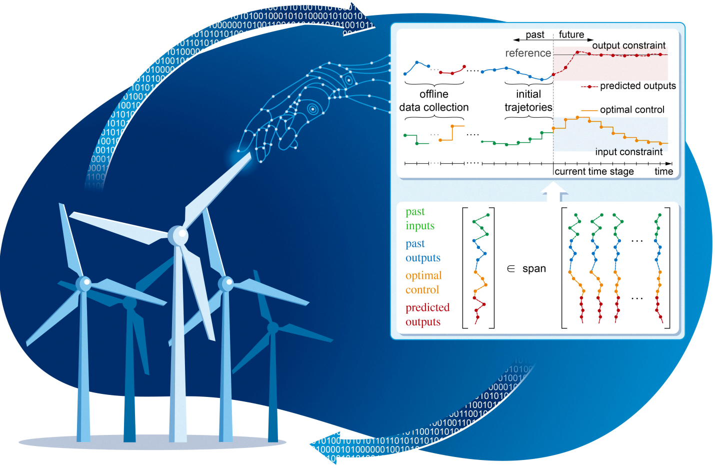
Data-Enabled Predictive Control of Autonomous Energy Systems



Florian Dörfler

ETH Zürich

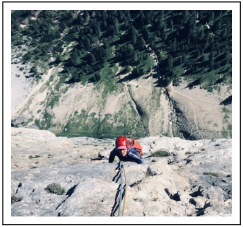
KIOS 2024 Graduate School

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

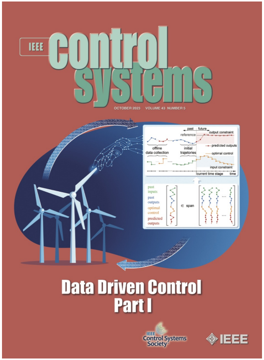
Thoughts on data in control systems

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

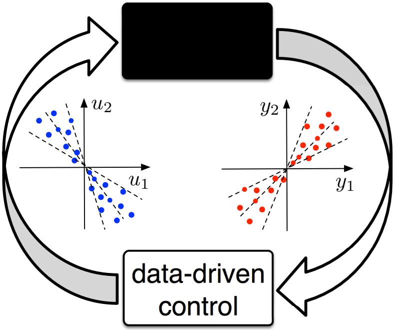
Make up your own opinion, but ML works too well to be ignored – also in control **?!?**

"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]



Approaches to 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)
subject to	(u, x) 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)

<u>no</u> separation ($\rightarrow$ ID-4-control)

$\rightarrow$ 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

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

modular vs. end-2-end

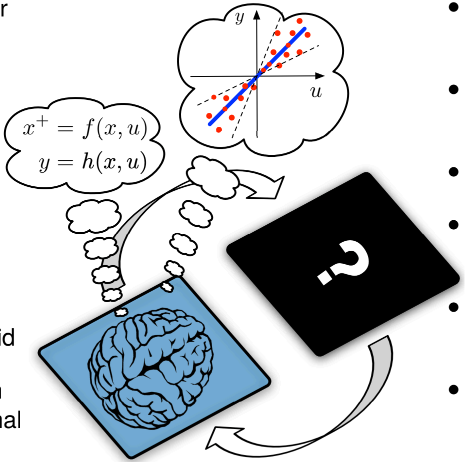
suboptimal (?) vs. optimal

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 $\to$ easy to debug & interpret
- id = noise filtering
- id = projection on model class
- harder to propagate uncertainty through id
- no robust separation principle $\to$ suboptimal
- …



- some models too complex to be useful
- end-to-end $\to$ 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
- …

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

A direct approach: dictionary + MPC

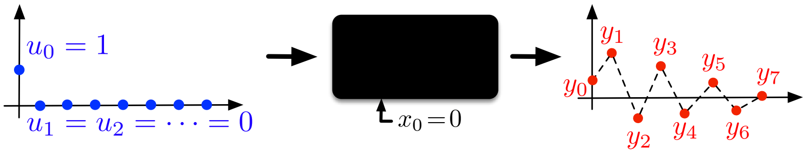
① trajectory ***dictionary learning***

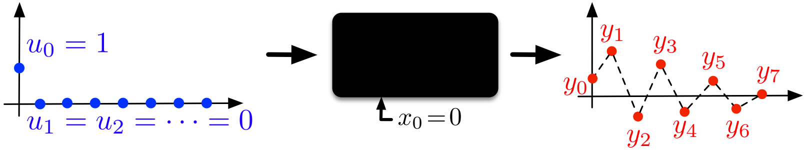
- motion primitives / basis functions
- theory: Koopman & Liouville
  practice: (E)DMD & particles

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

$\rightarrow$ huge ***theory vs. practice*** gap

$\rightarrow$ back to basics: ***impulse response***





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

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

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

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

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)

→ 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).

Organization of this lecture

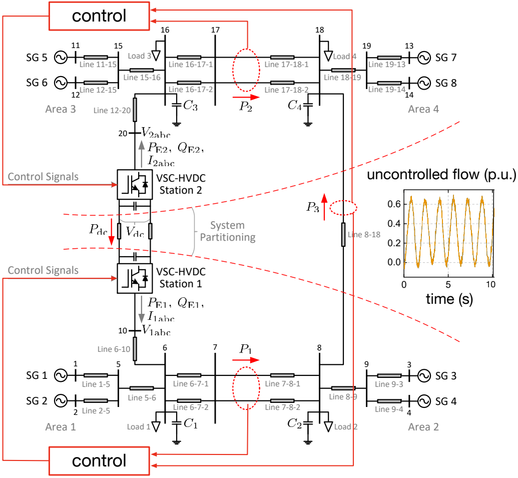
- I will ***teach the basics*** & provide pointers to more sophisticated research material $\longrightarrow$ study cutting-edge papers yourself
- it's a school: so we will spend time on the ***board*** $\longrightarrow$ take notes
- We teach this material also in the ETH Zürich bachelor & have plenty of ***background material*** + implementation experience $\longrightarrow$ please reach out to me or Saverio if you need anything
- we will take a ***break*** after 90 minutes $\longrightarrow$ coffee ☺

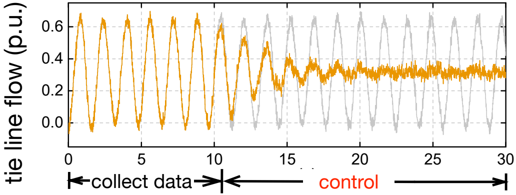
Preview

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

control objective: oscillation damping $\boxed{\text{without a model}}$

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





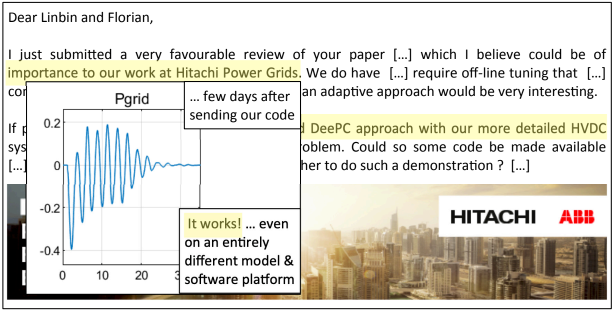
seek a method that ***works reliably***, can be ***efficiently*** implemented, & ***certifiable***

$\rightarrow$ 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

Behavioral view on dynamical systems

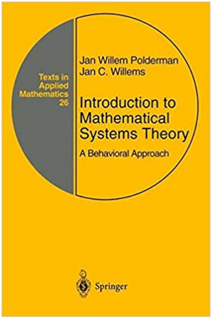
Definition: A discrete-time ***dynamical system*** is a 3-tuple $(\mathbb{Z}_{\geq 0}, \mathbb{W}, \mathscr{B})$ where

(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*.

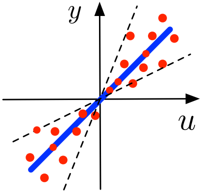
$\mathscr{B}$ is the set of all trajectories



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

LTI systems & matrix time series

foundation of subspace system identification & signal recovery algorithms

$(u(t), y(t))$ 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)

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

Fundamental Lemma

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

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

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

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

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***

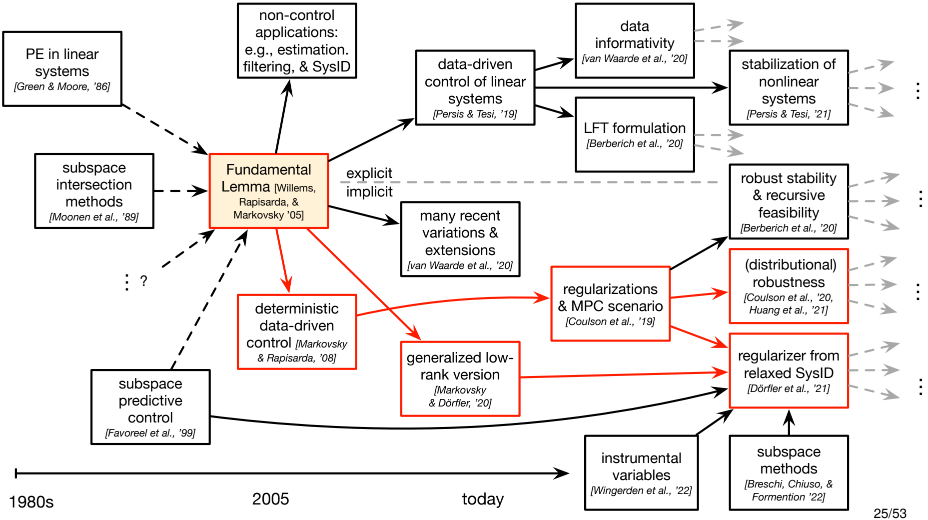
Input design for Fundamental Lemma

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.

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

Data matrix structures & preprocessing

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

Data-enabled Predictive Control (DeePC)

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

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

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)

$\rightarrow$ ***equivalent to MPC*** in deterministic LTI case …

but needs to be robustified in case of noise / nonlinearity !

Regularizations to make it work

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

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

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

measurement noise

$\rightarrow$ infeasible y_{ini} estimate

$\rightarrow$ estimation slack σ

$\rightarrow$ moving-horizon least-square filter

noisy or nonlinear (offline) **data matrix**

$\rightarrow$ any $\binom{u}{y}$ feasible

$\rightarrow$ add regularizer $h(g)$

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

Robustness intuition: regularization $\Leftrightarrow$ robustifies, e.g., in a simple case regularization

$$\Updownarrow$$

incorporating priors
+ implicit SysID

Regularization = relaxing low-rank approximation in pre-processing

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

optimal control

low-rank approximation

↓ sequence of convex relaxations ↓

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

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