studies



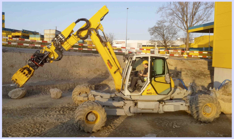
quad coptor fig-8 tracking

quadruped (by Fawcett, Afsari Amers, & Hamed)

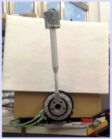
greenhouse automation (by Automatoes)



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

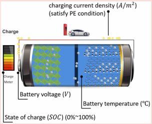


robotic excavator

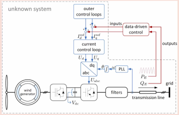


pendulum swing up

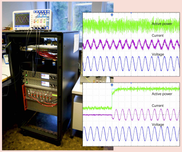
traffic coordination (by J. Wang et al.)



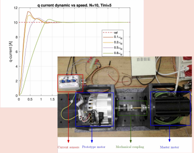
battery charging (by K. Chen et al.)



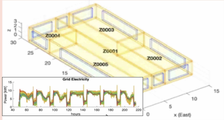
wind turbine control



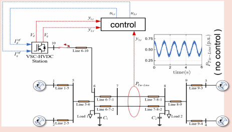
grid-connected converter



synchronous motor drive



energy hub & building automation



power system oscillation damping

regularization

$$\Updownarrow$$

robustification

Distributional robustification beyond LTI

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

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

⇒ ***poor out-of-sample performance*** of above sample-average solution $x^\star$

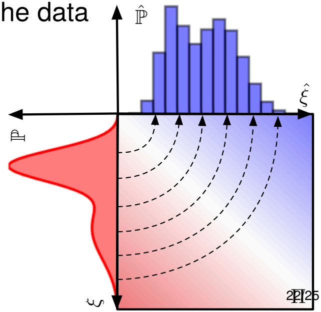
for real problem: $\mathbb{E}_{\xi \sim \mathbb{P}}\big[c\left(\xi, x^\star\right)\big]$ where $\mathbb{P}$ is the *unknown distribution* of ξ

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

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

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

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

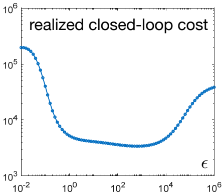


- ***distributionally robustness*** $\equiv$ ***regularization*** : under minor conditions

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

Cor : ℓ_∞-robustness in trajectory space

$\iff$ ℓ_1-regularization of DeePC

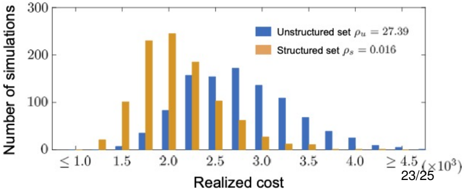


- similar for ***distributionally robust constraints***
- ***measure concentration***: average N i.i.d. data sets & $\epsilon \sim 1/N^{1/\dim(\xi)}$

  $\Longrightarrow \mathbb{P} \in \mathbb{B}_\epsilon(\widehat{\mathbb{P}})$ with high confidence
- more ***structured uncertainty sets*** :

  tractable reformulations (relaxations)

  & performance guarantees



white elephant: how does DeePC perform against SysID + control ?

surprise: **DeePC consistently beats** (certainty-equivalence) **identification & control** of LTI models across all real 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 $\left.\begin{array}{l}\text{model} \in \text{argmin id cost } (u^d, y^d) \\ \text{subject to model} \in \text{LTI}(n, \ell) \text{ class}\end{array}\right\} \text{ID}$

ID projects data on LTI class to ***learn predictor***

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

- no de-noising & no bias
- ***regularization robustifies prediction*** (not predictor)
- ***trade-off*** ID & control costs

take-away : ID wins when model class is known, noise is well behaved, & control task doesn't bias ID. Otherwise, ***DeePC can beat ID*** …it often does !

Conclusions

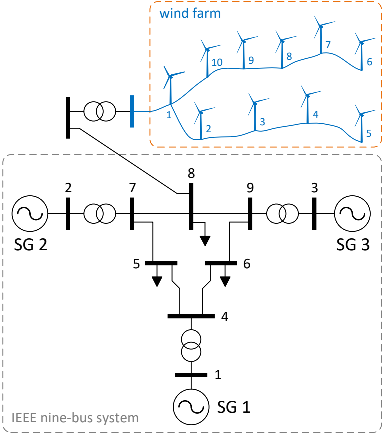
main take-aways

- matrix time series as predictive model
- robustness & side-info by regularization
- method that works in theory & practice
- focus is robust prediction not predictor ID

ongoing work

→ certificates for adaptive & nonlinear cases

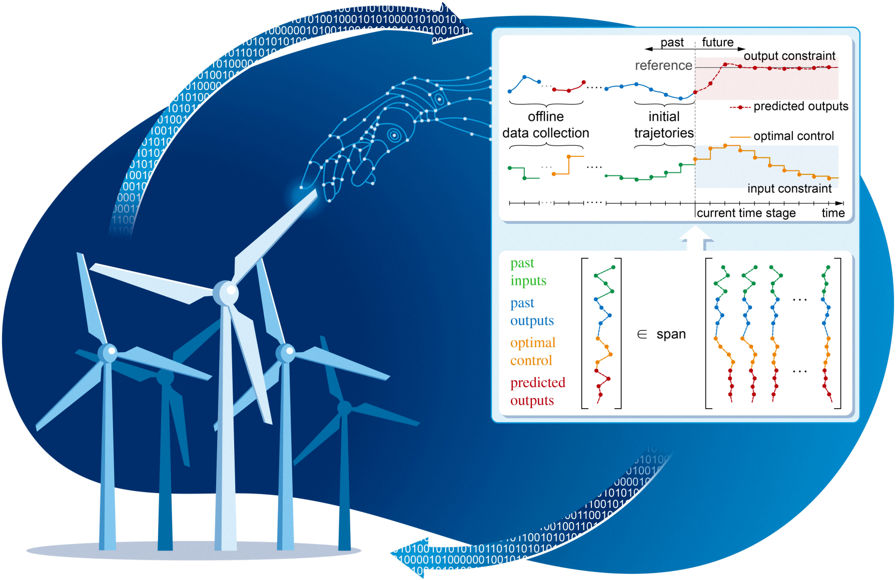
→ applications with a true “business case”, push TRL scale, & industry collaborations



questions we should discuss

- catch? violate no-free-lunch theorem ? → more real-time computation
- DeePC = subspace ID + robustification ? → more accessible & flexible
- when does direct beat indirect ? → Id4Control & bias/variance issues ?

Thanks !


past
future
reference
output constraint
predicted outputs
offline
data collection
initial
trajetories
optimal control
input constraint
current time stage
time
past
inputs
past
outputs
optimal
control
predicted
outputs
∈ span
…
…