Data-Driven Control in Autonomous Energy Systems

Florian Dörfler

ETH Zürich

UW Clean Energy Institute Seminar

Acknowledgements

Jeremy Coulson

Linbin Huang

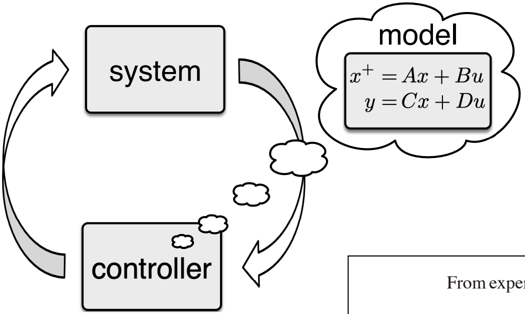
Paul Beuchat

John Lygeros

Ivan Markovsky

Ezzat Elokda

Perspectives on model-based control



→ ***models*** useful for system analysis, design, estimation, … ***control***

→ ***modeling*** from first principles & ***system ID***

recurring themes

- modeling & system ID are very expensive
- models not always useful for control
- need for end-to-end automation solutions

From experiment design to closed-loop control☆

Håkan Hjalmarsson*

1. Introduction

Ever increasing productivity demands and environmental standards necessitate more and more advanced control methods to be employed in industry. However, such methods usually require a model of the process and modeling and system identification are expensive. Quoting (Ogunnaike, 1996):

"*It is also widely recognized, however, that obtaining the process model is the single most time consuming task in the application of model-based control.*"

In Hussain (1999) it is reported that three quarters of the total costs associated with advanced control projects can be attributed to modeling. It is estimated that models exist for far less than one percent of all processes in regulatory control. According to Desborough and Miller (2001), one of the few instances when the cost of dynamic modeling can be justified is for the commissioning of model predictive controllers.

It has also been recognized that models for control pose special considerations. Again quoting (Ogunnaike, 1996):

"*There is abundant evidence in industrial practice that when modeling for control is not based on criteria related to the actual end use, the results can sometimes be quite disappointing.*"

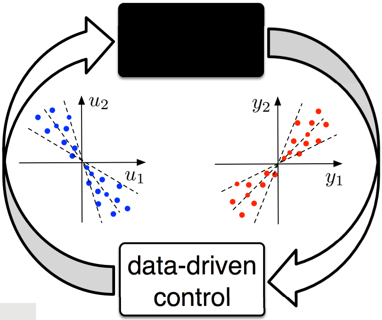
Hence, efficient modeling and system identification techniques suited for industrial use and tailored for control design applications have become important enablers for industrial advances. The Panel for Future Directions in Control, (Murray, Åström, Boyd, Brockett, & Stein, 2003), has identified *automatic synthesis of control algorithms, with integrated validation and verification* as one of the major future challenges in control. Quoting (Murray et al., 2003):

"*Researchers need to develop much more powerful design tools that automate the entire control design process from*

Control in a data-rich world

- ever-growing trend in CS & applications: ***data-driven control*** by-passing models
- canonical problem: ***black/gray-box system control*** based on I/O samples

Q: Why give up physical modeling & reliable model-based algorithms ?



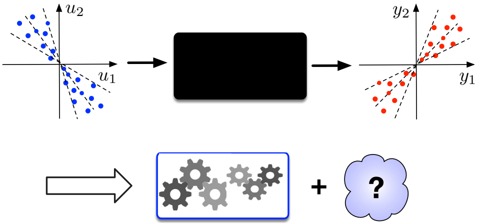
Data-driven control is ***viable alternative*** when

- models are too complex to be useful
  e.g., wind farm interactions & building automation
- first-principle models are not conceivable
  e.g., human-operator-in-the-loop & demand control
- modeling & system ID is too cumbersome
  e.g., drives & electronics applications

Central promise: *It is often easier to learn control policies directly from data, rather than learning a model.*

Example: PID [Åström, ’73]

Snippets from the literature

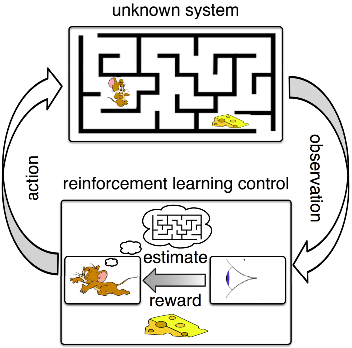


indirect data-driven control:

sequential system ID + uncertainty quantification + robust control

→ recent end-to-end design pipelines with finite-sample guarantees

ø ID seeks best but not most useful model: "easier to learn policies . . ."



direct data-driven control:

reinforcement learning / stochastic adaptive control / approximate dynamic programming

→ spectacular theoretic & practical advances

→ more brute force storage/computation/data

ø not suitable for physical systems: real-time, safety-critical, continuous today: something very different

Contents

I. Data-Enabled Predictive Control (DeePC): Basic Idea

J. Coulson, J. Lygeros, and F. Dörfler. *Data-Enabled Predictive Control: In the Shallows of the DeePC*. [arxiv.org/abs/1811.05890].

II. From Heuristics & Numerical Promises to Theorems

J. Coulson, J. Lygeros, and F. Dörfler. *Distributionally Robust Chance Constrained Data-enabled Predictive Control*. [https://arxiv.org/abs/2006.01702].

I. Markovsky and F. Dörfler. *Identifiability in the Behavioral Setting.* [link]

III. Application: End-to-End Automation in Energy & Robotics

L. Huang, J. Coulson, J. Lygeros, and F. Dörfler. *Decentralized Data-Enabled Predictive Control for Power System Oscillation Damping*. [arxiv.org/abs/1911.12151].

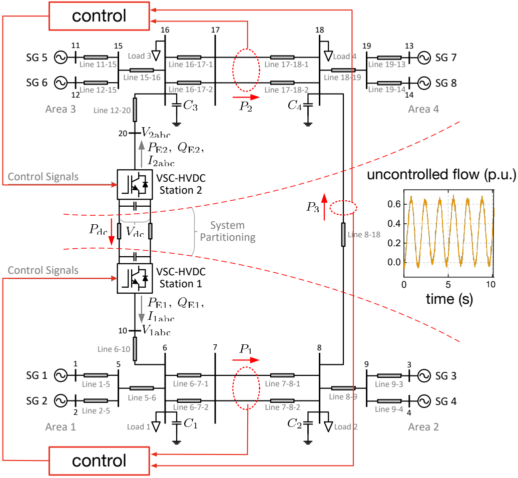
E. Elokda, J. Coulson, P. Beuchat, J. Lygeros, and F. Dörfler. *Data-Enabled Predictive Control for Quadcopters*. [link].

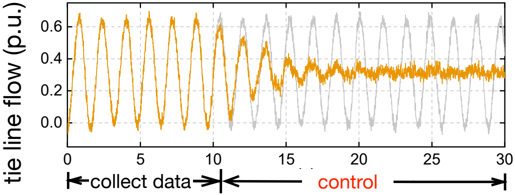
Preview

complex 4-area power ***system***:
large ($n = 208$), few sensors (8), nonlinear, noisy, stiff, input constraints, & decentralized control specifications

control objective: oscillation damping without model

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





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

→ automating ourselves

Reality check: magic or hoax ?

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

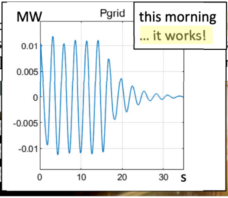
- on the **world's most complex engineered system** (the electric grid),
- using the **world's biggest actuators** (Gigawatt-sized HVDC links),
- and subject to **real-time, safety, & stability constraints** . . . right?

Dear Linbin and Florian,

last Friday

I just submitted a very favourable review of your paper [...] which I believe could be of importance to our work at Hitachi Power Grids. We do have [...] require off-line tuning that [...] comissisioning engineer can do on his own [...] an adaptive approach would be very interesting.

If p... ntralized DeePC approach with our more detailed HVDC sys... ation problem. Could so some code be made available [...] ng together to do such a demonstration ? [...]



this morning
... it works!

so at least someone believes that DeePC is practically useful . . .

Behavioral view on LTI systems

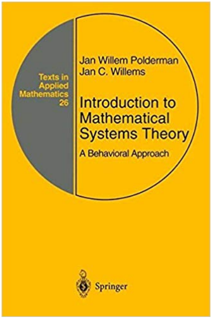
Definition: A discrete-time ***dynamical system*** is a 3-tuple $(\mathbb{Z}_{\geq 0}, \mathbb{W}, \mathscr{B})$ where

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

(ii) $\mathbb{W}$ is a signal space, and

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

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

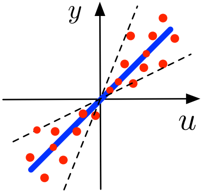


Definition: The dynamical system $(\mathbb{Z}_{\geq 0}, \mathbb{W}, \mathscr{B})$ is

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

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

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



LTI systems and 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)

$\Longleftarrow$

under assumptions

$\Longrightarrow$

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

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

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

Fundamental Lemma [Willems et al. '05], [Markovsky & Dörfler '20]

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

$$\underbrace{\begin{array}{l} \text{set of all } T\text{-length trajectories } = \\ \left\{ (u,y) \in \mathbb{R}^{(m+p)T} : \exists x \in \mathbb{R}^n \text{ s.t.} \right. \\ \left. x^+ = Ax + Bu \,,\; y = Cx + Du \right\} \end{array}}_{\text{parametric state-space model}} \quad = \quad \underbrace{\text{colspan} \begin{bmatrix} \begin{pmatrix} u_{1,1}^d \\ y_{1,1}^d \end{pmatrix} & \begin{pmatrix} u_{2,1}^d \\ y_{2,1}^d \end{pmatrix} & \begin{pmatrix} u_{3,1}^d \\ y_{3,1}^d \end{pmatrix} & \dots \\ \begin{pmatrix} u_{1,2}^d \\ y_{1,2}^d \end{pmatrix} & \begin{pmatrix} u_{2,2}^d \\ y_{2,2}^d \end{pmatrix} & \begin{pmatrix} u_{3,2}^d \\ y_{3,2}^d \end{pmatrix} & \dots \\ \vdots & \vdots & \vdots & \vdots \\ \begin{pmatrix} u_{1,T}^d \\ y_{1,T}^d \end{pmatrix} & \begin{pmatrix} u_{2,T}^d \\ y_{2,T}^d \end{pmatrix} & \begin{pmatrix} u_{3,T}^d \\ y_{3,T}^d \end{pmatrix} & \dots \end{bmatrix}}_{\text{non-parametric model from raw data}}$$

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

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

parametric state-space model — non-parametric model from raw data

all trajectories constructible from finitely many previous trajectories

- can also use other ***matrix data structures***: (mosaic) Hankel, Page, ...
- ***novelty (?)***: motion primitives, DMD, dictionary learning, subspace system id, ... all implicitly rely on this equivalence → c.f. “fundamental”
- ***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

Control from matrix time series data

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

We are all writing merely the dramatic corollaries . . .

implicit & stochastic

→ Ivan Markovsky & ourselves

explicit & deterministic

→ Claudio de Persis & Pietro Tesi

→ ***lots of recent momentum*** (∼ 1 ArXiv / week) with contributions by
Scherer, Allgöwer, Camlibel, Trentelman, Pappas, Fischer, Pasqualetti, Goulart, Mesbahi, . . .

→ more classic ***subspace predictive control*** (De Moor) literature

Data-driven prediction [Markovsky & Rapisarda '08]

Problem: predict future output $y \in \mathbb{R}^{p \cdot T_{\text{future}}}$ based on

- input signal $u \in \mathbb{R}^{m \cdot T_{\text{future}}}$ → to predict forward
- past data $\text{col}(u^{\text{d}}, y^{\text{d}}) \in \mathscr{B}_{T_{\text{data}}}$ → to form trajectory matrix

Solution: given $(u_1, \dots, u_{T_{\text{future}}})$ → compute $\boldsymbol{g}$ & $(y_1, \dots, y_{T_{\text{future}}})$ from

$$\mathscr{H}_{T_{\text{future}}}\begin{pmatrix} u^{\text{d}} \\ y^{\text{d}} \end{pmatrix} \boldsymbol{g} = \left[\begin{array}{cccc} u^{\text{d}}_{1,1} & u^{\text{d}}_{2,1} & u^{\text{d}}_{3,1} & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ u^{\text{d}}_{1,T_{\text{future}}} & u^{\text{d}}_{2,T_{\text{future}}} & u^{\text{d}}_{3,T_{\text{future}}} & \cdots \\ \hline y^{\text{d}}_{1,1} & y^{\text{d}}_{2,1} & y^{\text{d}}_{3,1} & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ y^{\text{d}}_{1,T_{\text{future}}} & y^{\text{d}}_{2,T_{\text{future}}} & y^{\text{d}}_{3,T_{\text{future}}} & \cdots \end{array}\right] \boldsymbol{g} = \left[\begin{array}{c} u_1 \\ \vdots \\ u_{T_{\text{future}}} \\ \hline y_1 \\ \vdots \\ y_{T_{\text{future}}} \end{array}\right]$$

Issue: predicted output is not unique → need to set initial conditions !

Data-driven prediction & estimation

Refined problem: predict future output $y \in \mathbb{R}^{p \cdot T_{\text{future}}}$ based on

- initial trajectory col$(u_{\text{ini}}, y_{\text{ini}}) \in \mathbb{R}^{(m+p)\cdot T_{\text{ini}}}$ $\rightarrow$ to estimate initial x_{ini}
- input signal $u \in \mathbb{R}^{m \cdot T_{\text{future}}}$ $\rightarrow$ to predict forward
- past data col$(u^{\text{d}}, y^{\text{d}}) \in \mathscr{B}_{T_{\text{data}}}$ $\rightarrow$ to form trajectory matrix

Solution: given u & col$(u_{\text{ini}}, y_{\text{ini}})$ $\rightarrow$ compute $\boldsymbol{g}$ & y from

$$\begin{bmatrix} \mathscr{H}_{T_{\text{ini}}}\begin{pmatrix} u^{\text{d}} \\ y^{\text{d}} \end{pmatrix} \\ \mathscr{H}_{T_{\text{future}}}\begin{pmatrix} u^{\text{d}} \\ y^{\text{d}} \end{pmatrix} \end{bmatrix} \boldsymbol{g} = \left[\begin{array}{cccc} u^{\text{d}}_{1,1} & u^{\text{d}}_{2,1} & u^{\text{d}}_{3,1} & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ u^{\text{d}}_{1,T_{\text{ini}}} & u^{\text{d}}_{2,T_{\text{ini}}} & u^{\text{d}}_{3,T_{\text{ini}}} & \cdots \\ \hline y^{\text{d}}_{1,1} & y^{\text{d}}_{2,1} & y^{\text{d}}_{3,1} & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ y^{\text{d}}_{1,T_{\text{ini}}} & y^{\text{d}}_{2,T_{\text{ini}}} & y^{\text{d}}_{3,T_{\text{ini}}} & \cdots \\ \hline u^{\text{d}}_{1,T_{\text{ini}}+1} & u^{\text{d}}_{2,T_{\text{ini}}+1} & u^{\text{d}}_{3,T_{\text{ini}}+1} & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ u^{\text{d}}_{1,T_{\text{ini}}+T_{\text{future}}} & u^{\text{d}}_{2,T_{\text{ini}}+T_{\text{future}}} & u^{\text{d}}_{3,T_{\text{ini}}+T_{\text{future}}} & \cdots \\ \hline y^{\text{d}}_{1,T_{\text{ini}}+1} & y^{\text{d}}_{2,T_{\text{ini}}+1} & y^{\text{d}}_{3,T_{\text{ini}}+1} & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ y^{\text{d}}_{1,T_{\text{ini}}+T_{\text{future}}} & y^{\text{d}}_{2,T_{\text{ini}}+T_{\text{future}}} & y^{\text{d}}_{3,T_{\text{ini}}+T_{\text{future}}} & \cdots \end{array}\right] \boldsymbol{g} = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \\ y \end{bmatrix}$$

$\Rightarrow$ observability condition: if $T_{\text{ini}} \geq$ lag of system, then y is ***unique***

Output **M**odel **P**redictive **C**ontrol

The canonical receding-horizon ***MPC optimization problem***:

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

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

model for **prediction** over $k \in [0, T_{\text{future}}-1]$

model for **estimation** (many variations)

hard operational or safety **constraints**

For a deterministic LTI plant and an exact model of the plant,
MPC is the ***gold standard of control***: safe, optimal, tracking, ...

Data-Enabled Predictive Control

DeePC uses Hankel matrix for receding-horizon prediction / estimation:

$$\begin{aligned} \underset{g,u,y}{\text{minimize}} \quad & \sum_{k=0}^{T_{\text{future}}-1} \|y_k - r_{t+k}\|_Q^2 + \|u_k\|_R^2 \\ \text{subject to} \quad & \mathscr{H}\begin{pmatrix} u^{\mathsf{d}} \\ y^{\mathsf{d}} \end{pmatrix} g = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \\ y \end{bmatrix}, \\ & u_k \in \mathcal{U}, \quad \forall k \in \{0, \ldots, T_{\text{future}} - 1\}, \\ & y_k \in \mathcal{Y}, \quad \forall k \in \{0, \ldots, T_{\text{future}} - 1\} \end{aligned}$$

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

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

hard operational or safety **constraints**

- trajectory matrix $\mathscr{H}\begin{pmatrix} u^{\mathsf{d}} \\ y^{\mathsf{d}} \end{pmatrix} = \begin{bmatrix} \mathscr{H}_{T_{\text{ini}}}\begin{pmatrix} u^{\mathsf{d}} \\ y^{\mathsf{d}} \end{pmatrix} \\ \mathscr{H}_{T_{\text{future}}}\begin{pmatrix} u^{\mathsf{d}} \\ y^{\mathsf{d}} \end{pmatrix} \end{bmatrix}$ from past data

collected **offline** (could be adapted online)

- past $T_{\text{ini}} \geq$ lag samples $(u_{\text{ini}}, y_{\text{ini}})$ for x_{ini} estimation

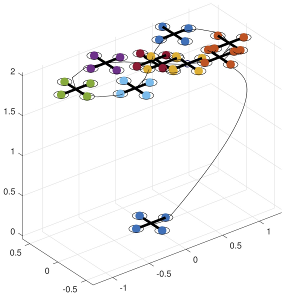
updated **online**

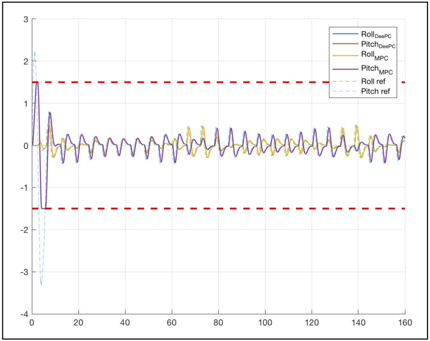
Consistency for LTI Systems

Theorem: Consider ***DeePC & MPC optimization problems***. If the rank condition holds (= rich data), then ***the feasible sets coincide***.

Corollary: closed-loop behavior under DeePC and MPC coincide.

Aerial robotics case study :





Thus, ***MPC carries over to DeePC***

…at least in the ***nominal case***.

(see e.g. [Berberich, Köhler, Müller, & Allgöwer '19] for stability proofs)

Beyond LTI, what about measurement noise, corrupted past data, and nonlinearities ?

Noisy real-time measurements

$$\underset{g,u,y}{\text{minimize}} \quad \sum_{k=0}^{T_{\text{future}}-1} \|y_k - r_{t+k}\|_Q^2 + \|u_k\|_R^2 + \lambda_y \|\sigma_{\text{ini}}\|_p$$

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

$$u_k \in \mathcal{U}, \quad \forall k \in \{0, \ldots, T_{\text{future}} - 1\},$$

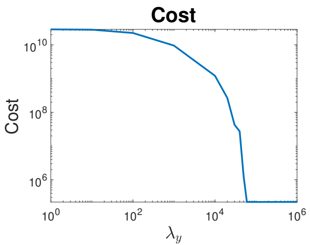
$$y_k \in \mathcal{Y}, \quad \forall k \in \{0, \ldots, T_{\text{future}} - 1\}$$

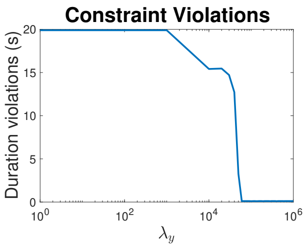
Solution: add ℓ_p-**slack** σ_{ini} to ensure feasibility

$\rightarrow$ receding-horizon least-square filter

$\rightarrow$ for $\lambda_y \gg 1$: constraint is slack only if infeasible

c.f. ***sensitivity analysis*** over randomized sims





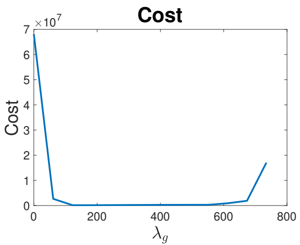
Trajectory matrix corrupted by noise

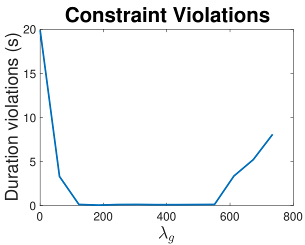
$$\begin{aligned} \underset{g,u,y}{\text{minimize}} \quad & \sum_{k=0}^{T_{\text{future}}-1} \|y_k - r_{t+k}\|_Q^2 + \|u_k\|_R^2 + \lambda_g \|g\|_1 \\ \text{subject to} \quad & \mathscr{H}\begin{pmatrix} u^{\text{d}} \\ y^{\text{d}} \end{pmatrix} g = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \\ y \end{bmatrix}, \\ & u_k \in \mathcal{U}, \quad \forall k \in \{0, \ldots, T_{\text{future}} - 1\}, \\ & y_k \in \mathcal{Y}, \quad \forall k \in \{0, \ldots, T_{\text{future}} - 1\} \end{aligned}$$

Solution: add a ℓ_1**-penalty** on g

intuition: ℓ_1 sparsely selects {trajectory matrix columns} = {past trajectories} = {motion primitives}

c.f. ***sensitivity analysis*** over randomized sims



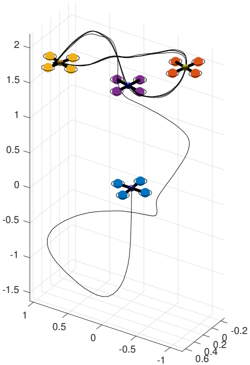


Towards nonlinear systems

Idea : lift nonlinear system to large/∞-dimensional bi-/linear system
$\rightarrow$ Carleman, Volterra, Fliess, Koopman, Sturm-Liouville methods
$\rightarrow$ nonlinear dynamics can be approximated LTI on finite horizons

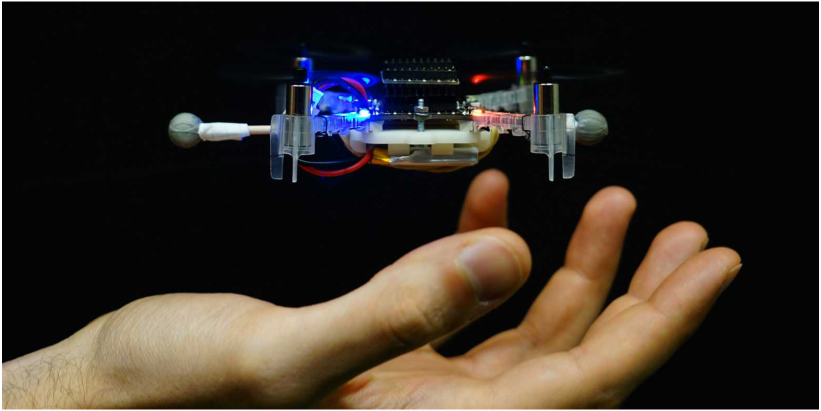
regularization singles out relevant features / basis functions in data

case study :
DeePC
+ σ_{ini} slack
+ $\|g\|_1$ regularizer
+ more columns in $\mathscr{H}\begin{pmatrix} u^{\text{d}} \\ y^{\text{d}} \end{pmatrix}$

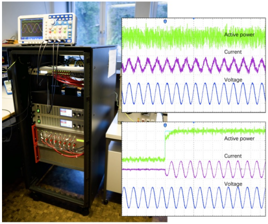


fluke
or
solid ?

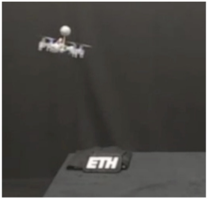
Experimental snippet



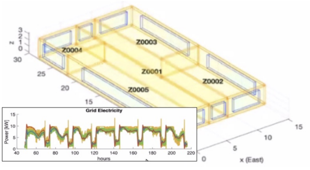
Consistent observations across case studies — more than a fluke



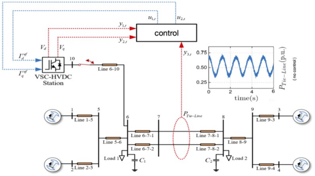
grid-connected converter



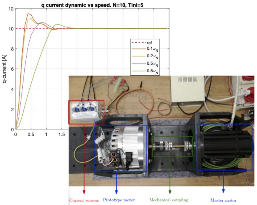
quad coptor fig-8 tracking



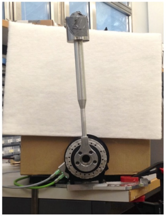
energy hub & building automation



power system oscillation damping (see later)



synchronous motor drive



pendulum swing up

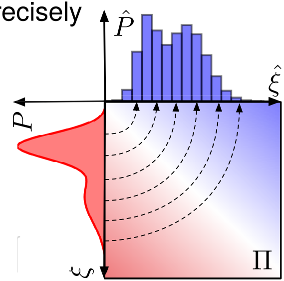
let’s try to put some theory behind all of this . . .

Distributional robust formulation [Coulson et al. '19]

- ***problem abstraction***: $\min_{x\in\mathcal{X}} \; c\left(\widehat{\xi}, x\right)$ where $\widehat{\xi}$ is *measured* data

- ***distributionally robust*** formulation $\longrightarrow$ "$\min_{x\in\mathcal{X}} \max_{\xi} \; c\left(\xi, x\right)$"
  where $\max$ accounts for all stochastic processes (linear or nonlinear) that could have generated the data ... more precisely

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



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

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

Theorem: Under minor technical conditions:

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

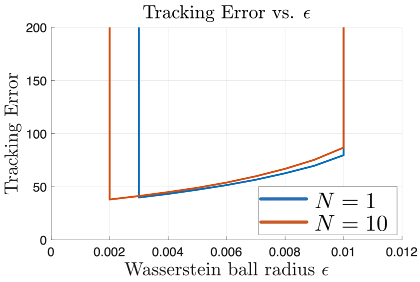
regularization of DeePC

$\Leftrightarrow$

distributional robustification in trajectory space

Further ingredients & improvements

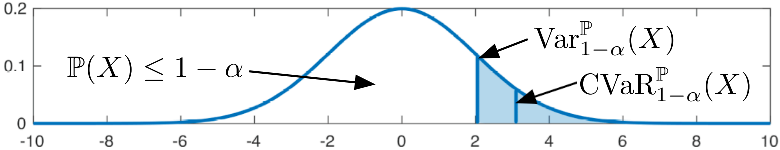
- multiple i.i.d. experiments $\rightarrow$ sample ***average data matrix*** $\frac{1}{N}\sum_{i=1}^{n}\mathscr{H}_i(y^{\mathsf{d}})$
- ***measure concentration***: Wasserstein ball $\mathbb{B}_\epsilon(\widehat{P})$ includes true distribution $\mathbb{P}$ with high confidence if $\epsilon \sim 1/N^{1/\dim(\xi)}$



- old online measurements $\rightarrow$ ***Kalman filtering*** with hidden state $=$ explicit $g^\star$

- ***distributionally robust probabilistic constraints***

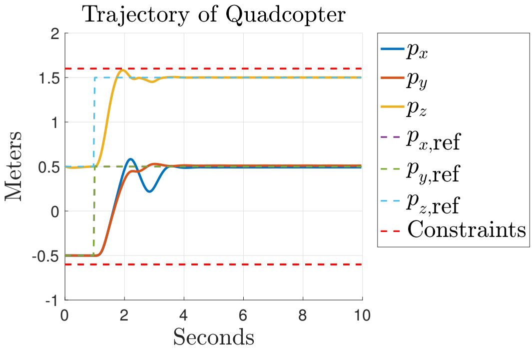
$\sup_{Q\in\mathbb{B}_\epsilon(\widehat{P})} \text{CVaR}_{1-\alpha}^{\mathbb{Q}} \quad \Leftrightarrow \quad$ averaging + regularization + tightening



All together in action for nonlinear & stochastic quadcoptor setup

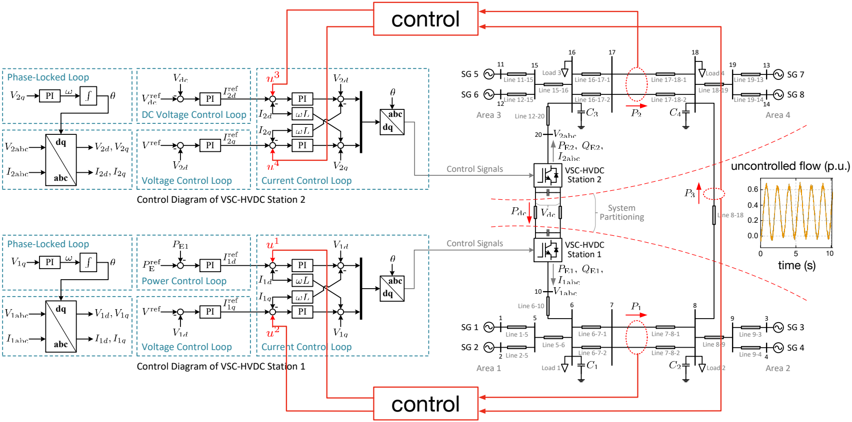
case study :
distr. robust objective
+ Page matrix predictor
+ averaging
+ CVaR constraints
+ σ_{ini} slack

→ DeePC works much better than it should !



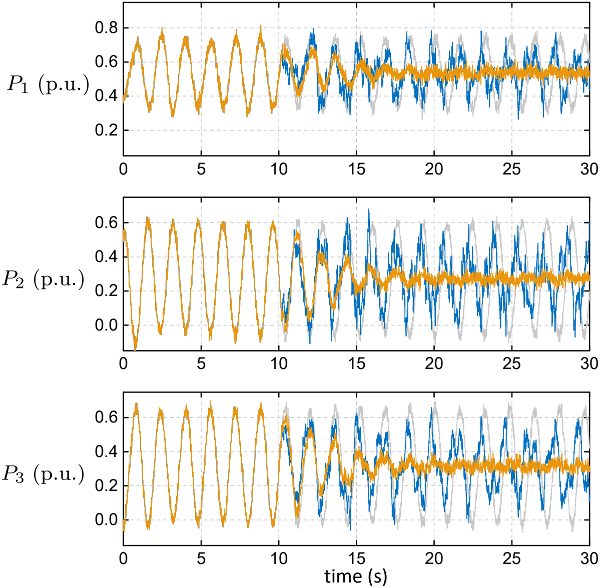
main catch : optimization problems become large (no-free-lunch)
→ models are compressed, de-noised, & tidied-up representations

Power system case study



- ***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 $\rightarrow$ choose T, T_{ini}, & T_{future} wisely

Centralized control



DeePC

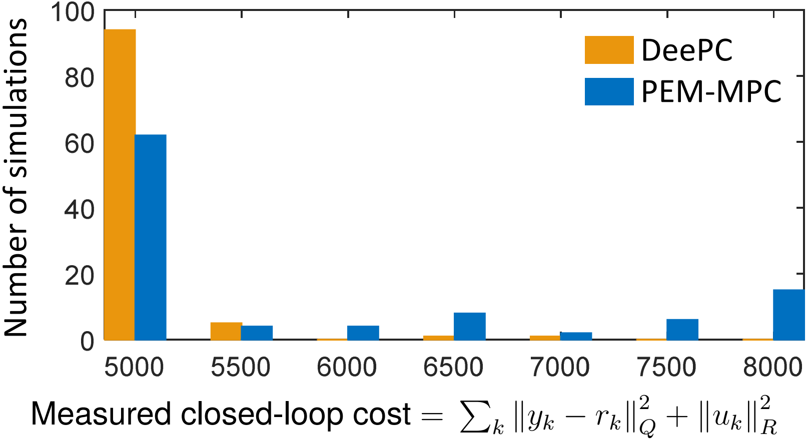
PEM-MPC

= Prediction Error Method (PEM) System ID + MPC

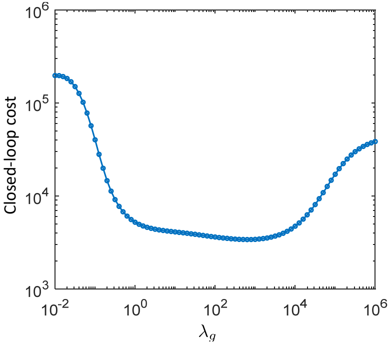
$t < 10\,$s : open loop data collection with white noise excitat.

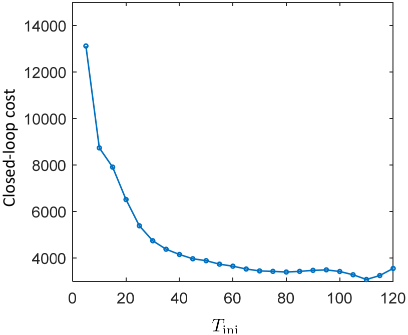
$t > 10\,$s : control

Performance: DeePC wins (clearly!)



DeePC hyper-parameter tuning





***regularizer* λ_g**

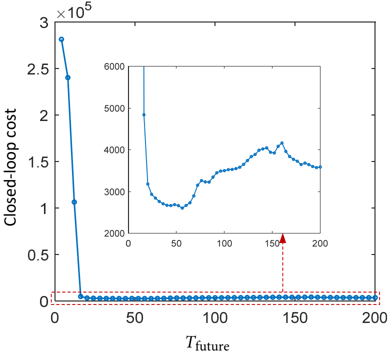
- for distributional robustness $\approx$ radius of Wasserstein ball
- wide range of sweet spots

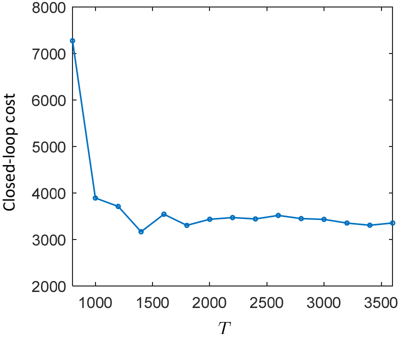
$\rightarrow$ choose $\lambda_g = 20$

***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

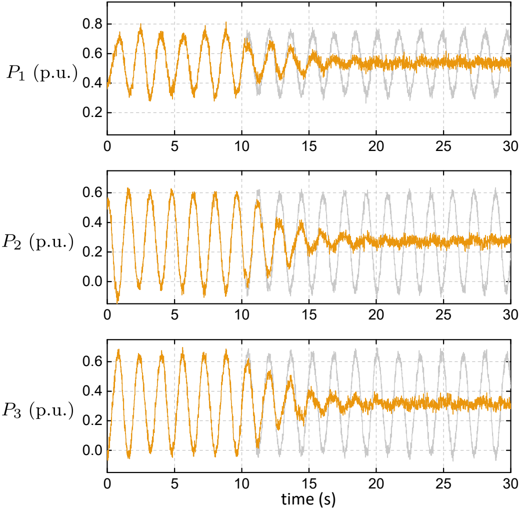
  $\rightarrow$ choose $T_{\text{future}} = 120$ and apply first 60 input steps

data length T

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

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

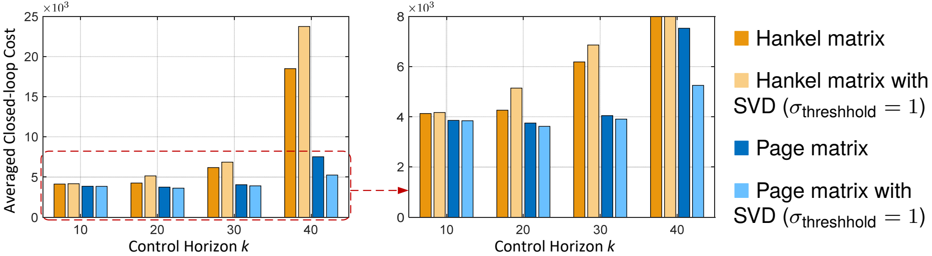
Computational cost



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

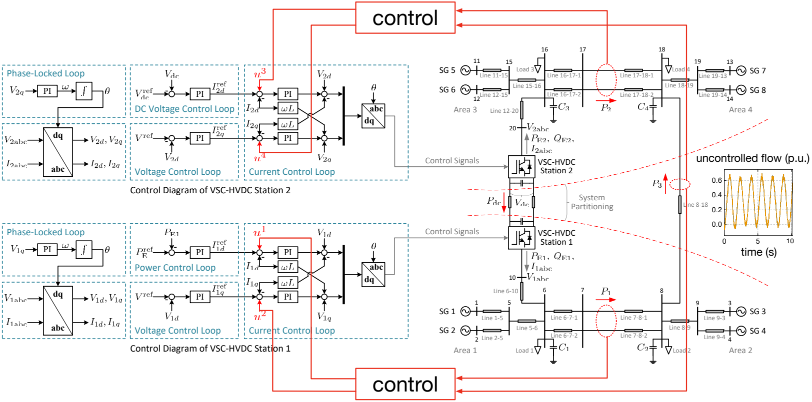
$\Rightarrow$ **implementable**

Comparison: Hankel & Page matrix



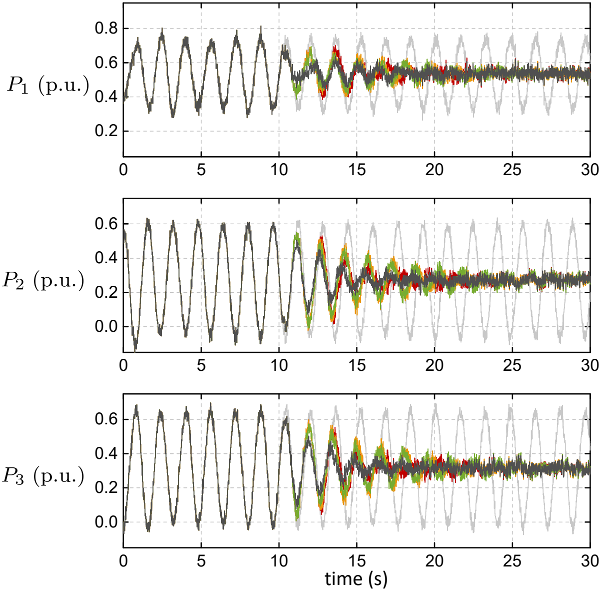
- comparison baseline: Hankel and Page matrices of ***same size***
- ***perfomance*** : Page consistency beats Hankel matrix predictors
- offline ***denoising via SVD threshholding*** works wonderfully for Page though obviously not for Hankel (entries are constrained)
- effects very pronounced for ***longer 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**

Summary & conclusions

main take-aways

- matrix time series serves as predictive model
- data-enabled predictive control (DeePC)

✓ consistent for deterministic LTI systems

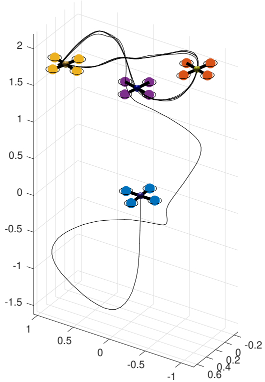
✓ distributional robustness via regularizations

future work

→ tighter certificates for nonlinear systems

→ explicit policies & direct adaptive control

→ online optimization & real-time iteration



Why have these powerful ideas not been mixed long before ?

Willems '07: *"[MPC] has perhaps too little system theory and too much brute force computation in it."*

The other side often proclaims *"behavioral systems theory is beautiful but did not prove utterly useful."*

Thanks !

Florian Dörfler

mail: dorfler@ethz.ch

[link] to homepage

[link] to related publications

appendix

relation to system ID

Data-driven control: a classification

indirect data-driven control

minimize	control cost (x, u)
subject to	(x, u) satisfy state-space model
where	x estimated from (u, y) & model
where	model identified from (u^d, y^d) data

} outer optimization
} middle opt.
} inner opt.

} separation & certainty equivalence ($\rightarrow$ LQG case)

} **no** separation ($\rightarrow$ ID-4-control)

$\rightarrow$ nested multi-level optimization problem

direct data-driven control

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

$\rightarrow$ ***trade-offs***

modular vs. end-2-end
suboptimal (?) vs. optimal
convex vs. non-convex (?)

Additionally: all above should be min-max or $\mathbb{E}(\cdot)$ accounting for ***uncertainty*** …

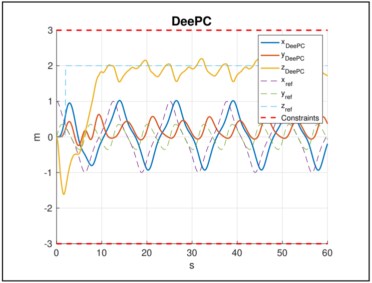
recall the ***central promise*** :

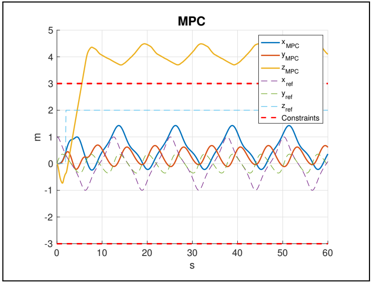
it is easier to learn control policies directly from data, rather than learning a model

Comparison: DeePC vs. ID + MPC

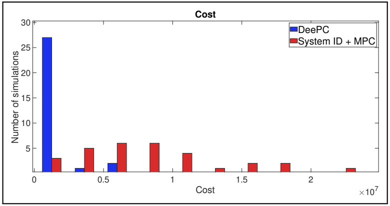
DeePC with ℓ_1-regularizer

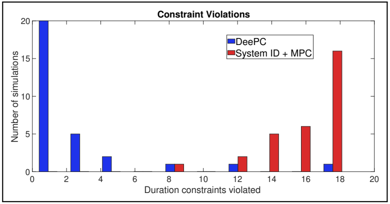
certainty-equivalence ***MPC*** based on prediction error ***ID***





single fig-8 run



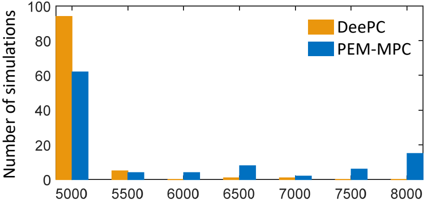


random sims

More to it than a single case study ?

consistent across all nonlinear case studies : DeePC always wins

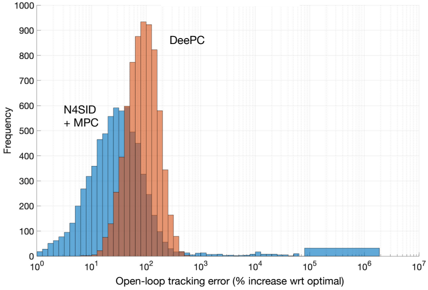
reason (?) : DeePC is robust, whereas certainty-equivalence control is based on identified model with a bias error



measured closed-loop cost $= \sum_k \|y_k - r_k\|_Q^2 + \|u_k\|_R^2$

stochastic LTI comparison (no bias) show certainty-equivalence vs. robust control trade-offs (mean vs. median)

link : DeePC includes implicit sys ID though biased by control objective & robustified through regularizations



$\rightarrow$ lot more to be understood . . .