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

Florian Dörfler

Automatic Control Laboratory, ETH Zürich
Acknowledgements

Jeremy Coulson

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

Linbin Huang

John Lygeros
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  
Division of Automatic Control, Linköping University, Linköping, Sweden  
NIKWA@ISY.LIU.SE

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

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

---

End-to-End Deep Learning for Self-Driving Cars

By Mariusz Biagiński, Ben Firner, Beat Flepp, Larry Jackel, Urs Muller, Karol Zieba and Uardo Ust Treza | August 17, 2016

---

NVIDIA Developer Blog
Feedback – our central paradigm

actuation
“making a difference to the world”
automation and control

physical world

sensing
“making sense of the world”
inference and data science

information technology
Control in a data-rich world

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

\textbf{Q:} Why give up physical modeling and reliable model-based algorithms?

Data-driven control is \textit{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)

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

\textit{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** → conservative & complex control
   - **adaptive** → hard & asymptotic performance
   - **contemporary learning** algorithms (e.g., MPC + Gaussian processes / RL)
   → 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
   → 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?
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


II. From Heuristics & Numerical Promises to Theorems


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

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 → automating ourselves
Behavioral view on LTI systems

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

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

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

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

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

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

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

(iii) *complete* if \(\mathcal{B}\) is closed \(\iff\) \(\mathcal{W}\) is finite dimensional.

In the remainder we focus on *discrete-time LTI systems.*
Behavioral view cont’d

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

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

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

→ different parametric representations: state space, kernel, image, ...

→ \textit{kernel representation} (ARMA): \(\mathcal{B} = \text{col}(u, y) \in (\mathbb{R}^{m+p})^{Z \geq 0} \text{ s.t.} \)

\[ b_0 u + b_1 \sigma u + \cdots + b_n \sigma^n u + a_0 y + a_1 \sigma y + \cdots + a_n \sigma^n y = 0 \]
LTI systems and matrix time series

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

\[(u(t), y(t))\] satisfy recursive

\[\text{difference equation}\]

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

(ARMA / kernel representation)

\[
[ b_0 \ a_0 \ b_1 \ a_1 \ \ldots \ b_n \ a_n ] \text{ spans left nullspace of } \textbf{Hankel matrix} \text{ (collected from data)}
\]

\[
\mathcal{H}_L (u \ y) = \\
\begin{pmatrix}
(u_1) & (u_2) & (u_3) & \ldots & (u_{T-L+1}) \\
(y_1) & (y_2) & (y_3) & \ldots & (y_{T-L+1}) \\
(u_2) & (u_3) & (u_4) & \ldots & \vdots \\
(y_2) & (y_3) & (y_4) & \ldots & \vdots \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
(u_L) & \ldots & \ldots & \ldots & (u_T) \\
(y_L) & \ldots & \ldots & \ldots & (y_T)
\end{pmatrix}
\]

\[\leftarrow \text{ under assumptions} \]
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 
\[
\mathcal{H}_L(u) = \begin{bmatrix}
u_1 & \cdots & u_{T-L+1} \\
\vdots & \ddots & \vdots \\
u_L & \cdots & u_T
\end{bmatrix}
\]
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 \((\mathbb{Z}_{\geq0}, \mathbb{R}^{m+p}, \mathcal{B})\), and
- a \( T \)-sample long **trajectory** \( \text{col}(u^d, y^d) \in \mathcal{B}_T \), where
- \( u \) is **persistently exciting** of order \( t + n \) (prediction span + # states).

Then \[
\text{colspan} \left( \mathcal{H}_t \left( \begin{bmatrix} u \\ y \end{bmatrix} \right) \right) = \mathcal{B}_t .
\]
Cartoon of Fundamental Lemma

Persistently exciting \rightarrow \text{controllable LTI} \rightarrow \text{sufficiently many samples}

\[
x_{k+1} = Ax_k + Bu_k \\
y_k = Cx_k + Du_k
\]

Parametric state-space model \leftrightarrow \text{colspan}

\[
\begin{bmatrix}
  u_1 \\
y_1 \\
u_2 \\
y_2 \\
u_3 \\
y_3 \\
\vdots
\end{bmatrix} \quad \begin{bmatrix}
  u_2 \\
y_2 \\
u_3 \\
y_4 \\
u_4 \\
y_5 \\
\vdots
\end{bmatrix} \quad \begin{bmatrix}
  u_3 \\
y_3 \\
u_4 \\
y_4 \\
u_5 \\
y_5 \\
\vdots
\end{bmatrix} \quad \ldots
\]

Non-parametric model from raw data

All trajectories constructible from finitely many previous trajectories
# Data-driven simulation

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

- input signal \( u \in \mathbb{R}^{m \cdot T_{\text{future}}} \)
- past data \( \text{col}(u^d, y^d) \in \mathcal{B}_{T_{\text{data}}} \)

**Assume**: \( \mathcal{B} \) controllable & \( u^d \) persistently exciting of order \( T_{\text{future}} + n \)

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

\[
\begin{bmatrix}
  u^d_1 & u^d_2 & \cdots & u^d_{T-N+1} \\
  \vdots & \vdots & \ddots & \vdots \\
  u^d_{T_{\text{future}}} & u^d_{T_{\text{future}}+1} & \cdots & u^d_T \\
  y^d_1 & y^d_2 & \cdots & y^d_{T-N+1} \\
  \vdots & \vdots & \ddots & \vdots \\
  y^d_{T_{\text{future}}} & y^d_{T_{\text{future}}+1} & \cdots & y^d_T \\
\end{bmatrix}
\]

\( g = \begin{bmatrix}
  u_1 \\
  \vdots \\
  u_{T_{\text{future}}} \\
  y_1 \\
  \vdots \\
  y_{T_{\text{future}}} \\
\end{bmatrix} \)

**Issue**: predicted output is not unique → 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 \( x_{\text{ini}} \)

- input signal \( u \in \mathbb{R}^{m \cdot T_{\text{future}}} \) \( \rightarrow \) to predict forward

- past data \( \text{col}(u^d, y^d) \in \mathcal{B}_{T_{\text{data}}} \) \( \rightarrow \) to form Hankel matrix

**Assume**: \( \mathcal{B} \) controllable & \( u^d \) persist. exciting of order \( T_{\text{ini}} + T_{\text{future}} + n \)

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

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

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

\[
\begin{bmatrix}
U_p \\
Y_p \\
U_f \\
Y_f
\end{bmatrix} \triangleq \begin{bmatrix}
u^d_1 & \cdots & u^d_{T-T_{\text{future}}-T_{\text{ini}}+1} \\
\vdots & \ddots & \vdots \\
u^d_{T_{\text{ini}}} & \cdots & u^d_{T-T_{\text{future}}} \\
u^d_{T_{\text{ini}}+1} & \cdots & u^d_{T-T_{\text{future}}+1} \\
\vdots & \ddots & \vdots \\
u^d_{T_{\text{ini}}+T_{\text{future}}} & \cdots & u^d_T
\end{bmatrix}
\]

\[
\begin{bmatrix}
y^d_1 \\
\vdots \\
y^d_{T_{\text{ini}}} \\
y^d_{T_{\text{ini}}+1} \\
\vdots \\
y^d_{T_{\text{ini}}+T_{\text{future}}} \\
y^d_T
\end{bmatrix} \triangleq \begin{bmatrix}
y_1 \\
\vdots \\
y_{T_{\text{ini}}} \\
y_{T_{\text{ini}}+1} \\
\vdots \\
y_{T_{T_{\text{future}}-T_{\text{ini}}+1}} \\
y_T
\end{bmatrix}
\]
Output Model Predictive Control

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

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

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:

\[
\begin{align*}
\text{minimize} & \quad g, u, y \\
\text{subject to} & \quad \sum_{k=0}^{T_{\text{future}}-1} \left\| y_k - r_{t+k} \right\|_Q^2 + \left\| u_k \right\|_R^2 \\
& \quad \left[ \begin{array}{c} U_p \\ Y_p \\ U_f \\ Y_f \end{array} \right] g = \left[ \begin{array}{c} u_{\text{ini}} \\ y_{\text{ini}} \\ u \\ y \end{array} \right], \\
& \quad u_k \in \mathcal{U}, \quad \forall k \in \{0, \ldots, T_{\text{future}} - 1\}, \\
& \quad y_k \in \mathcal{Y}, \quad \forall k \in \{0, \ldots, T_{\text{future}} - 1\}
\end{align*}
\]

- **Hankel matrix with** \( T_{\text{ini}} + T_{\text{future}} \) **rows from past data**
  \[
  \begin{bmatrix} U_p \\ U_f \end{bmatrix} = \mathcal{H}_{T_{\text{ini}}+T_{\text{future}}} (u^d) \quad \text{and} \quad \begin{bmatrix} Y_p \\ Y_f \end{bmatrix} = \mathcal{H}_{T_{\text{ini}}+T_{\text{future}}} (y^d)
  \]

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

**quadratic cost** with \( R \succ 0, Q \succeq 0 \) & ref. \( r \)

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

**hard operational or safety constraints**

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

**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

\[
\begin{align*}
\text{minimize} & \quad g, u, y \\
\quad & \quad \sum_{k=0}^{T_{\text{future}}-1} \| y_k - r_{t+k} \|^2_Q + \| u_k \|^2_R + \lambda_y \| \sigma_y \|_1 \\
\text{subject to} & \quad \begin{bmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{bmatrix} g = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ \sigma_y \\ 0 \\ 0 \end{bmatrix}, \\
& \quad u_k \in \mathcal{U}, \quad \forall k \in \{0, \ldots, T_{\text{future}} - 1\}, \\
& \quad y_k \in \mathcal{Y}, \quad \forall k \in \{0, \ldots, T_{\text{future}} - 1\}.
\end{align*}
\]

**Solution:** add slack to ensure feasibility with \( \ell_1 \)-penalty

\[ \Rightarrow \text{for } \lambda_y \text{ sufficiently large } \sigma_y \neq 0 \text{ only if constraint infeasible} \]

c.f. *sensitivity analysis* over randomized sims
Hankel matrix corrupted by noise

\[
\begin{align*}
\text{minimize} & \quad g, u, y \\
\text{subject to} & \quad \sum_{k=0}^{T_{\text{future}}-1} \|y_k - r_{t+k}\|^2_Q + \|u_k\|^2_R + \lambda_g \|g\|_1 \\
& \quad \begin{bmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{bmatrix} g = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \\ y \end{bmatrix}, \\
& \quad u_k \in \mathcal{U}, \quad \forall k \in \{0, \ldots, T_{\text{future}} - 1\}, \\
& \quad y_k \in \mathcal{Y}, \quad \forall k \in \{0, \ldots, T_{\text{future}} - 1\},
\end{align*}
\]

**Solution:** add a \( \ell_1 \)-penalty on \( g \)

**intuition:** \( \ell_1 \) sparsely selects

\{Hankel matrix columns\} = \{past trajectories\} = \{motion primitives\}

c.f. sensitivity analysis over randomized sims

![Graph 1: Average cost vs. \( \lambda_g \)]

![Graph 2: Average constraint violations vs. \( \lambda_g \)]
Towards nonlinear systems . . .

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

→ exploit size, collect more data, & build a **larger Hankel matrix**
→ **regularization** singles out relevant features / basis functions

case study: regularization for $g$ and $\sigma_y$
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**: $\ell_1$-regularization for $g$ and $\sigma_y$

**MPC**: system ID via prediction error method + nominal MPC

![Graphs showing comparisons between DeePC and System ID + MPC](image-url)
from heuristics & numerical promises to *theorems*
Robust problem formulation

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

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

   subject to
   \[
   \begin{bmatrix}
   \hat{U}_p \\
   \hat{Y}_p \\
   \hat{U}_f \\
   \hat{Y}_f
   \end{bmatrix} \quad g = \begin{bmatrix}
   u_{\text{ini}} \\
   y_{\text{ini}} \\
   u \\
   y
   \end{bmatrix} + \begin{bmatrix}
   0 \\
   \sigma_y \\
   0 \\
   0
   \end{bmatrix},
   \]

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

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

2. an **abstraction** of this problem

   minimize
   \[
   f \left( \hat{U}_fg, \hat{Y}_fg \right) + \lambda_y \left\| \hat{Y}_pg - y_{\text{ini}} \right\|_1
   \]

   where $G = \left\{ g : \hat{U}_pg = u_{\text{ini}} \land \hat{U}_fg \in \mathcal{U} \right\}$
3. a **further abstraction**

\[
\text{minimize} \quad g \in G \quad c \left( \hat{\xi}, g \right) = \text{minimize} \quad g \in G' \quad \mathbb{E}_{\hat{P}} \left[ c \left( \xi, g \right) \right]
\]

with \( G = \left\{ g : \hat{U}_p g = u_{\text{ini}} \quad \& \quad \hat{U}_f g \in \mathcal{U} \right\} \), measured \( \hat{\xi} = \left( \hat{Y}_p, \hat{Y}_f, \hat{y}_{\text{ini}} \right) \),

\& \( \hat{P} = \delta_{\hat{\xi}} \) denotes the **empirical distribution** from which we obtained \( \hat{\xi} \)

4. the solution \( g^* \) of the above problem gives **poor out-of-sample performance** for the problem we really want to solve:

\[
\mathbb{E}_P \left[ c \left( \xi, g^* \right) \right]
\]

where \( P \) is the **unknown** probability distribution of \( \xi \)

5. **distributionally robust** formulation

\[
\inf_{g \in G} \sup_{Q \in \mathcal{B}_\epsilon (\hat{P})} \mathbb{E}_Q \left[ c \left( \xi, g \right) \right]
\]

where the **ambiguity set** \( \mathcal{B}_\epsilon (\hat{P}) \) is an **\( \epsilon \)-Wasserstein ball centered at \( \hat{P} \)**:

\[
\mathcal{B}_\epsilon (\hat{P}) = \left\{ P : \inf_{\Pi} \int \| \xi - \xi' \|_W d\Pi \leq \epsilon \right\}
\]

where \( \Pi \) has marginals \( \hat{P} \) and \( P \)
5. **distributionally robust** formulation

\[
\inf_{g \in G} \sup_{Q \in \mathcal{B}_\epsilon(\hat{P})} \mathbb{E}_Q [c(\xi, g)]
\]

where the *ambiguity set* \( \mathcal{B}_\epsilon(\hat{P}) \) is an \( \epsilon \)-Wasserstein ball centered at \( \hat{P} \):

\[
\mathcal{B}_\epsilon(\hat{P}) = \left\{ P : \inf_{\Pi} \int \|\xi - \xi'\|_W d\Pi \leq \epsilon \right\}
\]

where \( \Pi \) has marginals \( \hat{P} \) and \( P \)

**Theorem** : Under minor technical conditions:

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

**Cor** : \( \ell_\infty \)-robustness in trajectory space \( \iff \ell_1 \)-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

\[
\begin{align*}
\text{minimize} & \quad f(u, y) + \lambda_g \|g\|_2^2 \\
\text{subject to} & \quad g = \begin{bmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{bmatrix} \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \\ y \end{bmatrix}
\end{align*}
\]

2. standard model-based MPC (ARMA parameterization)

\[
\begin{align*}
\text{minimize} & \quad f(u, y) \\
\text{subject to} & \quad y = K \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \end{bmatrix}
\end{align*}
\]

3. subspace ID

\[y = Y_f g^*\]

where \(g^* = g^*(u_{\text{ini}}, y_{\text{ini}}, u)\) solves

\[
\begin{align*}
\text{arg min} & \quad \|g\|_2^2 \\
\text{subject to} & \quad g = \begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix} \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \end{bmatrix}
\end{align*}
\]

4. equivalent prediction error ID

\[
\begin{align*}
\text{minimize} & \quad \sum_j \left\| y_j^d - K \begin{bmatrix} u_{\text{ini},j}^d \\ y_{\text{ini},j}^d \\ u_j^d \end{bmatrix} \right\|^2 \\
\rightarrow & \quad y = K \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \end{bmatrix} = Y_f g^*
\end{align*}
\]
subsequent ID & MPC

\[
\begin{align*}
\text{minimize} & \quad f(u, y) \\
u \in \mathcal{U}, y \in \mathcal{Y} & \quad \text{subject to} \quad y = K \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \end{bmatrix}
\end{align*}
\]

where \( K \) solves

\[
\arg \min_K \sum_j \left\| y_j - K \begin{bmatrix} u_{\text{ini}, j} \\ y_{\text{ini}, j} \\ u_j \end{bmatrix} \right\|^2
\]

**regularized DeePC**

\[
\begin{align*}
\text{minimize} & \quad f(u, y) + \lambda_g \| g \|^2_2 \\
g, u \in \mathcal{U}, y \in \mathcal{Y} & \quad \text{subject to} \quad \begin{bmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{bmatrix} g = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \\ y \end{bmatrix}
\end{align*}
\]

\[ \Rightarrow \text{feasible set of ID & MPC} \subseteq \text{feasible set for DeePC} \]

\[ \Rightarrow \text{DeePC} \leq \text{MPC} + \lambda_g \cdot \text{ID} \]

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

\[ = \sum_k \| y_k - r_k \|_Q^2 + \| u_k \|_R^2 + \lambda_g \| g \|_2^2 \]

(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

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

\(\rightarrow\) (possibly?) prohibitive on \(\mu\)DSP
Power system case study

**extrapolation** from previous case study: const. voltage → 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)

→ choose \(T\), \(T_{\text{ini}}\), and \(T_{\text{future}}\) wisely
Choice of time constants

$T_{\text{ini}} = 5$, $T_{\text{future}} = 10$

$T_{\text{ini}} = 10$, $T_{\text{future}} = 10$

$T_{\text{ini}} = 200$, $T_{\text{future}} = 80$

→ choose $T$ sufficiently large
→ short horizon $T_{\text{future}} \approx 10$
→ $T_{\text{ini}} \geq 10$ estimates sufficiently rich model complexity

PEM-MPC time-domain cost

$$T_{\text{ini}}$$

DeePC time-domain cost

$$T_{\text{ini}}$$

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”