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

Florian Dörfler

Automatic Control Laboratory, ETH Zürich

Acknowledgements

Jeremy Coulson

Linbin Huang

John Lygeros

Brain-storming: P. Mohajerin Esfahani, B. Recht, R. Smith, B. Bamieh, and M. Morari

Big, deep, intelligent and so on

- ***unprecedented availability*** of computation, storage, and data
- ***theoretical advances*** in optimization, statistics, and machine learning
- …and ***big-data*** frenzy

→ increasing importance of ***data-centric methods*** in all of science / engineering

Make up your own opinion, but machine learning works too well to be ignored.

From Pixels to Torques: Policy Learning with Deep Dynamical Models

Niklas Wahlström NIKWA@ISY.LIU.SE
Division of Automatic Control, Linköping University, Linköping, Sweden

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

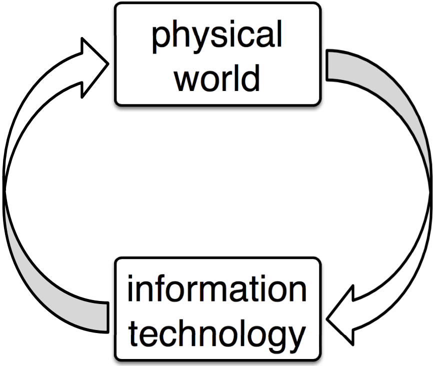
Marc Peter Deisenroth M.DEISENROTH@IMPERIAL.AC.UK
Department of Computing, Imperial College London, United Kingdom

Feedback – our central paradigm

actuation

"making a difference to the world"

automation and control



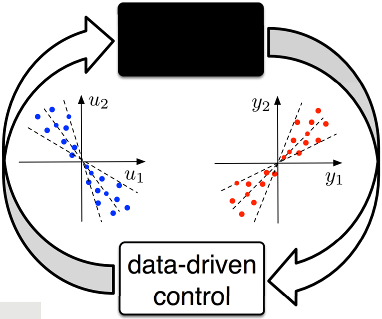
sensing

"making sense of the world"

inference and data science

Control in a data-rich world

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



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

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

- models are too complex to be useful (e.g., fluid dynamics & building automation)
- first-principle models are not conceivable (e.g., human-in-the-loop & perception)
- modeling & system ID is too cumbersome (e.g., robotics & power applications)

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

Example: PID

Snippets from the literature

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

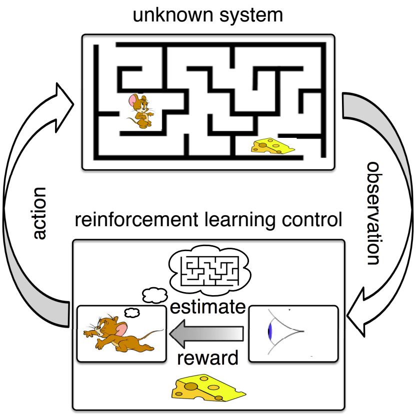
with key ***mathematical challenges***

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

and ***practical limitations***

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

ø suitable for physical control systems with real-time & safety constraints ?



A Tour of Reinforcement Learning
The View from Continuous Control

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

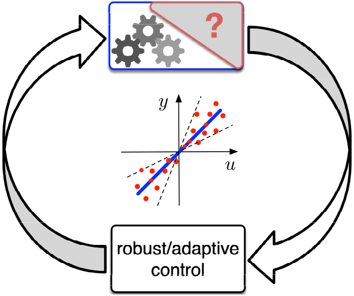
Snippets from the literature cont'd

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

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

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

ø limited applicability: need a-priori safety



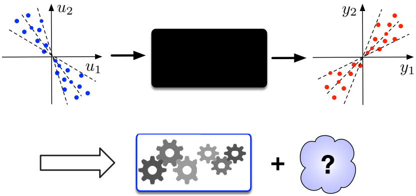
3. Sequential ***system ID + control***

- ID with uncertainty quantification followed by robust control design

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

ø ID seeks best but not most useful model

ø "easier to learn policies than models"



Key take-aways

- claim: ***easier to learn controllers*** from data rather than models
- data-driven approach is ***no silver bullet*** (see previous **ø**)
- ***predictive models are preferable over data*** (even approximate)

→ models are tidied-up, compressed, & de-noised representations

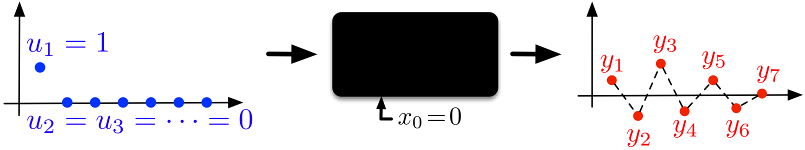
→ model-based methods vastly out-perform model-agnostic ones

ø deadlock ?

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

→ build a predictive & non-parametric model directly from raw data?

Colorful idea



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

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

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

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

Contents

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

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

II. From Heuristics & Numerical Promises to Theorems

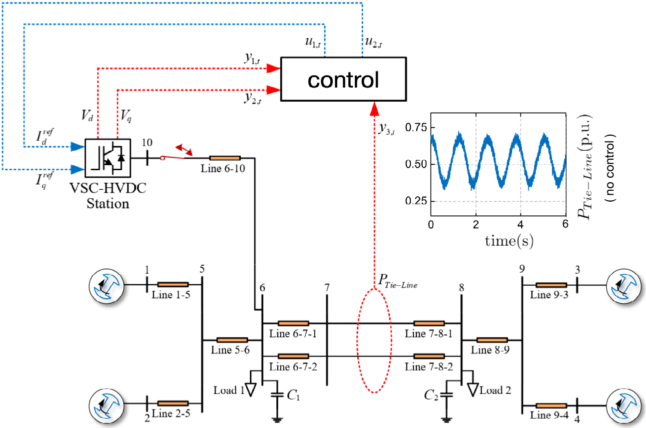
J. Coulson, J. Lygeros, and F. Dörfler. *Regularized and Distributionally Robust Data-Enabled Predictive Control*. arxiv.org/abs/1903.06804.

III. Application: End-to-End Automation in Energy Systems

L. Huang, J. Coulson, J. Lygeros, and F. Dörfler. *Data-Enabled Predictive Control for Grid-Connected Power Converters*. arxiv.org/abs/1903.07339.

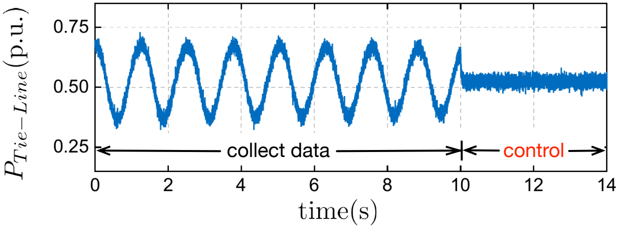
Preview

complex 2-area power ***system***: large ($n \approx 10^2$), nonlinear, noisy, stiff, & with input constraints



control objective:
damping of inter-area oscillations via HVDC

but without model



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

$\rightarrow$ automating ourselves

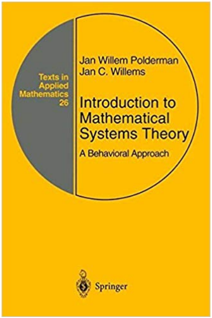
Behavioral view on LTI systems

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

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

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

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

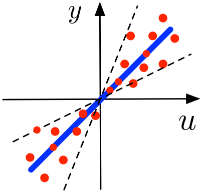


Definition: The dynamical system $(\mathbb{Z}_{\geq 0}, \mathbb{W}, \mathscr{B})$ is

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

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

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

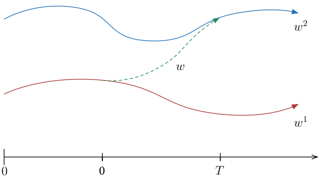


In the remainder we focus on ***discrete-time LTI systems***.

Behavioral view cont'd

$\mathscr{B}$ = ***set of trajectories*** in $\mathbb{W}^{\mathbb{Z}_{\geq 0}}$ & $\mathscr{B}_T$ is ***restriction*** to $t \in [0, T]$

A system $(\mathbb{Z}_{\geq 0}, \mathbb{W}, \mathscr{B})$ is ***controllable*** if any two trajectories w^1, $w^2 \in \mathscr{B}$ can be patched with a trajectory $w \in \mathscr{B}_T$.



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

$\rightarrow$ different parametric representations: state space, kernel, image, ...

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

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

LTI systems and matrix time series

foundation of state-space subspace system ID & signal recovery algorithms

$(u(t), y(t))$ satisfy recursive ***difference equation***

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

(ARMA / kernel representation)

$\Rightarrow$

$\Leftarrow$

under assumptions

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

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

The Fundamental Lemma

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

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

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

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

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

Cartoon of Fundamental Lemma

persistently exciting | controllable LTI | sufficiently many samples

$$
\underbrace{\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k + Du_k \end{aligned}}_{\text{parametric state-space model}}
\quad \Longleftrightarrow \quad
\underbrace{\text{colspan} \begin{bmatrix} \begin{pmatrix} u_1 \\ y_1 \end{pmatrix} & \begin{pmatrix} u_2 \\ y_2 \end{pmatrix} & \begin{pmatrix} u_3 \\ y_3 \end{pmatrix} & \cdots \\ \begin{pmatrix} u_2 \\ y_2 \end{pmatrix} & \begin{pmatrix} u_3 \\ y_3 \end{pmatrix} & \begin{pmatrix} u_4 \\ y_4 \end{pmatrix} & \cdots \\ \begin{pmatrix} u_3 \\ y_3 \end{pmatrix} & \begin{pmatrix} u_4 \\ y_4 \end{pmatrix} & \begin{pmatrix} u_5 \\ y_5 \end{pmatrix} & \cdots \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix}}_{\text{non-parametric model from raw data}}
$$

all trajectories constructible from finitely many previous trajectories

Data-driven simulation [Markovsky & Rapisarda '08]

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

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

Assume: $\mathscr{B}$ controllable & u^{d} persistently exciting of order $T_{\text{future}} + n$

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

$$\begin{bmatrix} u^{\text{d}}_1 & u^{\text{d}}_2 & \cdots & u^{\text{d}}_{T-N+1} \\ \vdots & \vdots & \ddots & \vdots \\ u^{\text{d}}_{T_{\text{future}}} & u^{\text{d}}_{T_{\text{future}}+1} & \cdots & u^{\text{d}}_T \\ \hline y^{\text{d}}_1 & y^{\text{d}}_2 & \cdots & y^{\text{d}}_{T-N+1} \\ \vdots & \vdots & \ddots & \vdots \\ y^{\text{d}}_{T_{\text{future}}} & y^{\text{d}}_{T_{\text{future}}+1} & \cdots & y^{\text{d}}_T \end{bmatrix} \boldsymbol{g} = \begin{bmatrix} u_1 \\ \vdots \\ u_{T_{\text{future}}} \\ \hline y_1 \\ \vdots \\ y_{T_{\text{future}}} \end{bmatrix}$$

Issue: predicted output is not unique $\rightarrow$ need to set initial conditions!

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

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

Assume: $\mathscr{B}$ controllable & u^{d} persist. exciting of order $T_{\text{ini}} + T_{\text{future}} + n$

Solution: given $(u_1, \dots, u_{T_{\text{future}}})$ & $\text{col}(u_{\text{ini}}, y_{\text{ini}})$

$\rightarrow$ compute $\boldsymbol{g}$ & $(y_1, \dots, y_{T_{\text{future}}})$ from

$$\begin{bmatrix} U_{\text{p}} \\ Y_{\text{p}} \\ U_{\text{f}} \\ Y_{\text{f}} \end{bmatrix} \boldsymbol{g} = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \\ y \end{bmatrix}$$

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

$$\begin{bmatrix} U_{\text{p}} \\ U_{\text{f}} \end{bmatrix} \triangleq \begin{bmatrix} u_1^{\text{d}} & \cdots & u_{T-T_{\text{future}}-T_{\text{ini}}+1}^{\text{d}} \\ \vdots & \ddots & \vdots \\ u_{T_{\text{ini}}}^{\text{d}} & \cdots & u_{T-T_{\text{future}}}^{\text{d}} \\ u_{T_{\text{ini}}+1}^{\text{d}} & \cdots & u_{T-T_{\text{future}}+1}^{\text{d}} \\ \vdots & \ddots & \vdots \\ u_{T_{\text{ini}}+T_{\text{future}}}^{\text{d}} & \cdots & u_T^{\text{d}} \end{bmatrix} \qquad \begin{bmatrix} Y_{\text{p}} \\ Y_{\text{f}} \end{bmatrix} \triangleq \begin{bmatrix} y_1^{\text{d}} & \cdots & y_{T-T_{\text{future}}-T_{\text{ini}}+1}^{\text{d}} \\ \vdots & \ddots & \vdots \\ y_{T_{\text{ini}}}^{\text{d}} & \cdots & y_{T-T_{\text{future}}}^{\text{d}} \\ y_{T_{\text{ini}}+1}^{\text{d}} & \cdots & y_{T-T_{\text{future}}+1}^{\text{d}} \\ \vdots & \ddots & \vdots \\ y_{T_{\text{ini}}+T_{\text{future}}}^{\text{d}} & \cdots & y_T^{\text{d}} \end{bmatrix}$$

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

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

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

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

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

model for **estimation** (many variations)

hard operational or safety **constraints**

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

Data-Enabled Predictive Control

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

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

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

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

hard operational or safety **constraints**

- Hankel matrix with $T_{\text{ini}} + T_{\text{future}}$ rows from past data $\begin{bmatrix} U_{\text{p}} \\ U_{\text{f}} \end{bmatrix} = \mathscr{H}_{T_{\text{ini}}+T_{\text{future}}}(u^{\text{d}})$ and $\begin{bmatrix} Y_{\text{p}} \\ Y_{\text{f}} \end{bmatrix} = \mathscr{H}_{T_{\text{ini}}+T_{\text{future}}}(y^{\text{d}})$

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

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

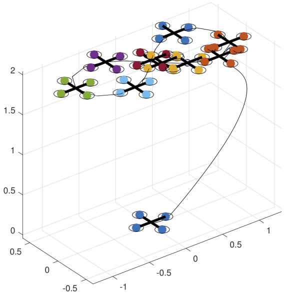
  updated **online**

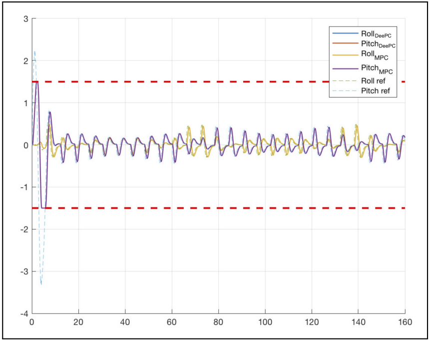
Correctness for LTI Systems

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

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

Aerial robotics case study :





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

. . . at least in the ***nominal case***.

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

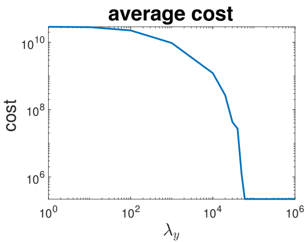
Noisy real-time measurements

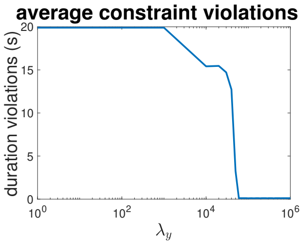
$$\begin{aligned} \underset{g,u,y}{\text{minimize}} \quad & \sum_{k=0}^{T_{\text{future}}-1} \|y_k - r_{t+k}\|_Q^2 + \|u_k\|_R^2 + \lambda_y \|\sigma_y\|_1 \\ \text{subject to} \quad & \begin{bmatrix} U_{\text{p}} \\ Y_{\text{p}} \\ U_{\text{f}} \\ Y_{\text{f}} \end{bmatrix} g = \begin{bmatrix} u_{\text{ini}}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

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

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

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





Hankel matrix corrupted by noise

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

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

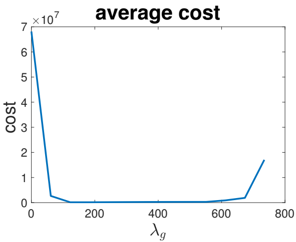
$$u_k \in \mathcal{U}, \quad \forall k \in \{0, \dots, T_{\text{future}} - 1\},$$

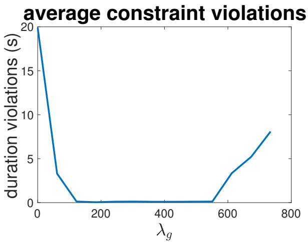
$$y_k \in \mathcal{Y}, \quad \forall k \in \{0, \dots, T_{\text{future}} - 1\}$$

Solution: add a ℓ_1-**penalty** on g

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

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





Towards nonlinear systems . . .

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

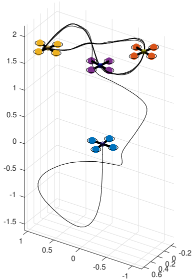
$\rightarrow$ Carleman, Volterra, Fliess, Koopman, Sturm-Liouville methods

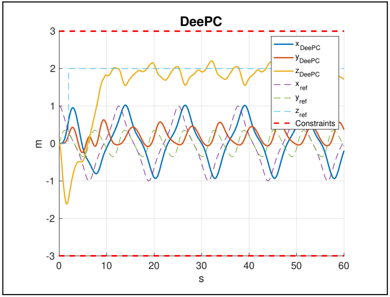
$\rightarrow$ ***exploit size rather than nonlinearity*** and find features in data

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

$\rightarrow$ ***regularization*** singles out relevant features / basis functions

case study : regularization for g and σ_y





recall the ***central promise***:

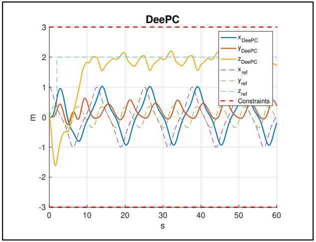
it is easier to learn control policies directly from data, rather than learning a model

Comparison to system ID + MPC

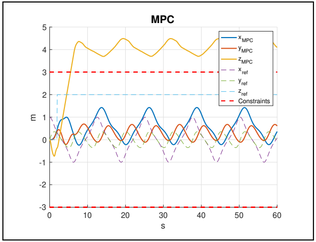
Setup : nonlinear stochastic quadcopter model with full state info

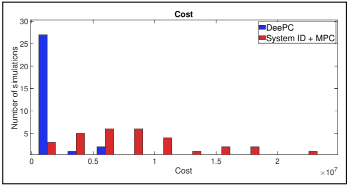
DeePC + ℓ_1-regularization for g and σ_y

MPC : system ID via prediction error method + nominal MPC

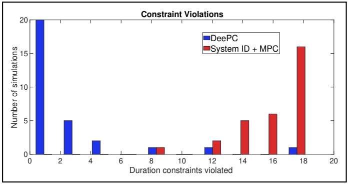


single
fig-8
run





random
sims



from heuristics &
numerical promises
to ***theorems***

Robust problem formulation

1. the ***nominal problem*** (without g-regularization)

$$\begin{aligned} \underset{g,u,y}{\text{minimize}} \quad & \sum_{k=0}^{T_{\text{future}}-1} \|y_k - r_{t+k}\|_Q^2 + \|u_k\|_R^2 + \lambda_y \|\sigma_y\|_1 \\ \text{subject to} \quad & \begin{bmatrix} \widehat{U_{\mathrm{p}}} \\ \widehat{Y_{\mathrm{p}}} \\ \widehat{U_{\mathrm{f}}} \\ \widehat{Y_{\mathrm{f}}} \end{bmatrix} g = \begin{bmatrix} u_{\text{ini}} \\ \widehat{y_{\text{ini}}} \\ u \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ \sigma_y \\ 0 \\ 0 \end{bmatrix}, \\ & u_k \in \mathcal{U}, \quad \forall k \in \{0, \ldots, T_{\text{future}} - 1\} \end{aligned}$$

where $\widehat{\cdot}$ denotes *measured* & thus possibly corrupted data

2. an ***abstraction*** of this problem $$\underset{g \in G}{\text{minimize}}\; f\left(\widehat{U_{\mathrm{f}}}g, \widehat{Y_{\mathrm{f}}}g\right) + \lambda_y \left\|\widehat{Y_{\mathrm{p}}}g - \widehat{y_{\text{ini}}}\right\|_1$$

where $G = \left\{g : \widehat{U_{\mathrm{p}}}g = u_{\text{ini}} \;\; \& \;\; \widehat{U_{\mathrm{f}}}g \in \mathcal{U}\right\}$

3. a ***further abstraction*** $\underset{g \in G}{\text{minimize}} \; c\left(\widehat{\xi}, g\right) = \underset{g \in G}{\text{minimize}} \; \mathbb{E}_{\widehat{\mathbb{P}}}\left[c\left(\xi, g\right)\right]$

with $G = \left\{ g : \widehat{U_{\mathrm{p}}} g = u_{\text{ini}} \;\&\; \widehat{U_{\mathrm{f}}} g \in \mathcal{U} \right\}$, *measured* $\widehat{\xi} = \left(\widehat{Y_{\mathrm{p}}}, \widehat{Y}_{\mathrm{f}}, \widehat{y_{\text{ini}}}\right)$,

& $\widehat{\mathbb{P}} = \delta_{\widehat{\xi}}$ denotes the *empirical distribution* from which we obtained $\widehat{\xi}$

4. the solution $g^\star$ of the above problem gives ***poor out-of-sample performance*** for the problem ***we really want to solve***: $\mathbb{E}_{\mathbb{P}}\left[c\left(\xi, g^\star\right)\right]$

where $\mathbb{P}$ is the *unknown* probability distribution of ξ

5. ***distributionally robust*** formulation $\inf_{g \in G} \sup_{Q \in \mathbb{B}_\epsilon(\widehat{P})} \mathbb{E}_Q\left[c\left(\xi, g\right)\right]$

where the *ambiguity set* $\mathbb{B}_\epsilon(\widehat{P})$ is an ***ϵ-Wasserstein ball centered at $\widehat{P}$***:

$$\mathbb{B}_\epsilon(\widehat{P}) = \left\{ P \,:\, \inf_{\Pi} \int \|\xi - \xi'\|_W \, d\Pi \leq \epsilon \right\}$$ where Π has marginals $\hat{P}$ and P

5. ***distributionally robust*** formulation $\inf_{g \in G} \sup_{Q \in \mathbb{B}_\epsilon(\widehat{P})} \mathbb{E}_Q \left[c\left(\xi, g\right)\right]$

where the *ambiguity set* $\mathbb{B}_\epsilon(\widehat{P})$ is an ***ϵ-Wasserstein ball centered at $\widehat{P}$***:

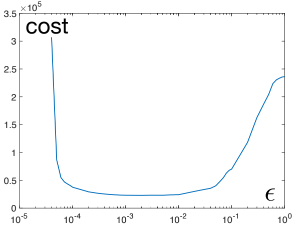
$$\mathbb{B}_\epsilon(\widehat{P}) = \left\{ P \,:\, \inf_{\Pi} \int \|\xi - \xi'\|_W \, d\Pi \,\leq\, \epsilon \right\}$$ where Π has marginals $\hat{P}$ and P

Theorem: Under minor technical conditions:

$$\inf_{g \in G} \sup_{Q \in \mathbb{B}_\epsilon(\widehat{P})} \mathbb{E}_Q \left[c\left(\xi, g\right)\right] \;\equiv\; \min_{g \in G} \, c\left(\widehat{\xi}, g\right) + \epsilon \lambda_y \|g\|_W^\star$$

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

Proof uses methods by Kuhn & Esfahani: semi-infinite problem becomes finite after marginalization & for discrete worst case



Relation to system ID & MPC

1. ***regularized DeePC*** problem

$$\begin{aligned} \underset{g,\, u \in \mathcal{U},\, y \in \mathcal{Y}}{\text{minimize}} \quad & f(u,y) + \lambda_g \|g\|_2^2 \\ \text{subject to} \quad & \begin{bmatrix} U_\mathrm{p} \\ Y_\mathrm{p} \\ U_\mathrm{f} \\ Y_\mathrm{f} \end{bmatrix} g = \begin{bmatrix} u_\mathsf{ini} \\ y_\mathsf{ini} \\ u \\ y \end{bmatrix} \end{aligned}$$

2. standard model-based ***MPC*** (ARMA parameterization)

$$\begin{aligned} \underset{u \in \mathcal{U},\, y \in \mathcal{Y}}{\text{minimize}} \quad & f(u,y) \\ \text{subject to} \quad & y = K \begin{bmatrix} u_\mathsf{ini} \\ y_\mathsf{ini} \\ u \end{bmatrix} \end{aligned}$$

3. ***subspace ID*** $y = Y_f g^\star$

where $g^\star = g^\star(u_\mathsf{ini}, y_\mathsf{ini}, u)$ solves

$$\begin{aligned} \underset{g}{\arg\min} \quad & \|g\|_2^2 \\ \text{subject to} \quad & \begin{bmatrix} U_\mathrm{p} \\ Y_\mathrm{p} \\ U_\mathrm{f} \end{bmatrix} g = \begin{bmatrix} u_\mathsf{ini} \\ y_\mathsf{ini} \\ u \end{bmatrix} \end{aligned}$$

4. equivalent ***prediction error ID***

$$\underset{K}{\text{minimize}} \quad \sum_j \left\| y_j^\mathsf{d} - K \begin{bmatrix} {u_\mathsf{ini}}_j^\mathsf{d} \\ {y_\mathsf{ini}}_j^\mathsf{d} \\ u_j^\mathsf{d} \end{bmatrix} \right\|^2$$

$$\rightarrow \quad y = K \begin{bmatrix} u_\mathsf{ini} \\ y_\mathsf{ini} \\ u \end{bmatrix} = Y_\mathrm{f}\, g^\star$$

subsequent ***ID & MPC***

$$\begin{aligned} \underset{u \in \mathcal{U}, y \in \mathcal{Y}}{\text{minimize}} \quad & f(u,y) \\ \text{subject to} \quad & y = K \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \end{bmatrix} \end{aligned}$$

where K solves

$$\underset{K}{\arg\min} \quad \sum_j \left\| y_j - K \begin{bmatrix} u_{\text{ini}\,j} \\ y_{\text{ini}\,j} \\ u_j \end{bmatrix} \right\|^2$$

$\equiv$

$$\begin{aligned} \underset{u \in \mathcal{U}, y \in \mathcal{Y}}{\text{minimize}} \quad & f(u,y) \\ \text{subject to} \quad & \begin{bmatrix} y \\ u \end{bmatrix} = \begin{bmatrix} Y_{\mathrm{f}} \\ U_{\mathrm{f}} \end{bmatrix} g \end{aligned}$$

where g solves

$$\begin{aligned} \underset{g}{\arg\min} \quad & \|g\|_2^2 \\ \text{subject to} \quad & \begin{bmatrix} U_{\mathrm{p}} \\ Y_{\mathrm{p}} \\ U_{\mathrm{f}} \end{bmatrix} g = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \end{bmatrix} \end{aligned}$$

regularized DeePC

$$\begin{aligned} \underset{g, u \in \mathcal{U}, y \in \mathcal{Y}}{\text{minimize}} \quad & f(u,y) + \lambda_g \|g\|_2^2 \\ \text{subject to} \quad & \begin{bmatrix} U_{\mathrm{p}} \\ Y_{\mathrm{p}} \\ U_{\mathrm{f}} \\ Y_{\mathrm{f}} \end{bmatrix} g = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \\ y \end{bmatrix} \end{aligned}$$

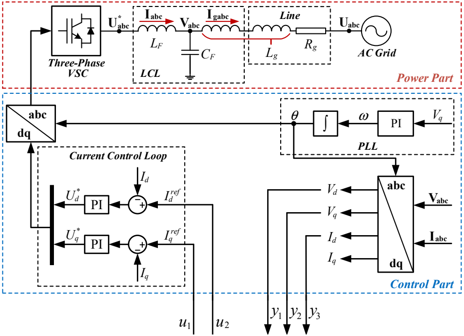
$\Rightarrow$ feasible set of ID & MPC
$\subseteq$ feasible set for DeePC

$\Rightarrow$ DeePC $\leq$ MPC + $\lambda_g \cdot$ ID

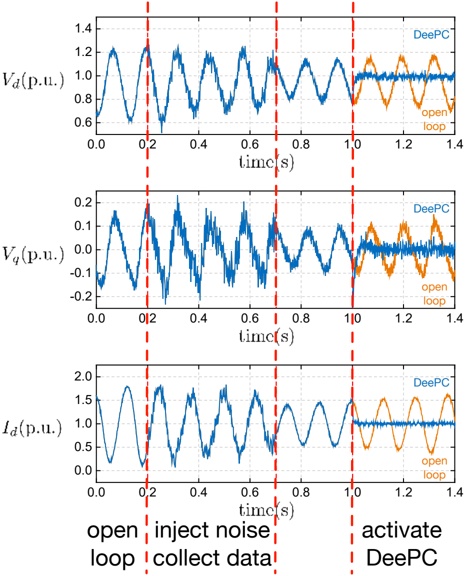
"easier to learn control policies from data rather than models" application: ***end-to-end automation*** in energy systems

Grid-connected converter control

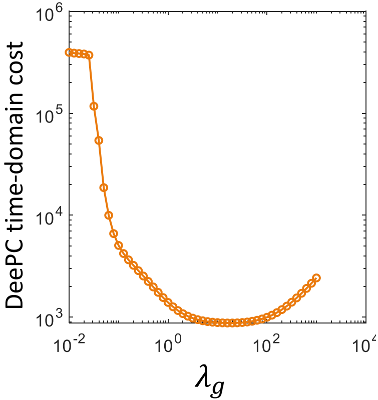
Task: control converter (nonlinear, noisy & constrained) without a model of the grid, line, passives, or inner loops

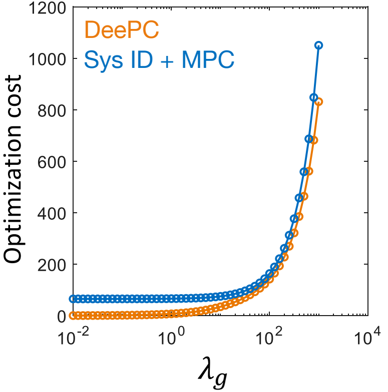


DeePC tracking constant dq-frame references subject to constraints



Effect of regularizations





DeePC time-domain cost

$= \sum_k \|y_k - r_k\|_Q^2 + \|u_k\|_R^2$

(closed-loop measurements)

Optimization cost

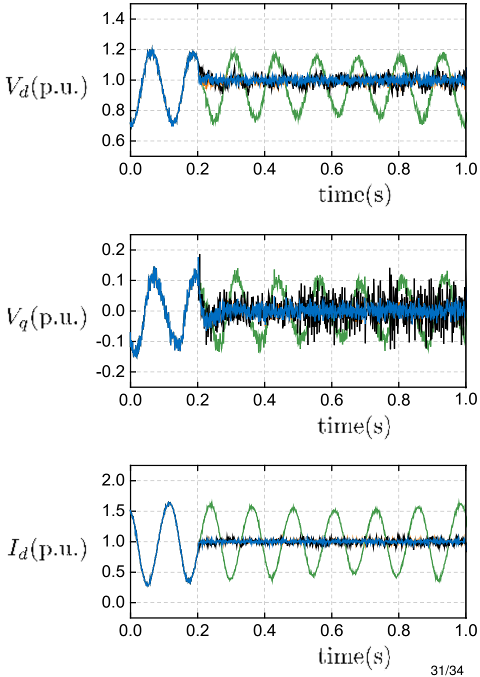
$= \sum_k \|y_k - r_k\|_Q^2 + \|u_k\|_R^2 + \lambda_g \|g\|^2$

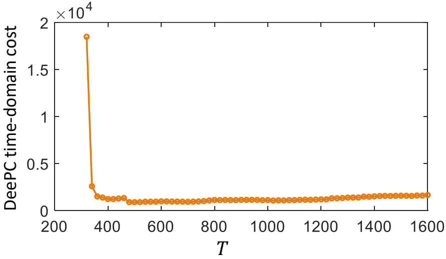
(closed-loop measurements)

Data length

$T_{\text{ini}} = 40$, $T_{\text{future}} = 30$

- Sys ID + MPC
- DeePC ($T = 500$)
- DeePC ($T = 330$)
- $I_d^{ref} = 1.0\text{p.u.}, I_q^{ref} = 0$





works like a charm for T large, ***but***

→ card$(g) = T - T_{\text{ini}} - T_{\text{future}} + 1$

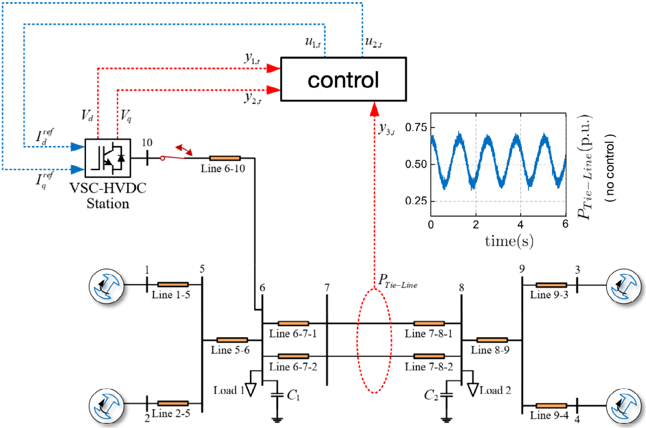
→ (possibly?) prohibitive on μDSP

Power system case study

extrapolation from previous case study: const. voltage $\to$ grid

complex 2-area power ***system***: large ($n \approx 10^2$), nonlinear, noisy, stiff, & with input constraints

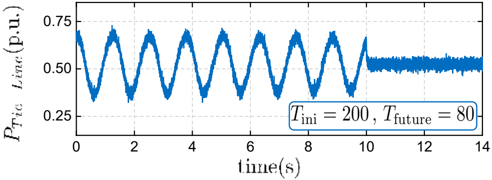
control objective: damping of inter-area oscillations via HVDC

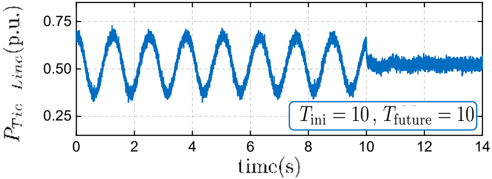


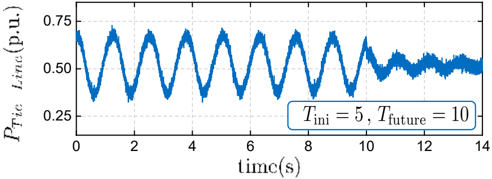
real-time closed-loop MPC & DeePC become prohibitive (on laptop)

$\to$ choose T, T_{ini}, and T_{future} wisely

Choice of time constants



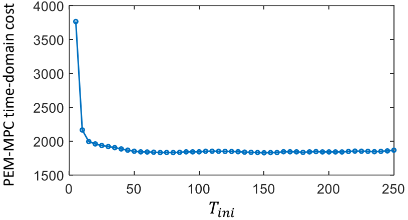




$\rightarrow$ choose T sufficiently large

$\rightarrow$ short horizon $T_{\text{future}} \approx 10$

$\rightarrow$ $T_{\text{ini}} \geq 10$ estimates sufficiently rich model complexity



time-domain cost

$$= \sum_k \|y_k - r_k\|_Q^2 + \|u_k\|_R^2$$

(closed-loop measurements)

Summary & conclusions

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

✓ certificates for deterministic LTI systems

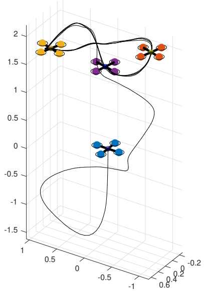
✓ distributional robustness via regularizations

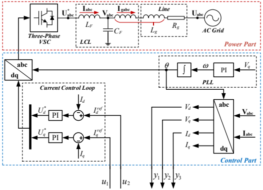
✓ outperforms ID + MPC in optimization metric

→ certificates for nonlinear & stochastic setup

→ adaptive extensions, explicit policies, ...

→ applications to building automation, bio, etc.





Why have these powerful ideas not been mixed long before ?

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

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