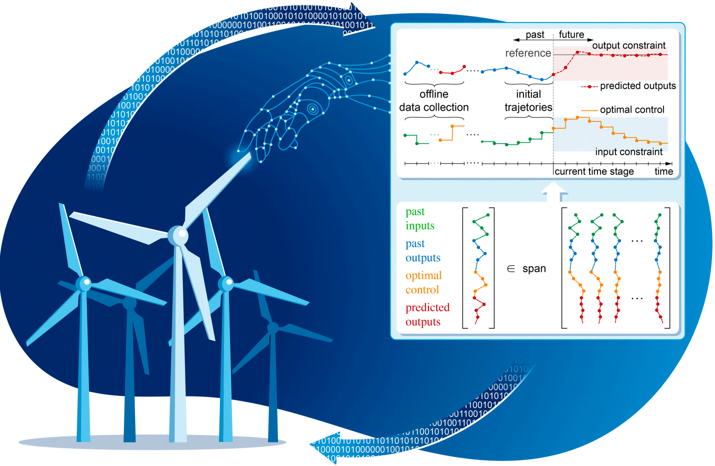
Data-Enabled Predictive Control of Autonomous Energy Systems



Florian Dörfler

ETH Zürich

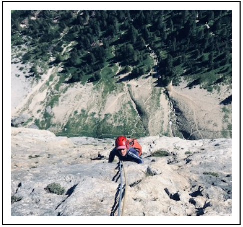
KIOS 2024 Graduate School

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

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

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 **?!?**

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]

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]

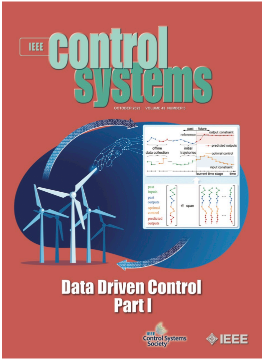
Thoughts on data in control systems

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 **?!?**

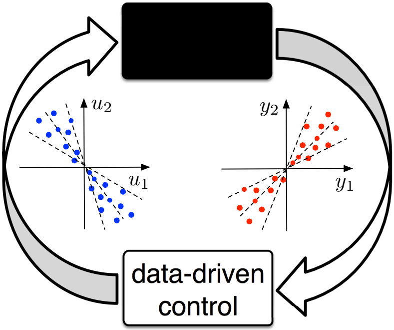
“One of the major developments in control over the past decade – & one of the most important moving forward – is the interaction of ML & control systems.” [CSS roadmap]



Approaches to data-driven control

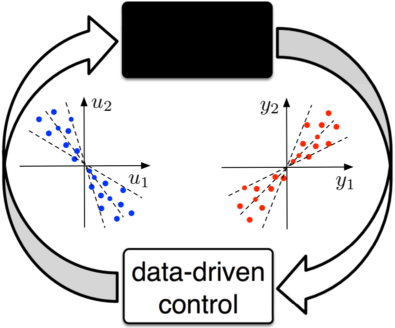
- ***indirect data-driven control*** via models:

  data $\xrightarrow{\text{SysID}}$ model + uncertainty $\rightarrow$ control



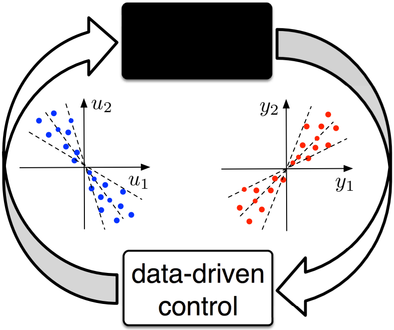
Approaches to data-driven control

- ***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 ?



Approaches to data-driven control

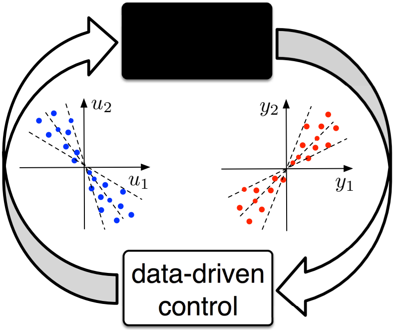
- ***indirect data-driven control*** via models:
  data $\xrightarrow{\text{SysID}}$ model + uncertainty $\rightarrow$ control
- growing trend: ***direct data-driven control***
  by-passing models . . . (again) hyped, why ?



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

Approaches to data-driven control

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



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

Example 1973: autotuned PID

Approaches to data-driven control

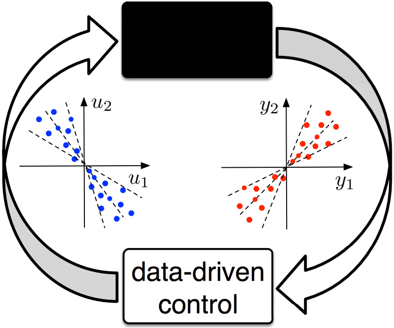
- ***indirect data-driven control*** via models:
  data $\xrightarrow{\text{SysID}}$ model + uncertainty $\to$ control

- 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

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

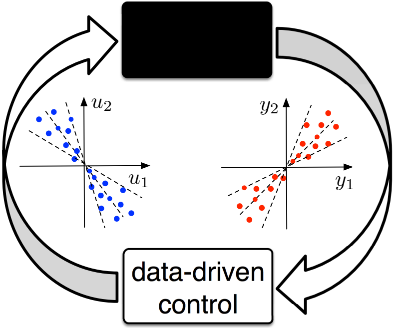


Central promise: *It is often easier to learn a control policy from data rather than a model.*

Example 1973: autotuned PID

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***

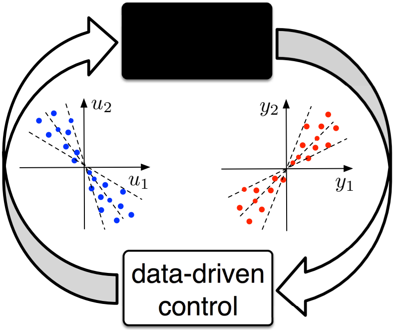
  $\rightarrow$ too cumbersome, models not identified for control, incompatible uncertainty estimates, ...

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

Example 1973: autotuned PID

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)

input

state

subject to (u, x) satisfy state-space model

Abstraction reveals pros & cons

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

minimize control cost (u, x)

subject to (u, x) satisfy state-space model

I output

where x estimated from (u, y) & model

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

Abstraction reveals pros & cons

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

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

$\rightarrow$ nested multi-level optimization problem

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)

$\rightarrow$ nested multi-level optimization problem

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

$\rightarrow$ nested multi-level optimization problem

outer optimization

middle opt.

inner opt.

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

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

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)

no 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

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 (?)

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)

no 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***

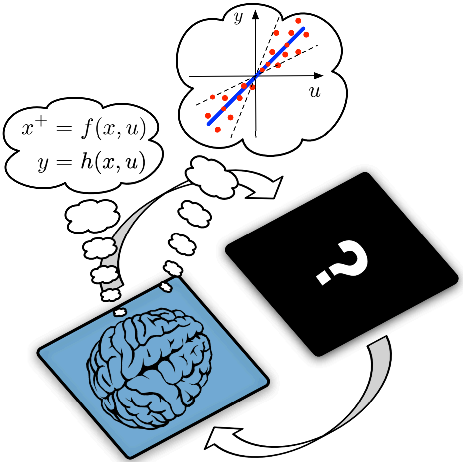
modular vs. end-2-end

suboptimal (?) vs. optimal

convex vs. non-convex (?)

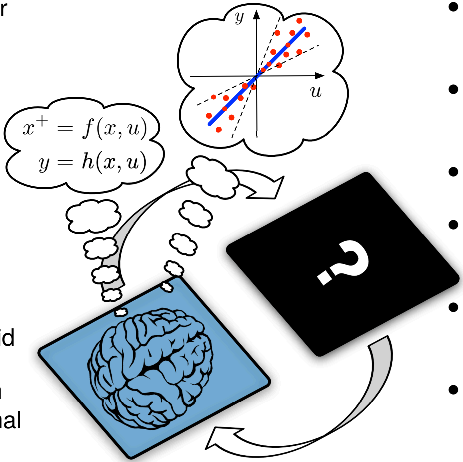
Additionally: account for ***uncertainty*** (hard to propagate in indirect approach)

Indirect (models) vs. direct (data)



Indirect (models) vs. direct (data)

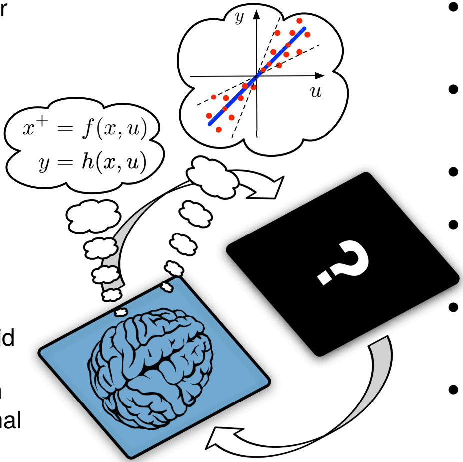
- models are useful for design & beyond
- modular $\to$ easy to debug & interpret
- id = noise filtering
- id = projection on model class
- harder to propagate uncertainty through id
- no robust separation principle $\to$ suboptimal
- …



- 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
- …

Indirect (models) vs. direct (data)

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



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

A direct approach: dictionary + MPC

① trajectory ***dictionary learning***

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

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

→ huge ***theory vs. practice*** gap

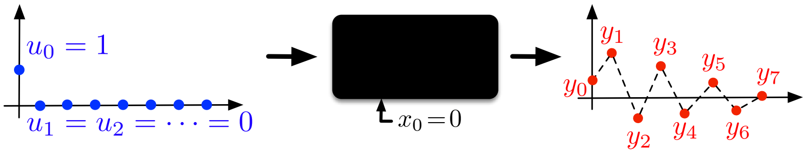
A direct approach: dictionary + MPC

① trajectory ***dictionary learning***

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

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

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



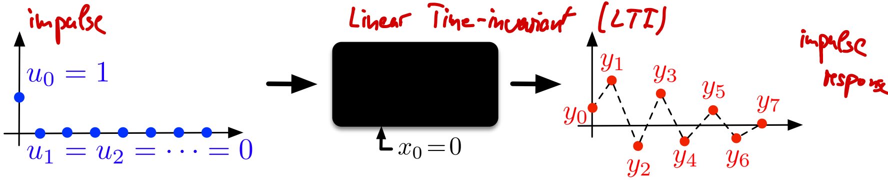
A direct approach: dictionary + MPC

① trajectory ***dictionary learning***

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

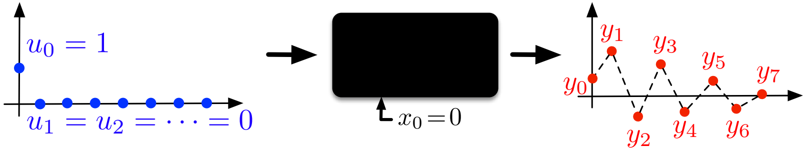
② ***MPC*** optimizing over dictionary span

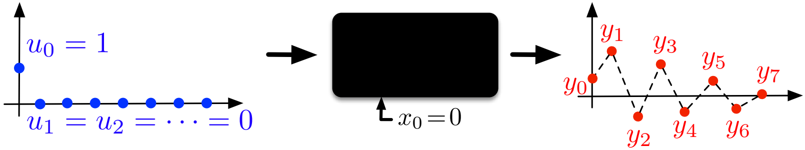
→ huge ***theory vs. practice*** gap
→ back to basics: ***impulse response***



$$\left.\begin{matrix} u = \delta_0 \\ x_0 = 0 \end{matrix}\right\} \; y(t) = \underset{\substack{\uparrow \\ \text{impulse response}}}{g(t)} = \{g_0, g_1, g_2 \dots\}$$

response to any other input $u(t)$ is $y(t) = \sum_{\tau=0}^{t-1} g(t-\tau)\cdot u(\tau)$


$u_0 = 1$
$u_1 = u_2 = \cdots = 0$
$x_0 = 0$
y_0
y_1
y_2
y_3
y_4
y_5
y_6
y_7

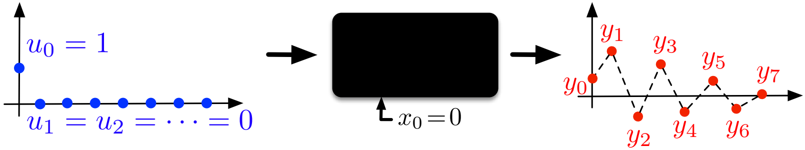


Now what if we had the impulse response recorded in our data-library?

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

response to any new input $u_{future}(t)$ is

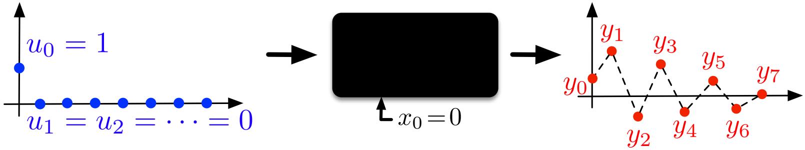
$$y_{future}(t) = \sum_{\tau=0}^{t} y^d(t-\tau)\cdot u_{future}(t)$$



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

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

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

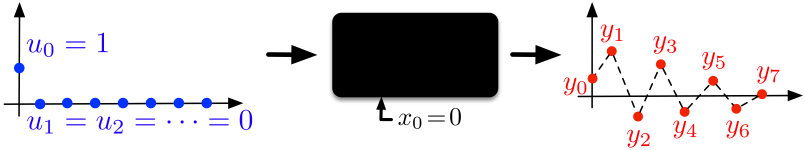


Now what if we had the impulse response recorded in our data-library?

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

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

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



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***

   + tomatoes

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)

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

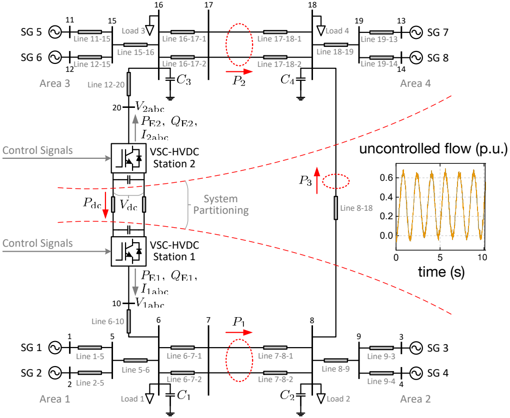
- We teach this material also in the ETH Zürich bachelor & have plenty of ***background material*** + implementation experience $\longrightarrow$ please reach out to me or Saverio if you need anything

- 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

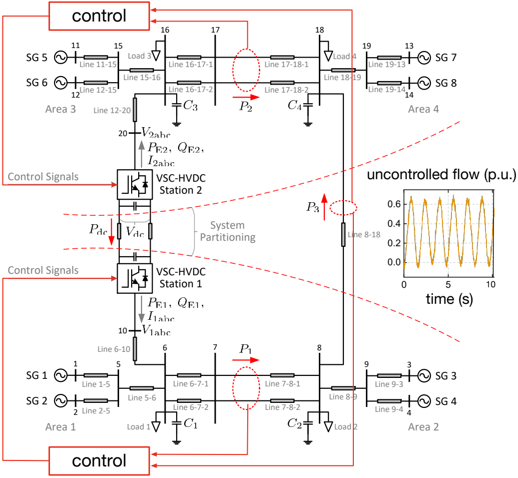
control objective: oscillation damping



Preview

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

control objective: oscillation damping

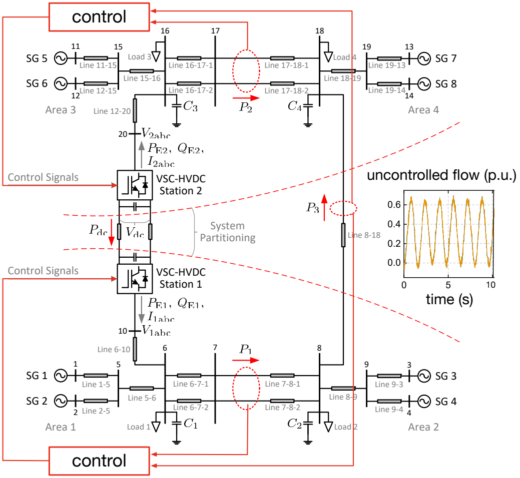


Preview

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, ...)

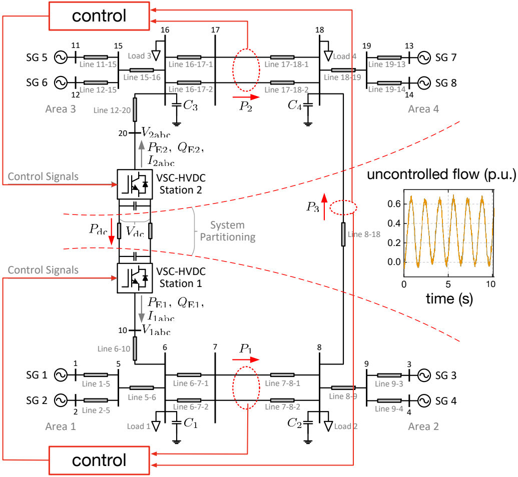


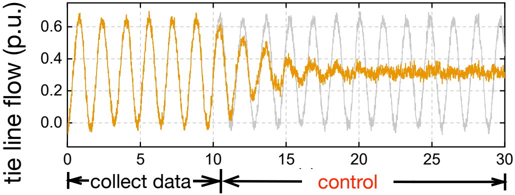
Preview

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, . . .)





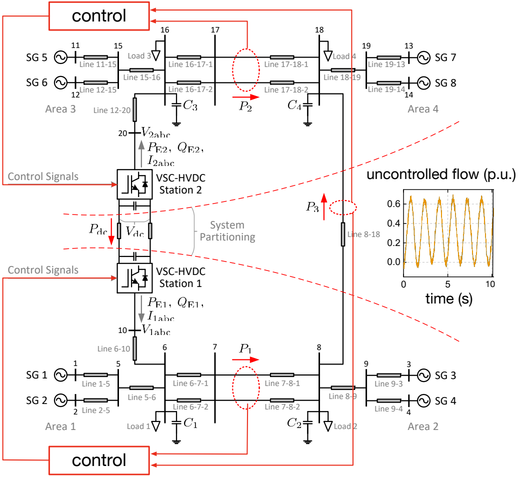
Preview

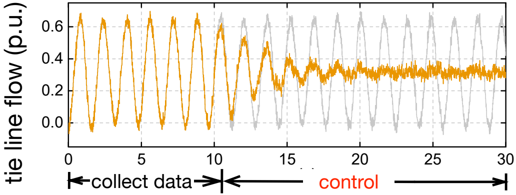
complex 4-area power ***system***:

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?

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?

Dear Linbin and Florian,

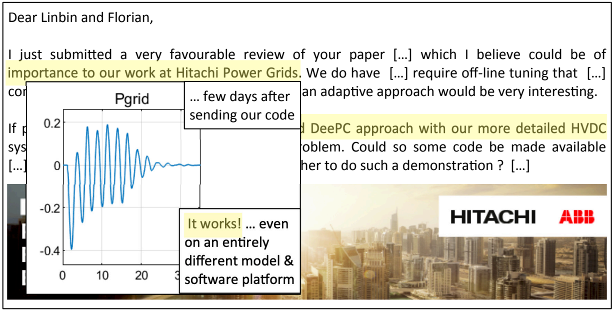
I just submitted a very favourable review of your paper […] which I believe could be of importance to our work at Hitachi Power Grids. We do have […] require off-line tuning that […] comissisioning engineer can do on his own […] an adaptive approach would be very interesting.

If possible I would like to try the decentralized DeePC approach with our more detailed HVDC system models on the interarea oscillation problem. Could so some code be made available […] ? Would you be interested in working together to do such a demonstration ? […]

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?


Dear Linbin and Florian,

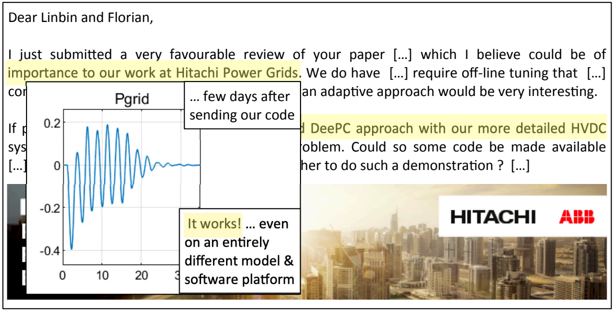
I just submitted a very favourable review of your paper [...] which I believe could be of importance to our work at Hitachi Power Grids. We do have [...] require off-line tuning that [...]
cor … an adaptive approach would be very interesting.

If p … d DeePC approach with our more detailed HVDC
sys … oblem. Could so some code be made available
[...] … her to do such a demonstration ? [...]

Reality check: black magic or hoax ?

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

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



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

LTI system representations

- ARX: exogeneous input, auto-regressive

$$y(t+2) + 2y(t+1) + 3y(t) = 4u(t)$$

- ARX → state space:

$$x(t) = \begin{bmatrix} y(t) \\ y(t+1) \end{bmatrix}$$

$$x(t+1) = \begin{bmatrix} 0 & 1 \\ -3 & -2 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 4 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(t)$$

- ARX → transfer function:

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

⇝ these are all parametric kernel representations

$$\begin{cases} y(t+2) + 2y(t+1) + 3y(t) = 4u(t) \\ \text{time shift } z \cdot y(t) = y(t+1) \end{cases}$$

$$\left[z^2 + 2z + 3,\ 4\right] \begin{bmatrix} y(t) \\ u(t) \end{bmatrix} = 0$$

"kernel representation"

Behavioral view on dynamical systems

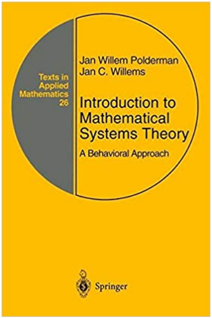
Definition: A discrete-time ***dynamical system*** is a 3-tuple $(\mathbb{Z}_{\geq 0}, \mathbb{W}, \mathscr{B})$ where

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

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

(iii) $\mathscr{B} \subseteq \underbrace{\mathbb{W}^{\mathbb{Z}_{\geq 0}}}_{\text{set of all discrete time series } \mathbb{W}}$ is the *behavior*.

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



Behavioral view on dynamical systems

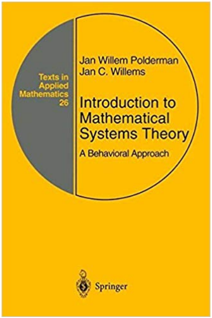
Definition: A discrete-time ***dynamical system*** is a 3-tuple $(\mathbb{Z}_{\geq 0}, \mathbb{W}, \mathscr{B})$ where

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

Behavioral view on dynamical systems

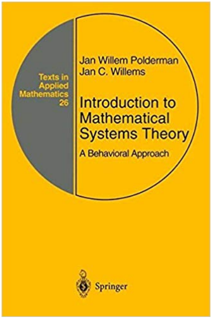
Definition: A discrete-time ***dynamical system*** is a 3-tuple $(\mathbb{Z}_{\geq 0}, \mathbb{W}, \mathscr{B})$ where

(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*.

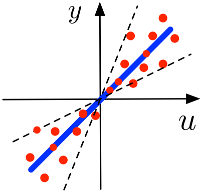
$\mathscr{B}$ is the set of all trajectories



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 $\sigma w_t = w_{t+1}$.



Behavioral view on dynamical systems

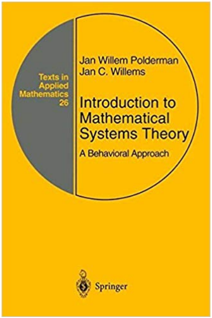
Definition: A discrete-time ***dynamical system*** is a 3-tuple $(\mathbb{Z}_{\geq 0}, \mathbb{W}, \mathscr{B})$ where

(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*.

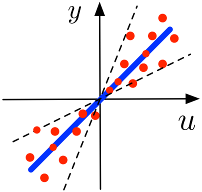
$\mathscr{B}$ is the set of all trajectories



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

- model $\underbrace{}$

$$x(t+1) = A\,\overset{\in \mathbb{R}^n}{x(t)} + B\,\overset{\in \mathbb{R}^m}{u(t)}$$

$$\underset{\in \mathbb{R}^p}{y(t)} = C\,x(t) + D\,u(t)$$

$$x(0) = x_{ini}$$

$$x(0) = x_{ini}$$

$$x(1) = A x_{ini} + B u(0)$$

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

$$\vdots$$

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

in vector notation:

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

compactly: $y = \mathcal{O}_T \cdot x_{ini} + \mathcal{C}_T \cdot u$

- observability: x_{ini} can be reconstructed from (y, u) from $\Leftrightarrow$ rank $\mathcal{O}_T = n$

- the smallest integer ℓ so that $\mathcal{O}_\ell$ has rank n is called the lag of the system: $\begin{cases} \ell = n & \text{for a SISO system} \\ \ell \leq n & \text{for MIMO} \end{cases}$

$\Rightarrow$ given past data $u_{ini} = \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(T_{ini}) \end{bmatrix}$ and $y_{ini} = \begin{bmatrix} y(0) \\ \vdots \\ y(T_{ini}) \end{bmatrix}$

$\Rightarrow$ x_{ini} can be uniquely reconstructed $\Leftrightarrow$ $T_{ini} \geq \ell$

dimension of the LTI trajectory space

$\forall\, x_{ini} \in \mathbb{R}^n$, what is the dimension of

$$\begin{bmatrix} u \\ \hline y \end{bmatrix} = \begin{bmatrix} u(0) \\ \vdots \\ u(T) \\ \hline y(0) \\ \vdots \\ y(T) \end{bmatrix} = \begin{bmatrix} 0 & I \\ \underbrace{\mathcal{O}_T}_{\substack{\text{column has} \\ \text{rank } n \text{ for } T \geq \ell}} & \underbrace{\mathcal{C}_T}_{\substack{\text{column always has} \\ \text{rank } m \cdot T}} \end{bmatrix} \begin{bmatrix} \overbrace{x_{ini}}^{\in \mathbb{R}^n} \\ \underbrace{u}_{\mathbb{R}^{m \cdot T}} \end{bmatrix}$$

$\Rightarrow$ dimension of $\begin{bmatrix} u \\ y \end{bmatrix} = m \cdot T + n$ for $T \geq \ell$

LTI systems & matrix time series

foundation of subspace system identification & signal recovery algorithms

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)

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)

$\Rightarrow$

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

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

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)

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

1st experiment   2nd   3rd ...

Fundamental Lemma

Given: data $\begin{pmatrix} u_i^d \\ y_i^d \end{pmatrix} \in \mathbb{R}^{m+p}$

Fundamental Lemma

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

Fundamental Lemma

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

$$\text{set of all } T\text{-length trajectories } = \underbrace{\left\{ (u, y) \in \mathbb{R}^{(m+p)T} : \exists x \in \mathbb{R}^{nT} \text{ s.t. } x^+ = Ax + Bu\,,\ y = Cx + Du \right\}}_{\text{parametric state-space model}}$$

Fundamental Lemma

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

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

Fundamental Lemma

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

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

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

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

set of all T-length trajectories $=$

$$\left\{ (u, y) \in \mathbb{R}^{(m+p)T} : \exists x \in \mathbb{R}^{nT} \text{ s.t. } x^+ = Ax + Bu \, , \; y = Cx + Du \right\}$$

$$= \quad \text{colspan} \begin{bmatrix} \begin{pmatrix} u^d_{1,1} \\ y^d_{1,1} \end{pmatrix} & \begin{pmatrix} u^d_{1,2} \\ y^d_{1,2} \end{pmatrix} & \begin{pmatrix} u^d_{1,3} \\ y^d_{1,3} \end{pmatrix} & \cdots \\ \begin{pmatrix} u^d_{2,1} \\ y^d_{2,1} \end{pmatrix} & \begin{pmatrix} u^d_{2,2} \\ y^d_{2,2} \end{pmatrix} & \begin{pmatrix} u^d_{2,3} \\ y^d_{2,3} \end{pmatrix} & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ \begin{pmatrix} u^d_{T,1} \\ y^d_{T,1} \end{pmatrix} & \begin{pmatrix} u^d_{T,2} \\ y^d_{T,2} \end{pmatrix} & \begin{pmatrix} u^d_{T,3} \\ y^d_{T,3} \end{pmatrix} & \cdots \end{bmatrix}$$

all trajectories constructible from finitely many previous trajectories

$$\text{set of all } T\text{-length trajectories } = \left\{ (u, y) \in \mathbb{R}^{(m+p)T} : \exists x \in \mathbb{R}^{nT} \text{ s.t. } x^{+} = Ax + Bu \,,\; y = Cx + Du \right\}$$

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

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

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

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

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

all trajectories constructible from finitely many previous trajectories

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

A note on persistency of excitation

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

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

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

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

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

Input design for Fundamental Lemma

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 $\begin{bmatrix} u_1 & u_2 & \cdots & u_{T_d-T+1} \\ \vdots & u_3 & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ u_T & u_{T+1} & \cdots & u_{T_d} \end{bmatrix}$ is of full rank.

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. $m \cdot T$

for full rank: $T_d - T + 1 \geq mT \Rightarrow T_d$ is sufficiently

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 $\left(\mathscr{H}\left(\begin{smallmatrix} u^d \\ y^d \end{smallmatrix}\right)\right) = mT + n$.

Data matrix structures & preprocessing

- trajectory matrix $\mathcal{H}(w)$ $\begin{bmatrix} u^d \\ y^d \end{bmatrix}$ $= \left[\begin{array}{c|c|c} \text{1st experiment} & \text{2nd experiment} & \cdots \end{array}\right]$

  requires independent experiment

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

  requires one long experiment

- Hankel matrix $= \mathcal{H}(w) = \begin{bmatrix} w_1 & w_2 & w_3 & \\ \vdots & \vdots & \vdots & \cdots \\ w_T & w_{T+1} & w_{T+2} & \end{bmatrix}$

  requires one short experiment

... or any combinations ...

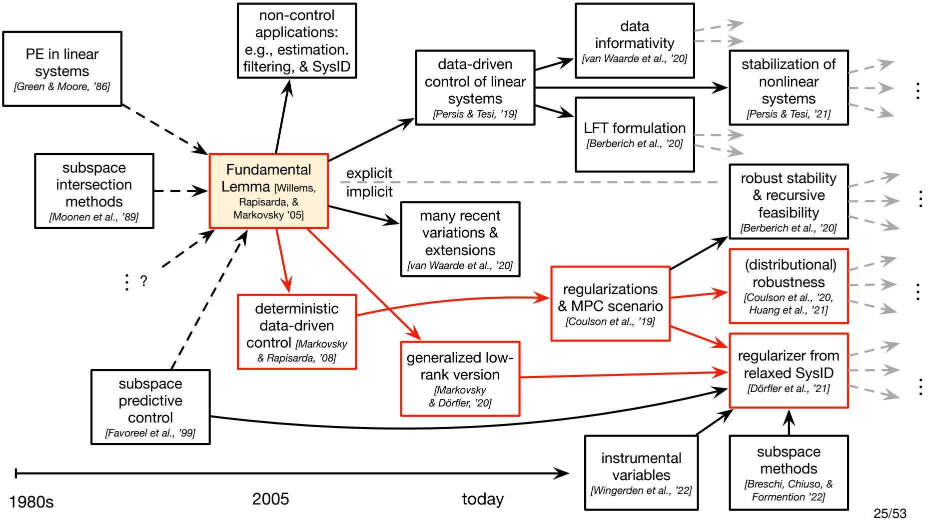
Pre-procession of noisy data: if w^d is noisy, then all of the above matrices have full rank.

low-rank-preprocessing: $\min_{\hat{w}} \|\hat{w} - w^d\|$
= find the closest low-rank data
$\text{rank}(H(\hat{w})) = mL+n$

=> standard solution is to take an SVD of $H(w^d)$ and only keep the largest $mL+n$ singular values

=> is optimal if the matrix is unstructured
... but does not apply time series are correlated

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



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

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

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

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

hard operational or safety **constraints**

Output Model Predictive Control (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**

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***

– Willems '07

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

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

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*

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**)

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**

of length $T_{ini} \geq \ell$

- trajectory matrix $\mathscr{H}\begin{pmatrix} u^{\mathsf{d}} \\ y^{\mathsf{d}} \end{pmatrix}$ from past experimental data — collected **offline** (could be adapted online)

Data-enabled Predictive Control (DeePC)

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

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 to make it work

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

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

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

Regularizations to make it work

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

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

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

measurement noise

$\rightarrow$ infeasible y_{ini} estimate

$\rightarrow$ estimation slack σ

$\rightarrow$ moving-horizon least-square filter

Regularizations to make it work

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

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

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

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 $h(g)$

Regularizations to make it work

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

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

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

measurement noise

$\rightarrow$ 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

Regularizations to make it work

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

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

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

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} = {motion primitives} ∼ low-order basis

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

Regularizations to make it work

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

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

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

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} = {motion primitives} ∼ low-order basis

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

$$\min_{x} \max_{\|\Delta\| \leq \rho} \|(A+\Delta)x - b\| \overset{=}{\leq} \min_{x} \max_{\|\Delta\| \leq \rho} \|Ax - b\| + \|\Delta x\| = \min_{x} \|Ax - b\| + \rho\|x\|$$

regularization

$$\Updownarrow$$

incorporating priors

+ implicit SysID

Regularization = relaxing low-rank approximation in pre-processing

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

optimal control

low-rank approximation

Regularization = relaxing low-rank approximation in pre-processing

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

optimal control

low-rank approximation

↓ sequence of convex relaxations ↓

$$\begin{aligned}
&\text{minimize}_{u,y,g} && \text{control cost}\,(u,y) \\
&\text{subject to} && \begin{bmatrix} u \\ y \end{bmatrix} = \mathscr{H}\begin{pmatrix} \hat{u} \\ \hat{y} \end{pmatrix} g \;, \quad \|g\|_0 \leq mL + n \\
&\text{where} && \begin{pmatrix} \hat{u} \\ \hat{y} \end{pmatrix} \in \operatorname{argmin} \left\| \begin{pmatrix} \hat{u} \\ \hat{y} \end{pmatrix} - \begin{pmatrix} u^{\mathrm{d}} \\ y^{\mathrm{d}} \end{pmatrix} \right\| \\
& && \quad \text{subject to } \operatorname{rank}\left(\mathscr{H}\left(\begin{smallmatrix} \hat{u} \\ \hat{y} \end{smallmatrix}\right)\right) = mL + n
\end{aligned}$$

Regularization = relaxing low-rank approximation in pre-processing

$$
\left.\begin{aligned}
&\text{minimize}_{u,y,g} && \text{control cost}\,(u,y)\\
&\text{subject to} && \begin{bmatrix} u \\ y \end{bmatrix} = \mathscr{H}\begin{pmatrix} \hat{u} \\ \hat{y} \end{pmatrix} g
\end{aligned}\right\}\ \text{optimal control}
$$

$$
\left.\begin{aligned}
&\text{where} && \begin{pmatrix} \hat{u} \\ \hat{y} \end{pmatrix} \in \operatorname{argmin} \left\| \begin{pmatrix} \hat{u} \\ \hat{y} \end{pmatrix} - \begin{pmatrix} u^{\mathrm{d}} \\ y^{\mathrm{d}} \end{pmatrix} \right\|\\
& && \text{subject to } \operatorname{rank}\left(\mathscr{H}\begin{pmatrix} \hat{u} \\ \hat{y} \end{pmatrix}\right) = mL + n
\end{aligned}\right\}\ \text{low-rank approximation}
$$

↓ sequence of convex relaxations ↓

$$
\begin{aligned}
&\text{minimize}_{u,y,g} && \text{control cost}\,(u,y)\\
&\text{subject to} && \begin{bmatrix} u \\ y \end{bmatrix} = \mathscr{H}\begin{pmatrix} \hat{u} \\ \hat{y} \end{pmatrix} g\ , \quad \|g\|_0 \leq mL + n\\
&\text{where} && \begin{pmatrix} \hat{u} \\ \hat{y} \end{pmatrix} \in \operatorname{argmin} \left\| \begin{pmatrix} \hat{u} \\ \hat{y} \end{pmatrix} - \begin{pmatrix} u^{\mathrm{d}} \\ y^{\mathrm{d}} \end{pmatrix} \right\|\\
& && \cancel{\text{subject to } \operatorname{rank}\left(\mathscr{H}\begin{pmatrix} \hat{u} \\ \hat{y} \end{pmatrix}\right) = mL + n}
\end{aligned}
$$

Regularization = relaxing low-rank approximation in pre-processing

$$\left.\begin{array}{ll}\text{minimize}_{u,y,g} & \text{control cost}\,(u,y) \\ \text{subject to} & \begin{bmatrix} u \\ y \end{bmatrix} = \mathscr{H}\begin{pmatrix} \hat{u} \\ \hat{y} \end{pmatrix} g\end{array}\right\} \text{optimal control}$$

$$\left.\begin{array}{ll}\text{where} & \begin{pmatrix} \hat{u} \\ \hat{y} \end{pmatrix} \in \text{argmin} \left\| \begin{pmatrix} \hat{u} \\ \hat{y} \end{pmatrix} - \begin{pmatrix} u^{\text{d}} \\ y^{\text{d}} \end{pmatrix} \right\| \\ & \quad \text{subject to rank}\left(\mathscr{H}\left(\begin{smallmatrix} \hat{u} \\ \hat{y} \end{smallmatrix}\right)\right) = mL + n\end{array}\right\} \text{low-rank approximation}$$

↓ sequence of convex relaxations ↓

$$\begin{array}{ll}\text{minimize}_{u,y,g} & \text{control cost}\,(u,y) \\ \text{subject to} & \begin{bmatrix} u \\ y \end{bmatrix} = \mathscr{H}\begin{pmatrix} \hat{u} \\ \hat{y} \end{pmatrix} g \ , \quad \|g\|_0 \leq mL + n \\ \text{where} & \begin{pmatrix} \hat{u} \\ \hat{y} \end{pmatrix} \in \text{argmin} \left\| \begin{pmatrix} \hat{u} \\ \hat{y} \end{pmatrix} - \begin{pmatrix} u^{\text{d}} \\ y^{\text{d}} \end{pmatrix} \right\| = \begin{pmatrix} u^{\text{d}} \\ y^{\text{d}} \end{pmatrix} \\ & \quad \cancel{\text{subject to rank}\left(\mathscr{H}\left(\begin{smallmatrix} \hat{u} \\ \hat{y} \end{smallmatrix}\right)\right) = mL + n}\end{array}$$

Regularization = relaxing low-rank approximation in pre-processing

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

optimal control

low-rank approximation

↓ sequence of convex relaxations ↓

$$\begin{aligned} &\text{minimize}_{u,y,g} && \text{control cost}\big(u,y\big) \\ &\text{subject to} && \begin{bmatrix} u \\ y \end{bmatrix} = \mathscr{H}\begin{pmatrix} u^{d} \\ y^{d} \end{pmatrix} g\,, \quad \|g\|_0 \leq mL + n \end{aligned}$$

Regularization = relaxing low-rank approximation in pre-processing

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

optimal control

low-rank approximation

$\downarrow$ sequence of convex relaxations $\downarrow$

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

Regularization = relaxing low-rank approximation in pre-processing

$$\left.\begin{array}{ll} \text{minimize}_{u,y,g} & \text{control cost}\,(u,y) \\ \text{subject to} & \begin{bmatrix} u \\ y \end{bmatrix} = \mathscr{H}\begin{pmatrix} \hat{u} \\ \hat{y} \end{pmatrix} g \end{array}\right\} \text{optimal control}$$

$$\left.\begin{array}{ll} \text{where} & \begin{pmatrix} \hat{u} \\ \hat{y} \end{pmatrix} \in \text{argmin} \left\| \begin{pmatrix} \hat{u} \\ \hat{y} \end{pmatrix} - \begin{pmatrix} u^{\mathrm{d}} \\ y^{\mathrm{d}} \end{pmatrix} \right\| \\ & \quad \text{subject to rank}\left(\mathscr{H}\left(\begin{smallmatrix} \hat{u} \\ \hat{y} \end{smallmatrix}\right)\right) = mL + n \end{array}\right\} \text{low-rank approximation}$$

↓ sequence of convex relaxations ↓

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

Regularization = relaxing low-rank approximation in pre-processing

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

optimal control

low-rank approximation

$\downarrow$ sequence of convex relaxations $\downarrow$

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

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

Regularization = relaxing low-rank approximation in pre-processing

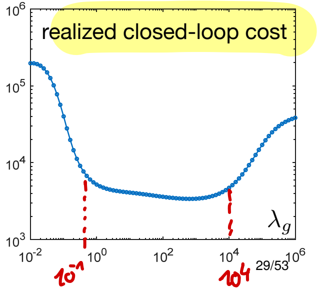
$$
\left.\begin{array}{ll}
\text{minimize}_{u,y,g} & \text{control cost}\,(u,y) \\
\text{subject to} & \begin{bmatrix} u \\ y \end{bmatrix} = \mathscr{H}\begin{pmatrix} \hat{u} \\ \hat{y} \end{pmatrix} g
\end{array}\right\} \text{optimal control}
$$

$$
\left.\begin{array}{ll}
\text{where} & \begin{pmatrix} \hat{u} \\ \hat{y} \end{pmatrix} \in \text{argmin} \left\| \begin{pmatrix} \hat{u} \\ \hat{y} \end{pmatrix} - \begin{pmatrix} u^{\mathrm{d}} \\ y^{\mathrm{d}} \end{pmatrix} \right\| \\
& \quad \text{subject to rank}\left(\mathscr{H}\begin{pmatrix} \hat{u} \\ \hat{y} \end{pmatrix}\right) = mL + n
\end{array}\right\} \text{low-rank approximation}
$$

↓ sequence of convex relaxations ↓

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

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



Certainty-Equivalence Regularizer

ARX representation of predictor:

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

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

$\Rightarrow$ $y = K \cdot \begin{bmatrix} u_{ini} \\ y_{ini} \\ u \end{bmatrix}$ + "noise"

where K is learned from data

$K = \arg\min \left\| Y_f - K \begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix} \right\|$

$= Y_f \cdot \begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix}^{+}$

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

DeePC representation of predictor:

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

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

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

$g_{ker} \in \operatorname{kernel} \begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix}$

to re-create the model-based solution we need to penalize g_{ker}

Regularization $\Leftrightarrow$ reformulate subspace ID

$\rightarrow$ ***indirect SysID + control*** problem

$$\begin{aligned} \text{minimize}_{u,y} \quad & \text{control cost}(u, y) \\ \text{subject to} \quad & y = K^\star \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \end{bmatrix} \end{aligned}$$

Regularization $\Leftrightarrow$ reformulate subspace ID

partition data as in subspace ID:

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

$\rightarrow$ ***indirect SysID + control*** problem

$$\begin{aligned} &\text{minimize}_{u,y} \quad \text{control cost}(u,y) \\ &\text{subject to} \quad y = K^\star \begin{bmatrix} u_{\mathsf{ini}} \\ y_{\mathsf{ini}} \\ u \end{bmatrix} \end{aligned}$$

Regularization $\Leftrightarrow$ reformulate subspace ID

partition data as in subspace ID:

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

$\rightarrow$ ***indirect SysID + control*** problem

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

Regularization $\Leftrightarrow$ reformulate subspace ID

partition data as in subspace ID:

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

ID of optimal multi-step predictor

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

$\rightarrow$ ***indirect SysID + control*** problem

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

Regularization $\Leftrightarrow$ reformulate subspace ID

partition data as in subspace ID:

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

ID of optimal multi-step predictor

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

$\rightarrow$ ***indirect SysID + control*** problem

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

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

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

Regularization $\Leftrightarrow$ reformulate subspace ID

partition data as in subspace ID:

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

ID of optimal multi-step predictor as in SPC: $K^\star = Y_F \begin{bmatrix} U_{\mathsf{p}} \\ Y_{\mathsf{p}} \\ U_{\mathsf{f}} \end{bmatrix}^\dagger$

$\to$ ***indirect SysID + control*** problem

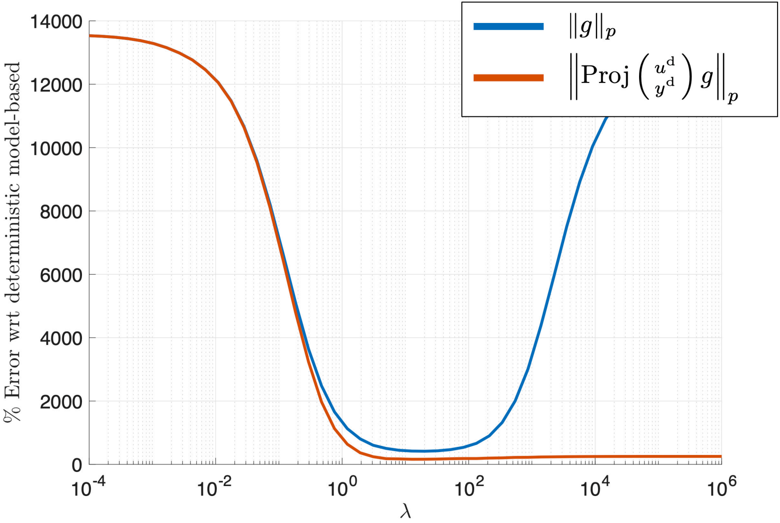
$$\begin{aligned} &\text{minimize}_{u,y} && \text{control cost}(u,y) \\ &\text{subject to} && y = K^\star \begin{bmatrix} u_{\mathsf{ini}} \\ y_{\mathsf{ini}} \\ u \end{bmatrix} \\ &\text{where} && K^\star = \text{argmin}_K \left\| Y_F - K \begin{bmatrix} U_{\mathsf{p}} \\ Y_{\mathsf{p}} \\ U_{\mathsf{f}}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

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

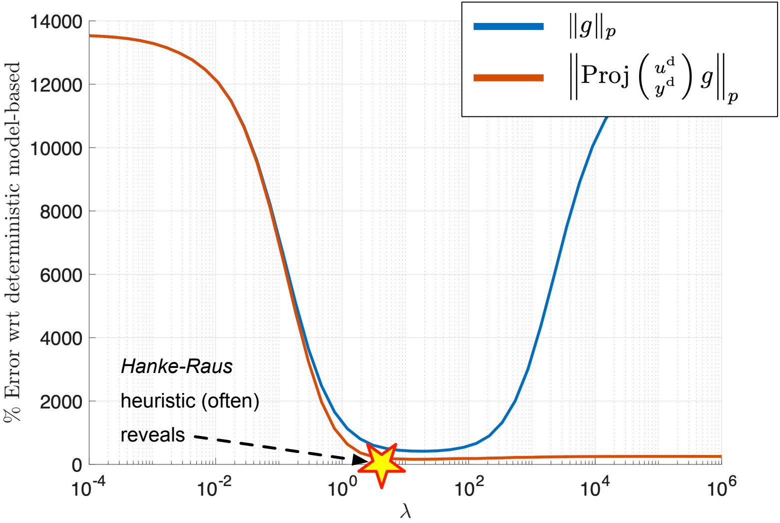
where $\text{Proj}\begin{pmatrix} u^{\mathsf{d}} \\ y^{\mathsf{d}} \end{pmatrix}$ projects orthogonal to $\ker \begin{bmatrix} U_{\mathsf{p}} \\ Y_{\mathsf{p}} \\ U_{\mathsf{f}} \end{bmatrix}$

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

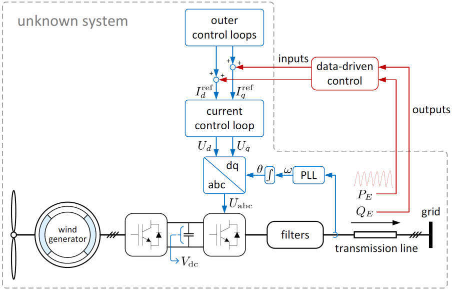
Performance of regularizers applied to a stochastic LTI system



Performance of regularizers applied to a stochastic LTI system

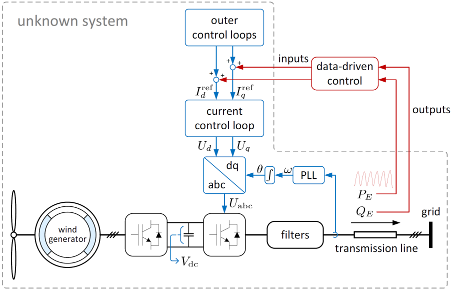


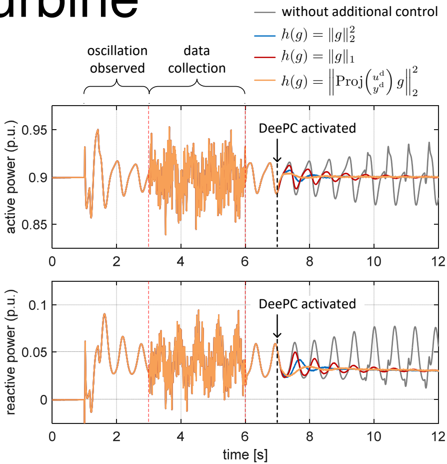
Case study: wind turbine



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

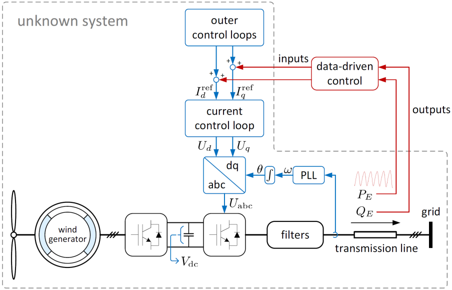
Case study: wind turbine

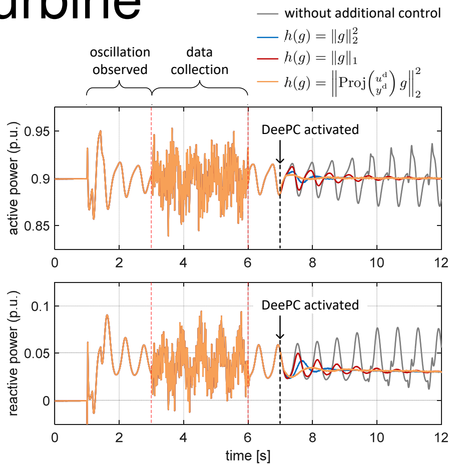




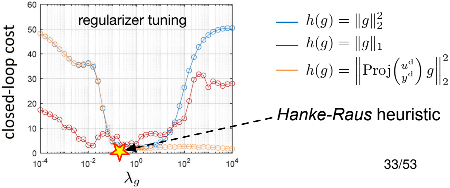
- detailed ***industrial model***: 37 states & highly nonlinear (abc $\leftrightarrow$ dq, MPTT, PLL, power specs, dynamics, etc.)
- turbine & grid model ***unknown*** to commissioning engineer & operator
- weak grid + PLL + fault $\rightarrow$ ***loss of sync***
- disturbance to be rejected by ***DeePC***

Case study: wind turbine

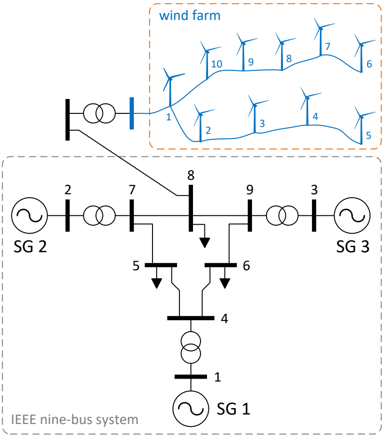




- detailed ***industrial model***: 37 states & highly nonlinear (abc $\leftrightarrow$ dq, MPTT, PLL, power specs, dynamics, etc.)
- turbine & grid model ***unknown*** to commissioning engineer & operator
- weak grid + PLL + fault $\rightarrow$ ***loss of sync***
- disturbance to be rejected by ***DeePC***

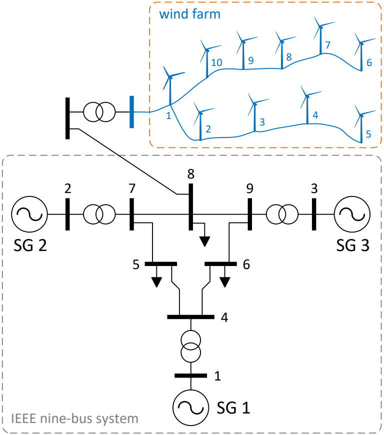


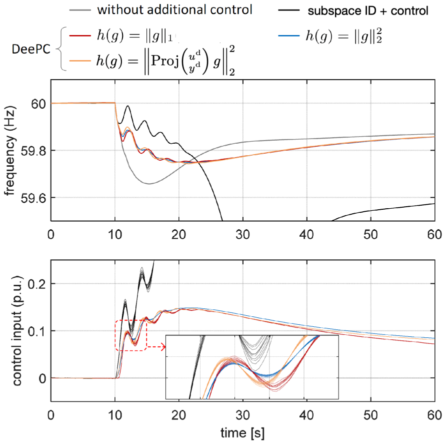
Case study ++ : wind farm



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

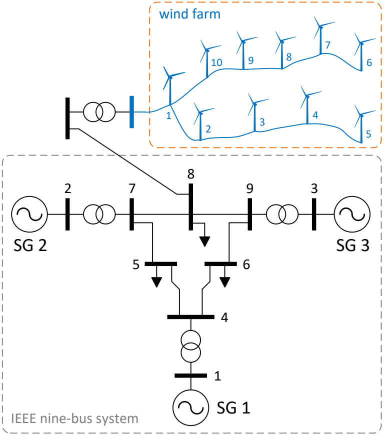
Case study ++ : wind farm

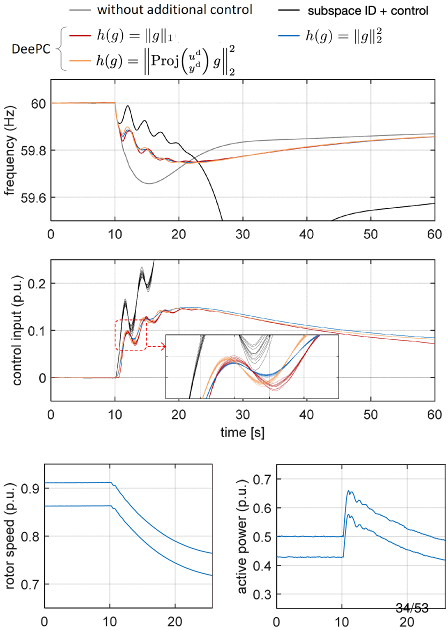




- ***high-fidelity models*** for turbines, machines, & IEEE-9-bus system
- ***fast frequency response*** via ***decentralized DeePC*** at turbines

Case study ++ : wind farm

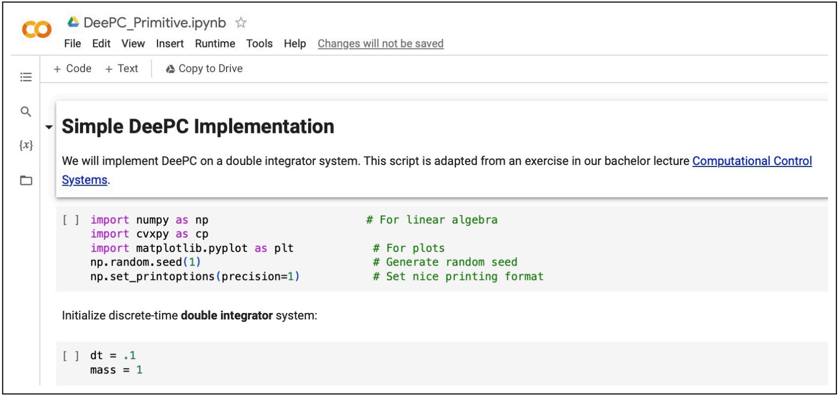




- ***high-fidelity models*** for turbines, machines, & IEEE-9-bus system
- ***fast frequency response*** via ***decentralized DeePC*** at turbines

DeePC is easy to implement $\rightarrow$ try it !

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


DeePC_Primitive.ipynb

File Edit View Insert Runtime Tools Help Changes will not be saved

+ Code + Text Copy to Drive

Simple DeePC Implementation

We will implement DeePC on a double integrator system. This script is adapted from an exercise in our bachelor lecture Computational Control Systems.

```
import numpy as np                          # For linear algebra
import cvxpy as cp
import matplotlib.pyplot as plt              # For plots
np.random.seed(1)                            # Generate random seed
np.set_printoptions(precision=1)             # Set nice printing format
```

Initialize discrete-time **double integrator** system:

```
dt = .1
mass = 1
```

Towards a theory for nonlinear systems

Towards a theory for nonlinear systems

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

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

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

Towards a theory for nonlinear systems

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

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

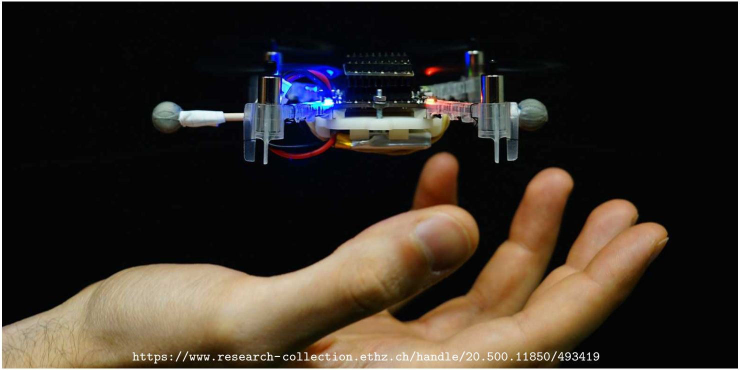
$\rightarrow$ nonlinear dynamics can be approximated by LTI on finite horizon

regularization singles out relevant features / basis functions in data

Towards a theory for nonlinear systems

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

regularization singles out relevant features / basis functions in data

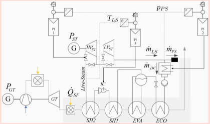


Works very well across case studies

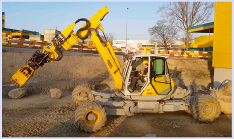
quad coptor fig-8 tracking

quadruped (by Fawcett, Afsari Amers, & Hamed)

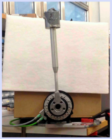
greenhouse automation (by Automatoes)



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

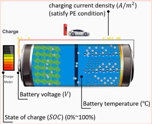


robotic excavator

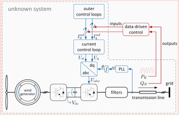


pendulum swing up

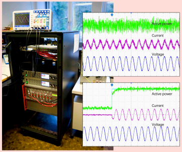
traffic coordination (by J. Wang et al.)



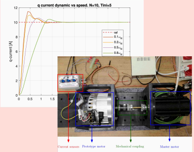
battery charging (by K. Chen et al.)



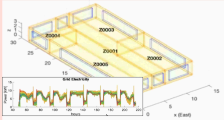
wind turbine control



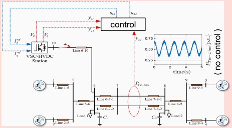
grid-connected converter



synchronous motor drive



energy hub & building automation



power system oscillation damping

regularization

$$\Updownarrow$$

robustification

Distributional robustification beyond LTI

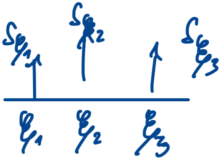
- ***problem abstraction*** : $\min_{x \in \mathcal{X}} \; c\left(\widehat{\xi}, x\right)$

  where $\widehat{\xi}$ denotes *measured* data

Distributional robustification beyond LTI

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

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



Distributional robustification beyond LTI

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

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

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

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

Distributional robustification beyond LTI

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

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

$\Rightarrow$ ***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

Distributional robustification beyond LTI

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

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

$\Rightarrow$ ***poor out-of-sample performance*** of above sample-average solution $x^\star$
for real problem: $\mathbb{E}_{\xi\sim\mathbb{P}}\big[c(\xi, x^\star)\big]$ 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

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

maximize over all Q which are "ε - close"
to my samples $\hat{\mathbb{P}}$

Distributional robustification beyond LTI

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

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

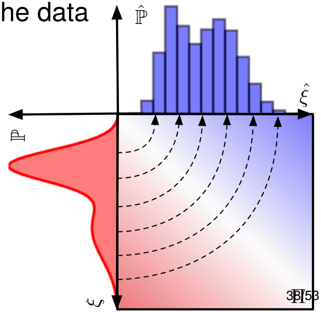
$\Rightarrow$ ***poor out-of-sample performance*** of above sample-average solution $x^\star$
for real problem: $\mathbb{E}_{\xi\sim\mathbb{P}}\big[c\,(\xi,x^\star)\big]$ where $\mathbb{P}$ is the *unknown distribution* of ξ

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

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

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

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



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

Theorem: $\underbrace{\inf_{x\in\mathcal{X}} \sup_{\mathbb{Q}\in\mathbb{B}_\epsilon(\widehat{\mathbb{P}})} \mathbb{E}_{\xi\sim\mathbb{Q}}\left[c\left(\xi,x\right)\right]}_{\text{distributional robust formulation}}$

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

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

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

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

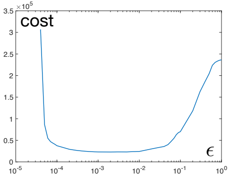
Cor : ℓ_∞-robustness in trajectory space
$\iff$ ℓ_1-regularization of DeePC

- ***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

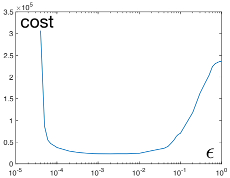
$\iff$ ℓ_1-regularization of DeePC



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

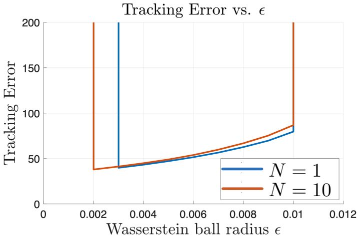
Theorem: $\underbrace{\inf_{x \in \mathcal{X}} \sup_{\mathbb{Q} \in \mathbb{B}_\epsilon(\widehat{\mathbb{P}})} \mathbb{E}_{\xi \sim \mathbb{Q}} \left[c\left(\xi, x\right) \right]}_{\text{distributional robust formulation}} \equiv \underbrace{\min_{x \in \mathcal{X}} c\left(\widehat{\xi}, x\right) + \epsilon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

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



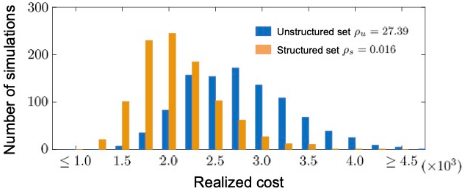
- ***measure concentration***: average matrix $\frac{1}{N}\sum_{i=1}^{N} \mathscr{H}_i(y^{\text{d}})$ from i.i.d. experiments

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



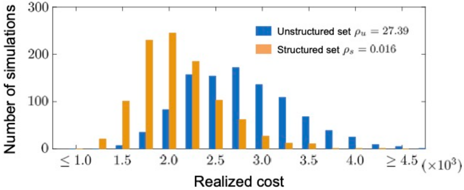
Further ingredients

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



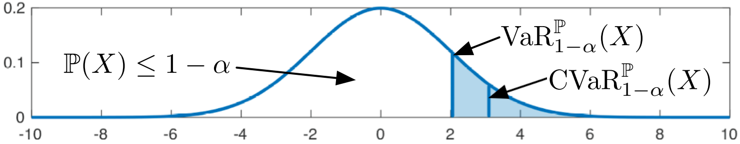
Further ingredients

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



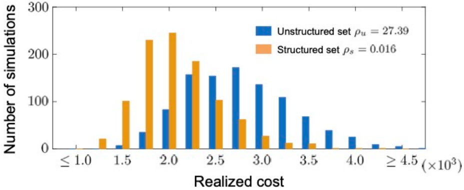
- ***distributionally robust probabilistic constraints***

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



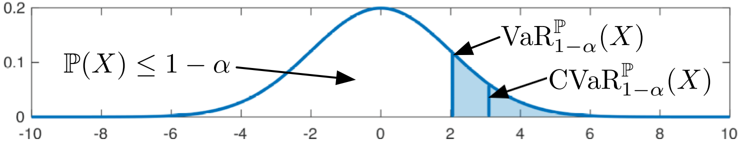
Further ingredients

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



- ***distributionally robust probabilistic constraints***

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



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

white elephant

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

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 !

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	model $\in$ argmin id cost (u^d, y^d)
	subject to model $\in$ LTI(n, ℓ) class

Comparison: direct vs. indirect control

indirect* **ID-based** *data-driven control

$$\begin{array}{ll} \text{minimize} & \text{control cost } (u, y) \\ \text{subject to} & (u, y) \text{ satisfy parametric model} \\ \text{where} & \left.\begin{array}{l} \text{model} \in \text{argmin id cost } (u^d, y^d) \\ \text{subject to model} \in \text{LTI}(n, \ell) \text{ class} \end{array}\right\} \text{ID} \end{array}$$

Comparison: direct vs. indirect control

indirect* ID-based *data-driven control

minimize control cost (u, y)

subject to (u, y) satisfy parametric model

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

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

$\left.\begin{array}{l} \\ \end{array}\right\}$ ID

ID projects data on the set of LTI models

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

Comparison: direct vs. indirect control

indirect* ID-based *data-driven control

minimize control cost (u, y)

subject to (u, y) satisfy parametric model

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

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

ID

ID projects data on the set of LTI models

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

direct* regularized *data-driven control

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

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

$$\begin{bmatrix} u \\ y \end{bmatrix} \in \operatorname{Im} H\begin{pmatrix} u^d \\ y^d \end{pmatrix}$$

Comparison: direct vs. indirect control

indirect* ID-based *data-driven control

minimize control cost (u, y)

subject to (u, y) satisfy parametric model

$$\left.\begin{array}{ll} \text{where} & \text{model} \in \text{argmin id cost } (u^d, y^d) \\ & \text{subject to model} \in \text{LTI}(n, \ell) \text{ class} \end{array}\right\} \text{ID}$$

ID projects data on the set of LTI models

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

direct* regularized *data-driven control

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

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

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

Comparison: direct vs. indirect control

indirect* ID-based *data-driven control

minimize control cost (u, y)

subject to (u, y) satisfy parametric model

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

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

ID

ID projects data on the set of LTI models

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

direct* regularized *data-driven control

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

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

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

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

Case study: direct vs. indirect control

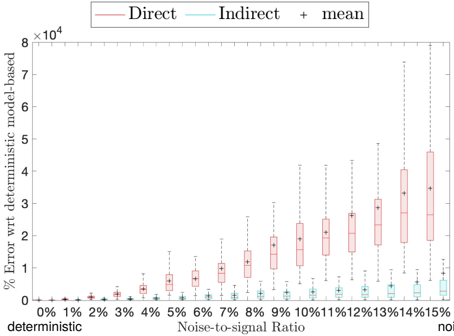
stochastic LTI case

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

Case study: direct vs. indirect control

stochastic LTI case

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

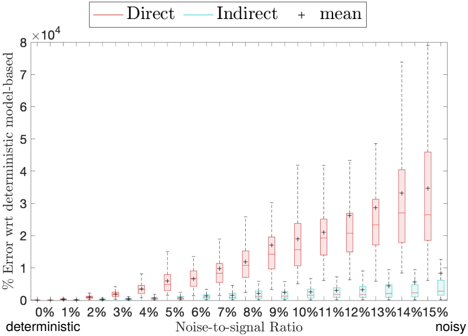


deterministic noisy

Case study: direct vs. indirect control

stochastic LTI case $\rightarrow$ indirect ID wins

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



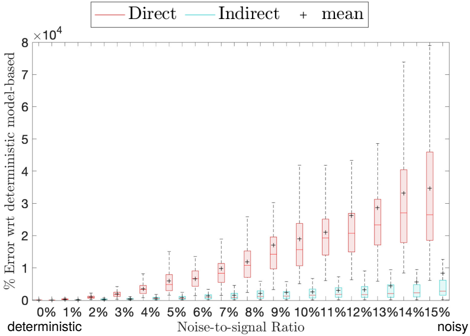
Case study: direct vs. indirect control

stochastic LTI case $\rightarrow$ indirect ID wins

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

nonlinear case

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



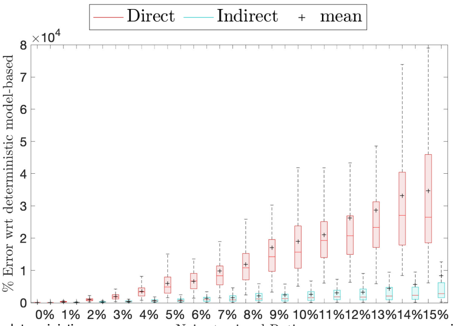
Case study: direct vs. indirect control

stochastic LTI case $\rightarrow$ indirect ID wins

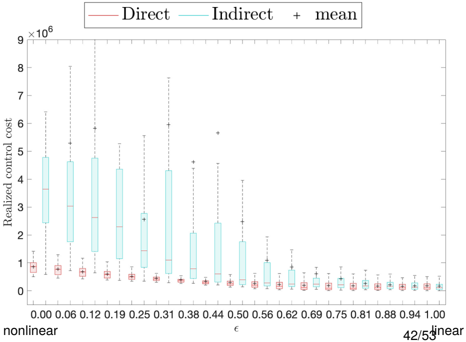
- LQR control of 5th order LTI system
- Gaussian noise with varying noise to signal ratio (100 rollouts each case)
- ℓ_1-regularized DeePC, SysID via N4SID, & judicious hyper-parameters

nonlinear case

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



deterministic noisy



nonlinear linear

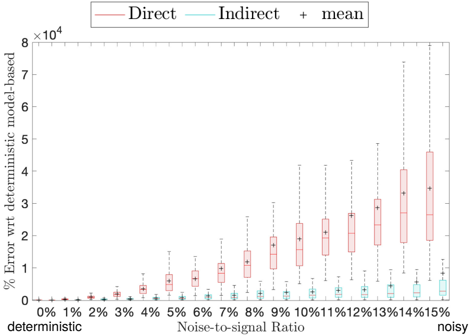
Case study: direct vs. indirect control

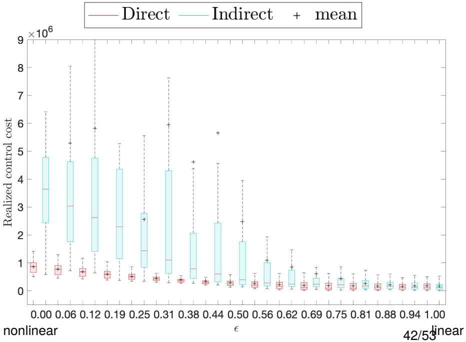
stochastic LTI case → indirect ID wins

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

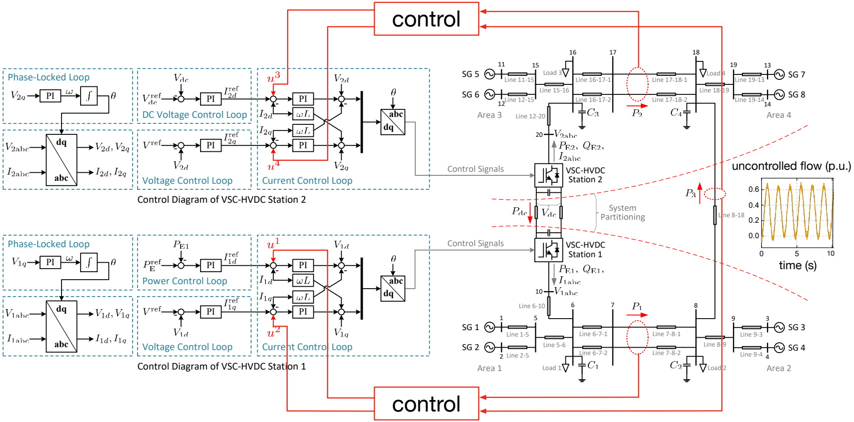
nonlinear case → direct DeePC wins

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



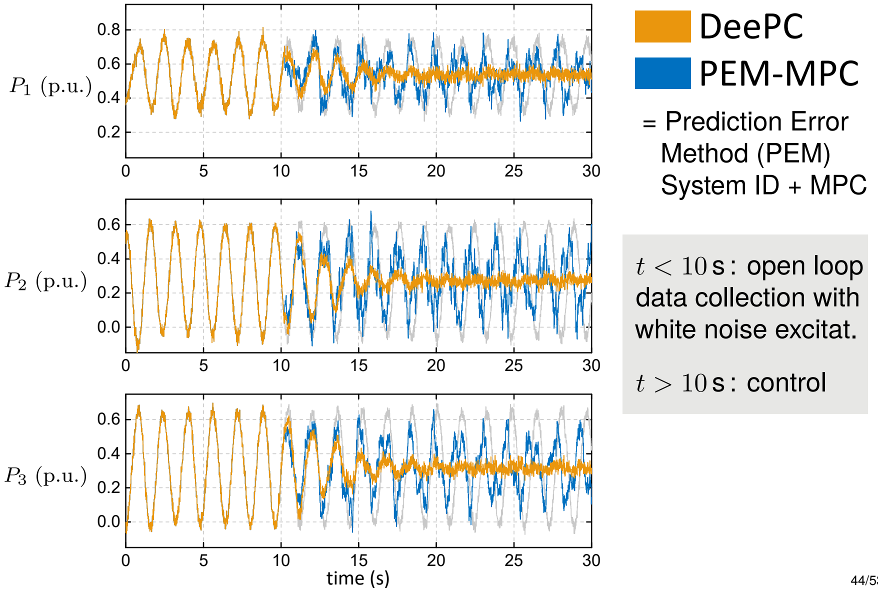


Power system case study revisited



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

Centralized control

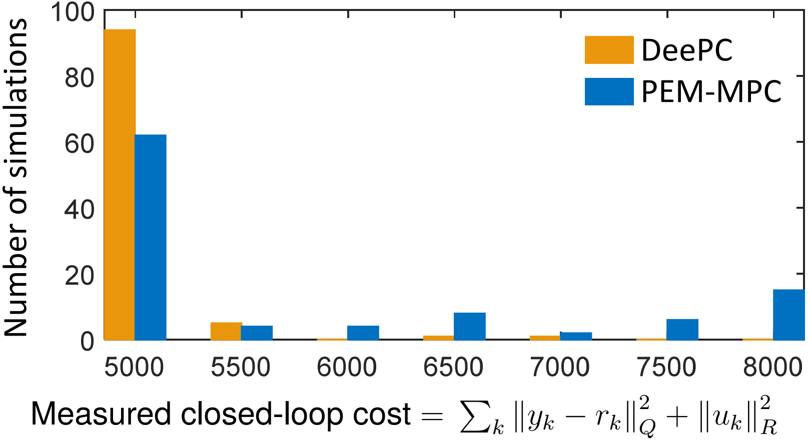


= Prediction Error Method (PEM) System ID + MPC

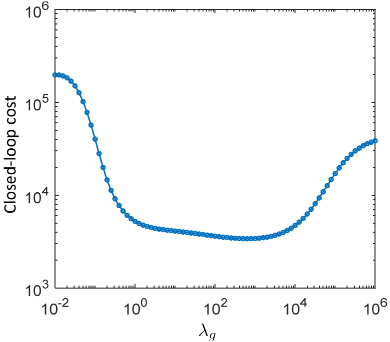
$t < 10\,$s: open loop data collection with white noise excitat.

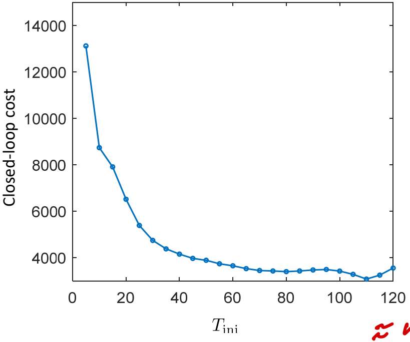
$t > 10\,$s: control

Performance: DeePC wins (clearly!)



DeePC hyper-parameter tuning





ℓ_1 ***regularizer*** λ_g

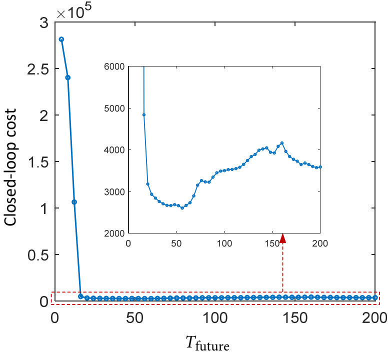
- for distributional robustness $\approx$ radius of Wasserstein ball
- wide range of sweet spots

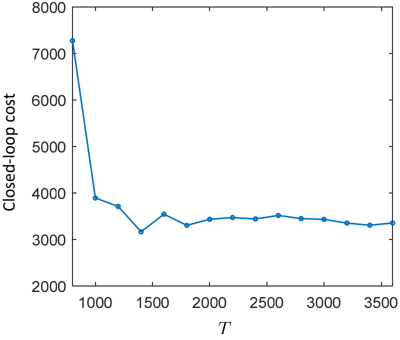
$\rightarrow$ choose $\lambda_g = 20$

estimation horizon T_{ini} $\approx n$

- 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

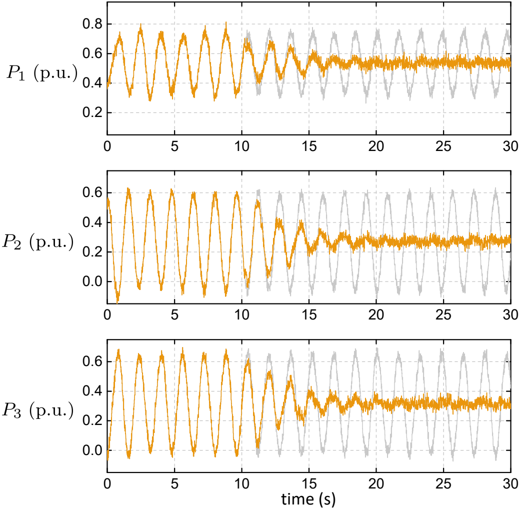
  $\rightarrow$ choose $T_{\text{future}} = 120$ & apply first 60 input steps

data length T

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

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

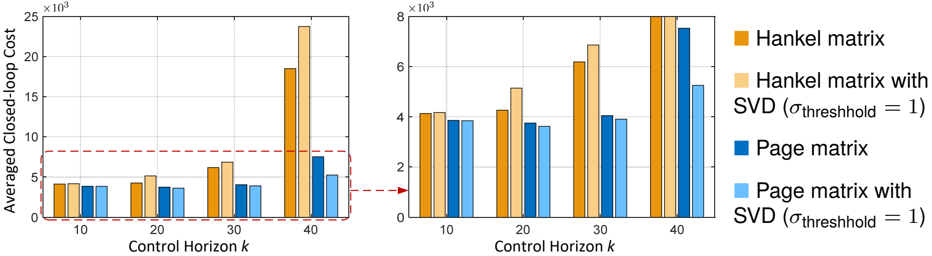
Computational cost



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

$\Rightarrow$ **implementable**

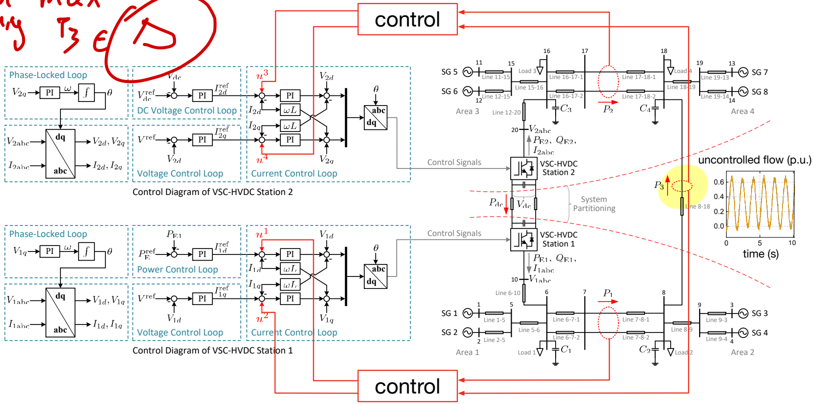
Comparison: Hankel & Page matrix



- 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

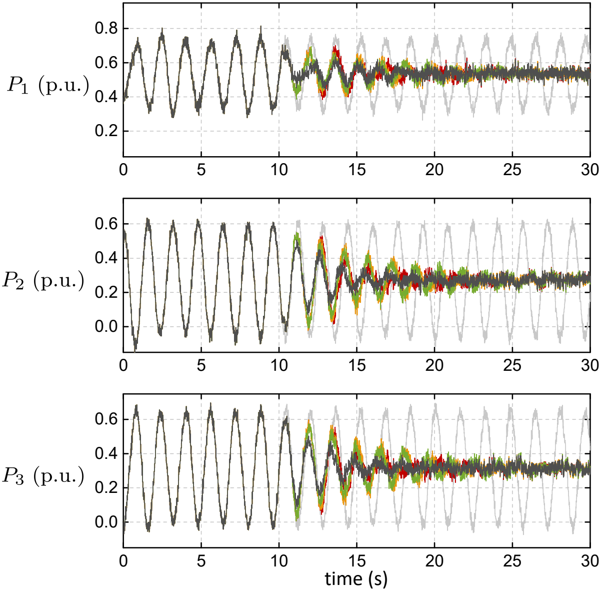
Decentralized implementation

min max
g,u,y $T_3 \in D$



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

Decentralized control performance



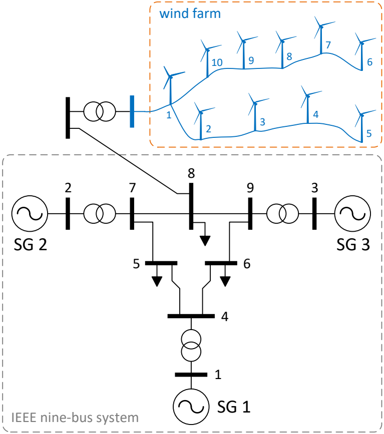
- colors correspond to different hyper-parameter settings (not discernible)
- ambiguity set $\mathcal{W}$ is ∞-ball (box)
- for computational efficiency $\mathcal{W}$ is downsampled (piece-wise linear)
- solver time ≈ 2.6 s

$\Rightarrow$ **implementable**

Conclusions

main take-aways

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



Conclusions

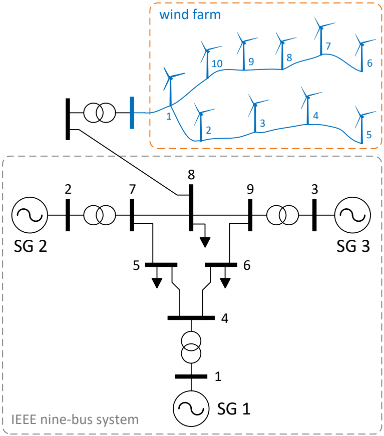
main take-aways

- 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



Conclusions

main take-aways

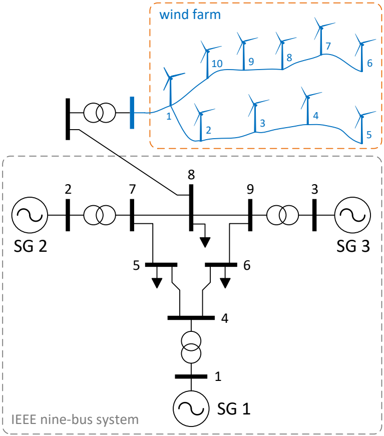
- matrix time series as predictive model
- robustness & side-info by regularization
- method that works in theory & practice
- focus is robust prediction not predictor ID

LQR

ongoing work

→ certificates for adaptive & nonlinear cases

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



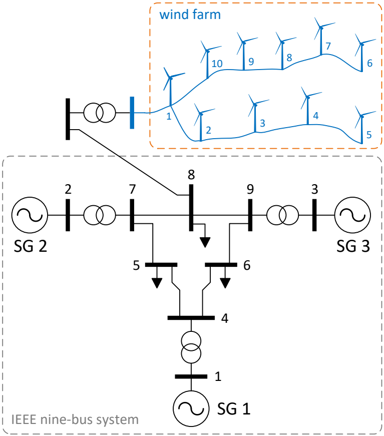
questions we should discuss

- catch? violate no-free-lunch theorem ? → more real-time computation

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

LQR

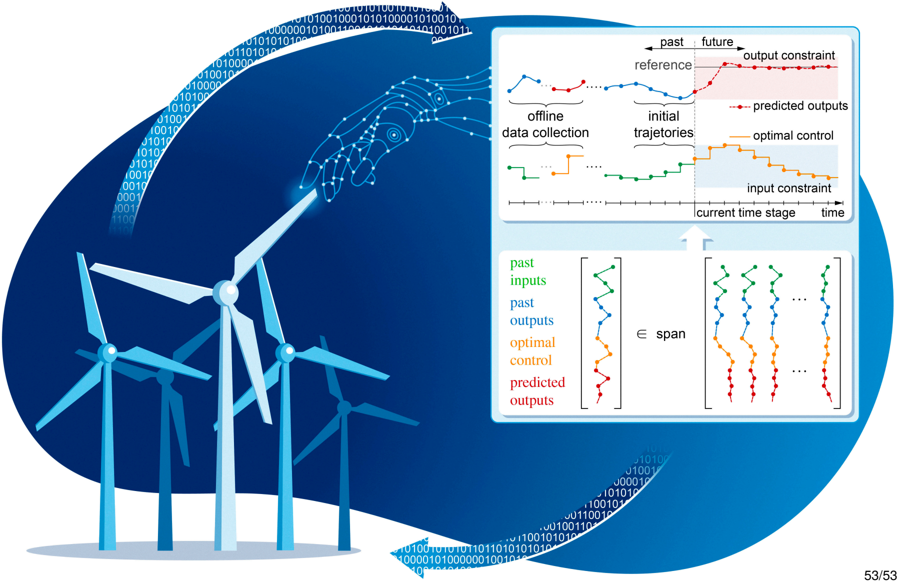
→ certificates for adaptive & nonlinear cases

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

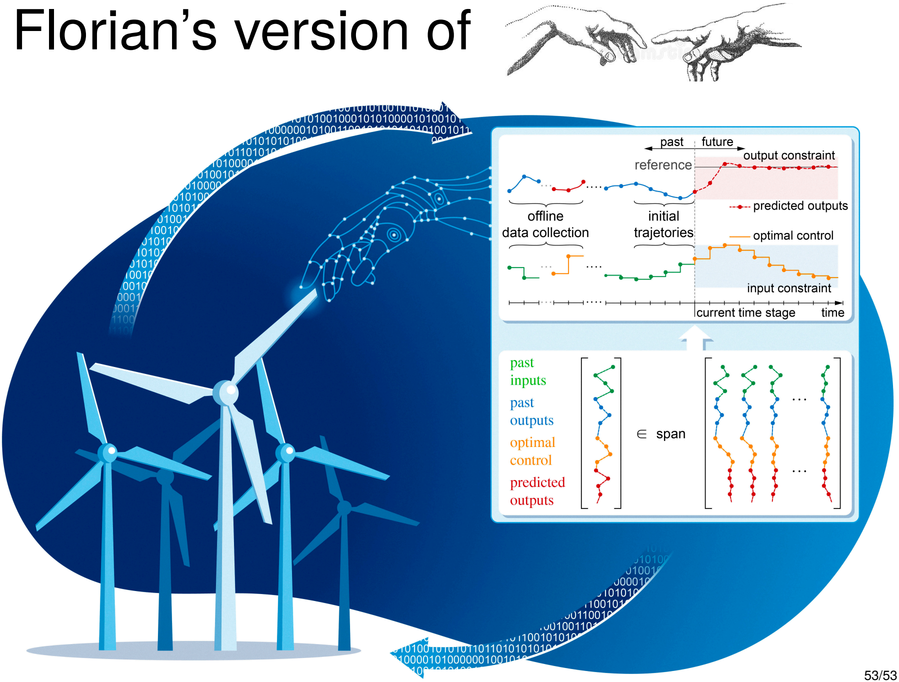
questions we should discuss

- catch? violate no-free-lunch theorem ? → more real-time computation
- when does direct beat indirect ? → Id4Control & bias/variance issues ?

control objective should bias ID


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

Florian's version of



Thanks !

Florian Dörfler

mail: dorfler@ethz.ch

[link] to homepage

[link] to related publications