In the Shallows of the DeePC : Data-Enabled Predictive Control

Florian Dörfler

Automatic Control Laboratory, ETH Zürich

Acknowledgements

John Lygeros

Jeremy Coulson

Funding: ETH Zürich

Simulation data: M. Zeilinger and C. Jones

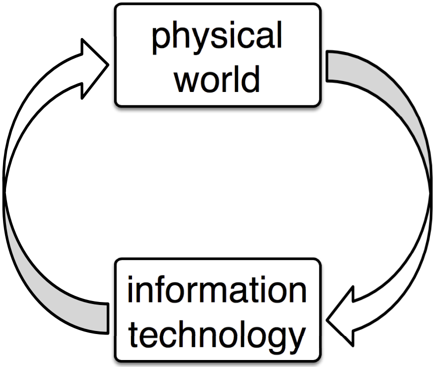
Brainstorming: B. Bamieh, B. Recht, A. Cherukuri, and M. Morari

Feedback – our central paradigm

actuation

“making a difference to the world”

automation and control



sensing

“making sense of the world”

inference and data science

Big, deep, data, and so on

- ***unprecedented availability*** of computation, storage, and data
- ***theoretical advances*** in optimization, statistics, and 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

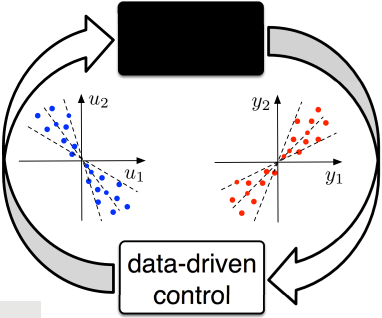
Thomas B. Schön THOMAS.SCHON@IT.UU.SE
Department of Information Technology, Uppsala University, Sweden

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

Control in a data-rich world

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

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



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

- models are too complex to be useful (e.g., control of fluid dynamics)
- first-principle models are not conceivable (e.g., human-in-the-loop applications)
- modeling and system ID is too costly (e.g., non-critical robotics applications)

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

Example: PID

…of course, we are all tempted, annoyed, …

machine learning often achieves super-human performance, but it performs nowhere near MPC

…but that's an entirely unfair comparison, is it?

today: *preliminary* ideas on a new approach that seems equally simple & powerful

Snippets from the literature

1. ***reinforcement learning*** / or
   stochastic adaptive control / or
   approximate dynamic programming

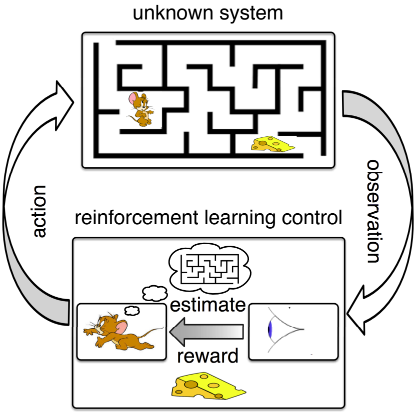
with key ***mathematical challenges***

- (approximate/neuro) **DP** to learn approx. value/Q-function or optimal policy
- (stochastic) **function approximation** in continuous state and action spaces
- **exploration-exploitation** trade-offs

and ***practical limitations***

- **inefficiency**: computation & samples
- **complex and fragile** algorithms
- **safe real-time** exploration

Ø suitable for physical control systems?



A Tour of Reinforcement Learning
The View from Continuous Control

Benjamin Recht
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley

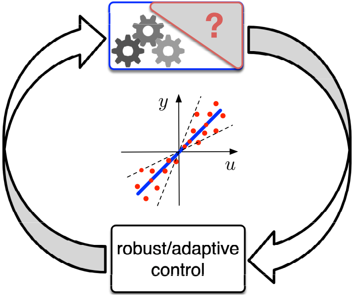
Snippets from the literature cont'd

2. gray-box ***safe learning & control***

- ***robust*** $\rightarrow$ conservative & complex control
- ***adaptive*** $\rightarrow$ hard & asymptotic performance
- ***contemporary learning*** algorithms
  (e.g., MPC + Gaussian processes / RL)

$\rightarrow$ non-conservative, optimal, & safe

ø limited applicability: need a-priori safety



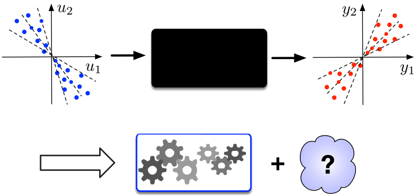
3. Sequential ***system ID + control***

- ID with uncertainty quantification followed by robust control design

$\rightarrow$ recent finite-sample & end-to-end ID + control pipelines out-performing RL

ø ID seeks best but not most useful model

ø "easier to learn policies than models"



Key take-aways

Quintessence of literature review:

- data-driven approach is ***no silver bullet*** (see previous ø), and we did not even discuss output feedback, safety constraints, ...
- ***predictive models are preferable over data*** (even approximate)

$\rightarrow$ models are tidied-up, compressed, and de-noised representations

$\rightarrow$ model-based methods vastly out-perform model-agnostic strategies

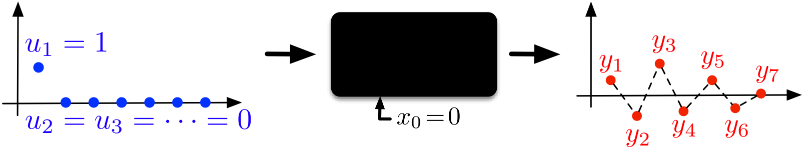
- but often ***easier to learn controllers*** from data rather than models

ø deadlock ?

- a useful ML insight: ***non-parametric methods*** are often preferable over parametric ones (e.g., basis functions vs. kernels)

$\rightarrow$ build a predictive & non-parametric model directly from raw data ?

Colorful idea



If you had the ***impulse response*** of a LTI system, then …

- can build state-space ***system identification*** (Kalman-Ho realization)
- …but can also build ***predictive model directly from raw data*** :

$$y_{\text{future}}(t) = \begin{bmatrix} y_1 & y_2 & y_3 & \cdots \end{bmatrix} \cdot \begin{bmatrix} u_{\text{future}}(t) \\ u_{\text{future}}(t-1) \\ u_{\text{future}}(t-2) \\ \vdots \end{bmatrix}$$

- ***model predictive control*** from data: dynamic matrix control (DMC)
- ***today:*** can we do so with arbitrary, finite, and corrupted I/O samples ?

Contents

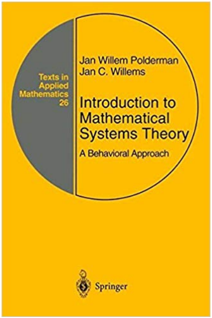
Behavioral view on LTI systems

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.

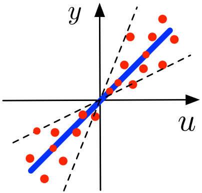


Definition: The dynamical system $(\mathbb{Z}_{\geq 0}, \mathbb{W}, \mathscr{B})$ is

(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 $\sigma w_t = w_{t+1}$, and

(iii) ***complete*** if $\mathscr{B}$ is closed $\Leftrightarrow$ $\mathbb{W}$ is finite dimensional.

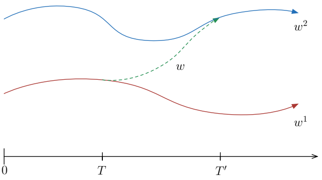


In the remainder we focus on ***discrete-time LTI systems***.

Behavioral view cont'd

Behavior $\mathscr{B}$ = set of trajectories in $\mathbb{W}^{\mathbb{Z}_{\geq 0}}$, and set of ***truncated trajectories*** $\mathscr{B}_T = \{w \in \mathbb{W}^T \mid \exists\, v \in \mathscr{B}$ s.t. $w_t = v_t,\ t \in [0, T]\}$

A system $(\mathbb{Z}_{\geq 0}, \mathbb{W}, \mathscr{B})$ is ***controllable*** if any two truncated trajectories w^1, w^2 $\in \mathscr{B}$ can be patched together in finite time with a trajectory $w \in \mathscr{B}_{[T,T']}$.



I/O: $\mathscr{B} = \mathscr{B}^u \times \mathscr{B}^y$ where $\mathscr{B}^u = (\mathbb{R}^m)^{\mathbb{Z}_{\geq 0}}$ and $\mathscr{B}^y \subseteq (\mathbb{R}^p)^{\mathbb{Z}_{\geq 0}}$ are the spaces of ***input and output*** signals $\Rightarrow w = \text{col}(u, y) \in \mathscr{B}$

parametric ***kernel representation***: $\mathscr{B} = \text{col}(u, y) \in (\mathbb{R}^{m+p})^{\mathbb{Z}_{\geq 0}}$ s.t.

$$b_0 u + b_1 \sigma u + \cdots + b_n \sigma^n u \ + \ a_0 y + a_1 \sigma y + \ldots a_n \sigma^n y \ = \ 0$$

$$\Leftrightarrow \boxed{\text{col}(u, y) \in \ker \left[b_0\ b_1\sigma\ \ldots\ b_n\sigma^n \quad a_0\ a_1\sigma\ \ldots\ a_n\sigma^n \right]}$$

Behavioral view cont'd

- parametric ***state-space representation*** with minimal realization

$$\mathscr{B}(A,B,C,D) = \left\{\text{col}(u,y) \in (\mathbb{R}^{m+p})^{\mathbb{Z}_{\geq 0}} \mid \exists\, x \in (\mathbb{R}^n)^{\mathbb{Z}_{\geq 0}} \right.$$
$$\left. \text{s.t. } \sigma x = Ax + Bu,\ y = Cx + Du \right\}$$

- ***lag*** smallest $\ell \in \mathbb{Z}_{>0}$ s.t. observability matrix $\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{\ell-1} \end{bmatrix}$ has rank n

Lemma [Markovsky & Rapisarda '08]: Consider a minimal state-space model $\mathscr{B}(A,B,C,D)$ & a trajectory $\text{col}(u_{\text{ini}}, u\,, y_{\text{ini}}, y) \in \mathscr{B}_{T_{\text{ini}}+T_{\text{future}}}$ of length $T_{\text{ini}} + T_{\text{future}}$ with $T_{\text{ini}} \geq \ell$. Then $\exists$ unique $x_{\text{ini}} \in \mathbb{R}^n$ such that

$$y = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{\ell-1} \end{bmatrix} x_{\text{ini}} + \begin{bmatrix} D & 0 & \cdots & 0 \\ CB & D & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ CA^{N-2}B & \cdots & CB & D \end{bmatrix} u\,.$$

i.e., we can ***recover the initial condition*** *from past ℓ samples.*

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***

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

(kernel representation)

$\Rightarrow$

$\Leftarrow$ under assumptions

$\begin{bmatrix} b_0 & a_0 & b_1 & a_1 & \ldots & b_n & a_n \end{bmatrix}$ is in the left nullspace of the ***Hankel matrix***

$$\mathscr{H}_t \begin{pmatrix} u \\ y \end{pmatrix} = \begin{bmatrix} \begin{pmatrix} u_1 \\ y_1 \end{pmatrix} & \begin{pmatrix} u_2 \\ y_2 \end{pmatrix} & \begin{pmatrix} u_3 \\ y_3 \end{pmatrix} & \cdots & \begin{pmatrix} u_{T-L+1} \\ y_{T-L+1} \end{pmatrix} \\ \begin{pmatrix} u_2 \\ y_2 \end{pmatrix} & \begin{pmatrix} u_3 \\ y_3 \end{pmatrix} & \begin{pmatrix} u_4 \\ y_4 \end{pmatrix} & \cdots & \vdots \\ \begin{pmatrix} u_3 \\ y_3 \end{pmatrix} & \begin{pmatrix} u_4 \\ y_4 \end{pmatrix} & \begin{pmatrix} u_5 \\ y_5 \end{pmatrix} & \cdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \begin{pmatrix} u_L \\ y_L \end{pmatrix} & \cdots & \cdots & \cdots & \begin{pmatrix} u_T \\ y_T \end{pmatrix} \end{bmatrix}$$

(collected from data $\in \{1, \ldots, T\}$)

The Fundamental Lemma

Definition : The signal $u = \text{col}(u_1, \ldots, u_T) \in \mathbb{R}^{Tm}$ is ***persistently exciting of order L if*** $\mathscr{H}_L(u) = \begin{bmatrix} u_1 & \cdots & u_{T-L+1} \\ \vdots & \ddots & \vdots \\ u_L & \cdots & u_T \end{bmatrix}$ is of full row rank,

*i.e., if the signal is **sufficiently rich** and **long** $(T - L + 1 \geq mL)$.*

Fundamental lemma [Willems et al, '05] : Let $T, t \in \mathbb{Z}_{>0}$, Consider

- a *controllable* LTI system $(\mathbb{Z}_{>0}, \mathbb{R}^{m+p}, \mathscr{B})$, and
- a T-sample long *trajectory* $\text{col}(u, y) \in \mathscr{B}_T$, where
- u is *persistently exciting* of order $t + n$. Then

$$\boxed{\text{colspan}\left(\mathscr{H}_t\left(\begin{smallmatrix} u \\ y \end{smallmatrix}\right)\right) = \mathscr{B}_t}\ .$$

Cartoon of Fundamental Lemma

persistently exciting | controllable LTI | sufficiently many samples

$$\mathscr{B}_t \equiv \text{colspan}\left(\begin{bmatrix} \begin{pmatrix} u_1 \\ y_1 \end{pmatrix} & \begin{pmatrix} u_2 \\ y_2 \end{pmatrix} & \begin{pmatrix} u_3 \\ y_3 \end{pmatrix} & \cdots & \begin{pmatrix} u_{T-t+1} \\ y_{T-t+1} \end{pmatrix} \\ \begin{pmatrix} u_2 \\ y_2 \end{pmatrix} & \begin{pmatrix} u_3 \\ y_3 \end{pmatrix} & \begin{pmatrix} u_4 \\ y_4 \end{pmatrix} & \cdots & \vdots \\ \begin{pmatrix} u_3 \\ y_3 \end{pmatrix} & \begin{pmatrix} u_4 \\ y_4 \end{pmatrix} & \begin{pmatrix} u_5 \\ y_5 \end{pmatrix} & \cdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \begin{pmatrix} u_t \\ y_t \end{pmatrix} & \cdots & \cdots & \cdots & \begin{pmatrix} u_T \\ y_T \end{pmatrix} \end{bmatrix} \right)$$

all trajectories constructible from finitely many previous trajectories

Consequences

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \iff \operatorname{colspan} \begin{bmatrix} \begin{pmatrix} u_1 \\ y_1 \end{pmatrix} & \begin{pmatrix} u_2 \\ y_2 \end{pmatrix} & \begin{pmatrix} u_3 \\ y_3 \end{pmatrix} & \cdots \\ \begin{pmatrix} u_2 \\ y_2 \end{pmatrix} & \begin{pmatrix} u_3 \\ y_3 \end{pmatrix} & \begin{pmatrix} u_4 \\ y_4 \end{pmatrix} & \cdots \\ \begin{pmatrix} u_3 \\ y_3 \end{pmatrix} & \begin{pmatrix} u_4 \\ y_4 \end{pmatrix} & \begin{pmatrix} u_5 \\ y_5 \end{pmatrix} & \cdots \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix}$$

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

Now let us draw the dramatic corollaries . . .

Data-driven simulation [Markovsky & Rapisarda '08]

Problem: ***predict*** future output $y_{\text{future}} \in \mathbb{R}^{pT_{\text{future}}}$ based on

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

Solution: ***Assume*** that $\mathscr{B}$ is controllable and u_{data} is persistently exciting of oder $T_{\text{ini}} + T_{\text{future}} + n$. Form partitioned ***Hankel matrices***

$$\begin{bmatrix} U_{\mathrm{p}} \\ U_{\mathrm{f}} \end{bmatrix} = \mathscr{H}_{T_{\text{ini}}+T_{\text{future}}}(u_{\text{data}}) \quad \text{and} \quad \begin{bmatrix} Y_{\mathrm{p}} \\ Y_{\mathrm{f}} \end{bmatrix} = \mathscr{H}_{T_{\text{ini}}+T_{\text{future}}}(y_{\text{data}}) \,.$$

Solve predictive model for (g, y_{future}):

$$\begin{bmatrix} U_{\mathrm{p}} \\ Y_{\mathrm{p}} \\ U_{\mathrm{f}} \\ Y_{\mathrm{f}} \end{bmatrix} g = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u_{\text{future}} \\ y_{\text{future}} \end{bmatrix} \begin{matrix} \left.\vphantom{\begin{matrix} a \\ b \end{matrix}}\right\} \text{recover } x_{\text{ini}} \\ \left.\vphantom{\begin{matrix} a \\ b \end{matrix}}\right\} \text{prediction} \end{matrix}$$

Markovsky et al. similarly address feedforward control problem

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

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

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

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

model for **prediction** over $k \in [0, T-1]$

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 non-parametric and data-based Hankel matrix time series as prediction/estimation model inside MPC optimization problem:

$$\begin{aligned}
\underset{g,u,y}{\text{minimize}} \quad & \sum_{k=0}^{T-1} \|y_k - r_{t+k}\|_Q^2 + \|u_k\|_R^2 \\
\text{subject to} \quad & \begin{bmatrix} U_\mathrm{p} \\ Y_\mathrm{p} \\ U_\mathrm{f} \\ Y_\mathrm{f} \end{bmatrix} g = \begin{bmatrix} u_\text{ini} \\ y_\text{ini} \\ u \\ y \end{bmatrix}, \\
& u_k \in \mathcal{U}, \quad \forall k \in \{0, \ldots, T-1\}, \\
& y_k \in \mathcal{Y}, \quad \forall k \in \{0, \ldots, T-1\}
\end{aligned}$$

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

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

hard operational or safety **constraints**

- Hankel matrix with $T_\text{ini} + T$ rows from past data $\begin{bmatrix} U_\mathrm{p} \\ U_\mathrm{f} \end{bmatrix} = \mathscr{H}_{T_\text{ini}+T}(u_\text{data})$ and $\begin{bmatrix} Y_\mathrm{p} \\ Y_\mathrm{f} \end{bmatrix} = \mathscr{H}_{T_\text{ini}+T}(y_\text{data})$

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

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

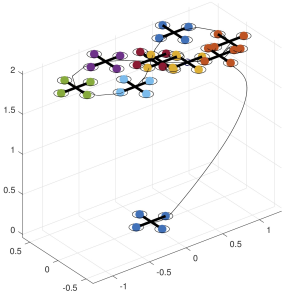
updated **online**

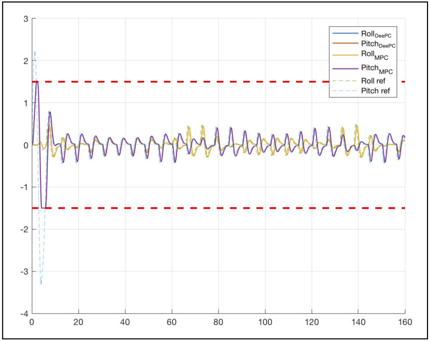
Correctness for LTI Systems

Theorem: Consider a ***controllable LTI system*** and the DeePC & MPC optimization problems with ***persistently exciting*** data of order $T_{\text{ini}} + T + n$. Then the ***feasible sets of DeePC & MPC coincide***.

Corollary: If $\mathcal{U}, \mathcal{Y}$ are ***convex***, then also the ***trajectories coincide***.

Aerial robotics case study :





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

…at least in the ***nominal case***.

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

Noisy real-time measurements

$$\underset{g,u,y}{\text{minimize}} \quad \sum_{k=0}^{T-1} \|y_k - r_{t+k}\|_Q^2 + \|u_k\|_R^2 + \lambda_y \|\sigma_y\|_1$$

$$\text{subject to} \quad \begin{bmatrix} U_\text{p} \\ Y_\text{p} \\ U_\text{f} \\ Y_\text{f} \end{bmatrix} g = \begin{bmatrix} u_\text{ini} \\ y_\text{ini} \\ u \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ \sigma_y \\ 0 \\ 0 \end{bmatrix},$$

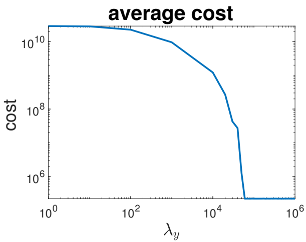
$$u_k \in \mathcal{U}, \quad \forall k \in \{0, \ldots, T-1\},$$

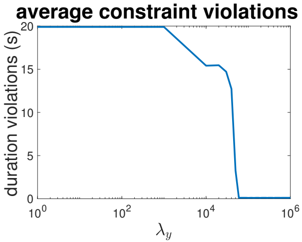
$$y_k \in \mathcal{Y}, \quad \forall k \in \{0, \ldots, T-1\}$$

Solution : add **slack** to ensure feasibility with **ℓ_1-penalty**

$\Rightarrow$ for λ_y sufficiently large $\sigma_y \neq 0$ only if constraint infeasible

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





Hankel matrix corrupted by noise

$$\underset{g,u,y}{\text{minimize}} \quad \sum_{k=0}^{T-1} \|y_k - r_{t+k}\|_Q^2 + \|u_k\|_R^2 + \lambda_g \|g\|_1$$

$$\text{subject to} \quad \begin{bmatrix} U_\mathrm{p} \\ Y_\mathrm{p} \\ U_\mathrm{f} \\ Y_\mathrm{f} \end{bmatrix} g = \begin{bmatrix} u_\mathsf{ini} \\ y_\mathsf{ini} \\ u \\ y \end{bmatrix},$$

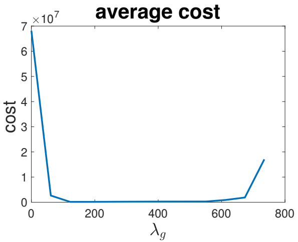
$$u_k \in \mathcal{U}, \quad \forall k \in \{0, \ldots, T-1\},$$

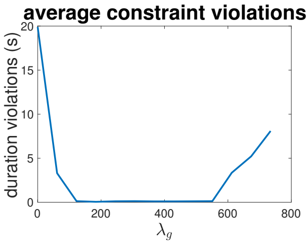
$$y_k \in \mathcal{Y}, \quad \forall k \in \{0, \ldots, T-1\}$$

Solution : add a ℓ_1-**penalty** on g

another solution **:**
low-rank approximation of $\mathscr{H}\begin{pmatrix} u_\mathsf{data} \\ y_\mathsf{data} \end{pmatrix}$ seems to perform much less well

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





Why an ℓ_1-penalty on g ?

$$\begin{bmatrix} \begin{pmatrix} u_1 \\ y_1 \end{pmatrix} & \begin{pmatrix} u_2 \\ y_2 \end{pmatrix} & \begin{pmatrix} u_3 \\ y_3 \end{pmatrix} & \cdots \\ \begin{pmatrix} u_2 \\ y_2 \end{pmatrix} & \begin{pmatrix} u_3 \\ y_3 \end{pmatrix} & \begin{pmatrix} u_4 \\ y_4 \end{pmatrix} & \cdots \\ \begin{pmatrix} u_3 \\ y_3 \end{pmatrix} & \begin{pmatrix} u_4 \\ y_4 \end{pmatrix} & \begin{pmatrix} u_5 \\ y_5 \end{pmatrix} & \cdots \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix} g = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \\ y \end{bmatrix}$$

- ***intuition*** : each column of Hankel matrix $\equiv$ a past trajectory

$\rightarrow$ ℓ_1 induces sparse column selection $\equiv$ ***motion primitive*** combination

- why not ℓ_2-average over columns? $\rightarrow$ ***scenario-based programming*** reasoning : sparse set of support constraints picket out by ℓ_1-penalty

- ***distributional robustness*** reasoning : ℓ_1-penalty $\equiv$ ℓ_∞-robustness

$\rightarrow$ $\min\limits_{x} \max\limits_{\mathbb{P} \in \{\|\mathbb{P} - \mathbb{P}_{\text{sample}}\|_{\infty,\,\text{Wasserstein}} \leq \rho\}} \mathbb{E}^{\mathbb{P}}[f(x)] \equiv \min\limits_{x} \mathbb{E}^{\mathbb{P}_{\text{sample}}}[f(x)] + \frac{1}{\rho}\|x\|_1$

- …***still working*** on providing exact proofs and quantitative guarantees

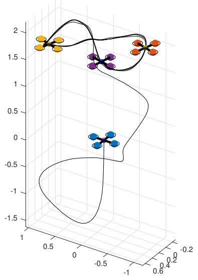
Towards nonlinear systems

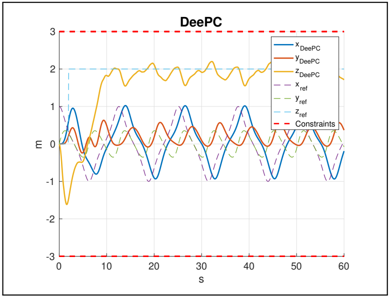
Idea : lift nonlinear system to large/∞-dimensional bi-/linear system
$\rightarrow$ Carleman, Volterra, Fliess, Koopman, Sturm-Liouville methods
$\rightarrow$ ***exploit size rather than nonlinearity*** and find features in data

$\rightarrow$ exploit size, collect more data, & build a ***larger Hankel matrix***

$\rightarrow$ ***low-rank approximation*** singles out relevant basis functions

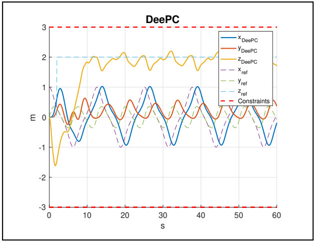
case study : low-rank approximation + regularization for g and σ_y



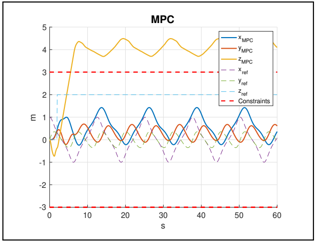


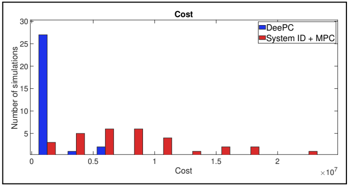
Comparison to system ID + MPC

Setup : nonlinear stochastic quadcopter model with full state info
DeePC : low-rank approximation + ℓ_1-regularization for g and σ_y
MPC : sys ID via prediction error method + nominal MPC

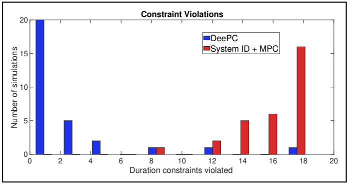


single fig-8 run





random sims



Summary and conclusions

- fundamental lemma from behavioral systems
- matrix time series serves as predictive model
- data-enabled predictive control (DeePC)

✓ certificates for deterministic LTI systems

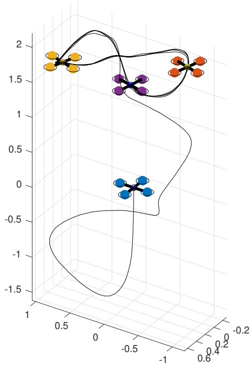
✓ robustification through salient regularizations

✓ DeePC works extremely well on case study

→ certificates for stochastic/nonlinear setup

→ adaptive extensions, explicit policies, . . .

→ other non-parametric data-based models



Why have these powerful ideas not been mixed long before us ?

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 useful”*