
# Data-enabled Predictive Control

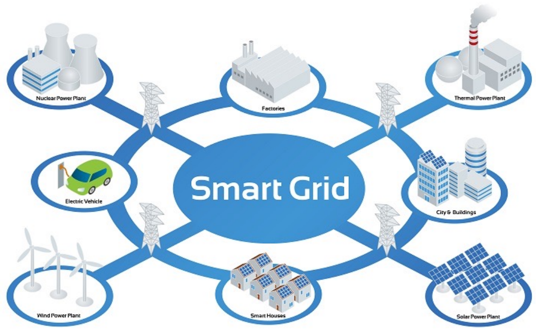
## Regularization and Robustness

Jeremy Coulson (joint work with Florian Dörfler & John Lygeros)

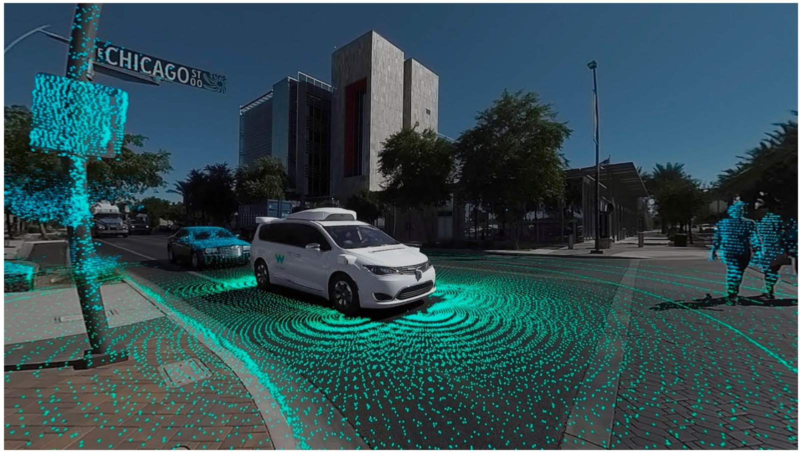
2024 Midwest Workshop on Control and Game Theory

# Data in today's world


Artificial intelligence

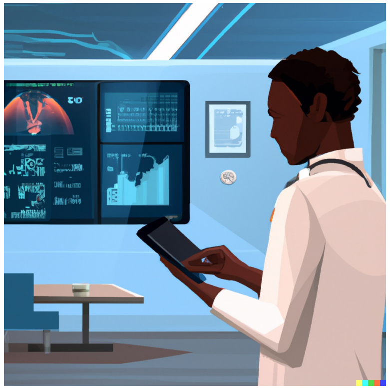



Smart infrastructure




Smart transportation

Smart agriculture

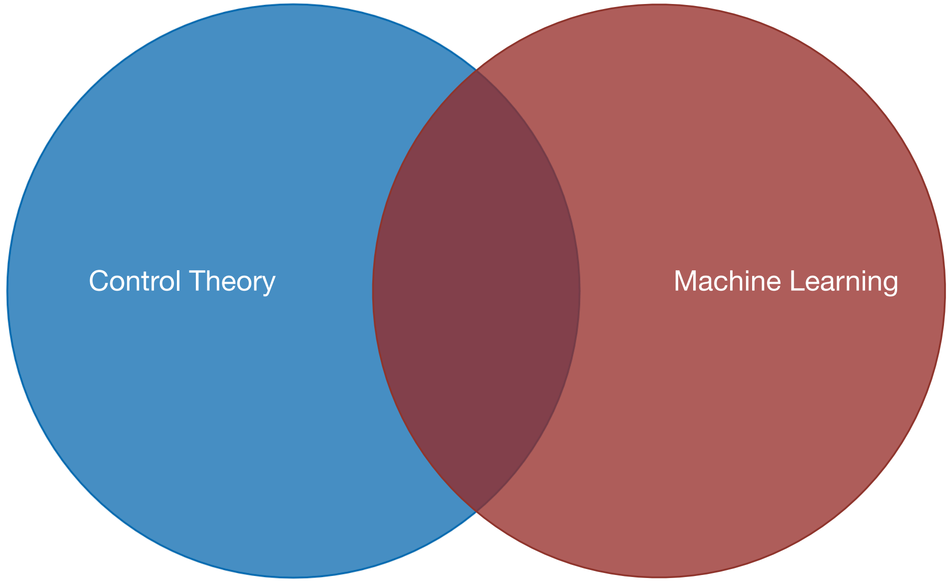


Smart healthcare


Not there yet…

safety, robustness, reliability?

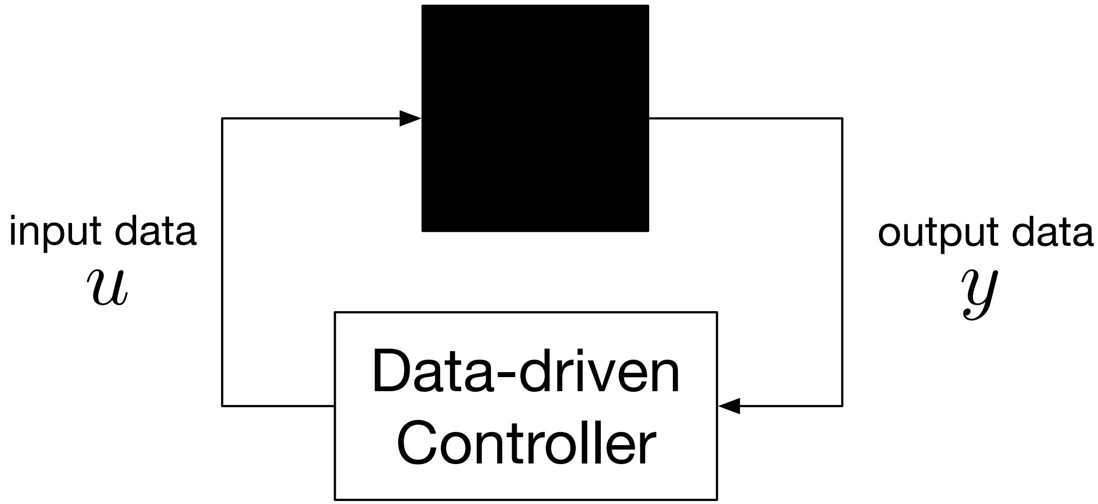
# Control meets Learning





"One of the major developments in control over the past decade – and one of the most important moving forward – is the interaction of machine learning and control systems" – [Control for societal-scale challenges]

# Control without a model?



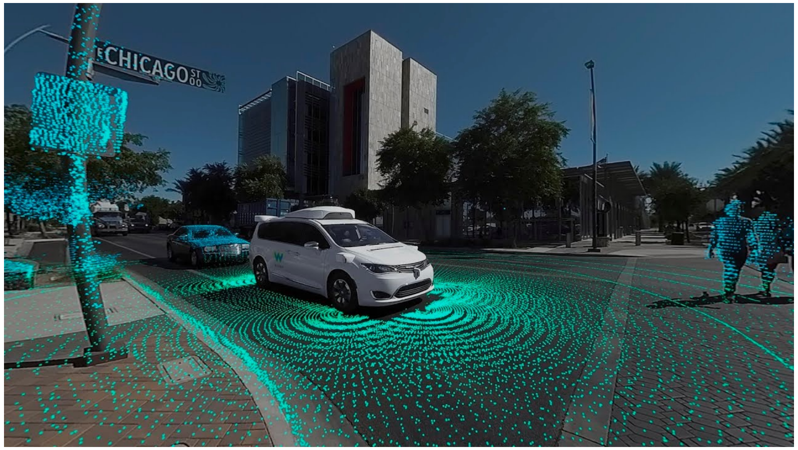

Learn a model?

**data $\longrightarrow$ model $\longrightarrow$ controller**

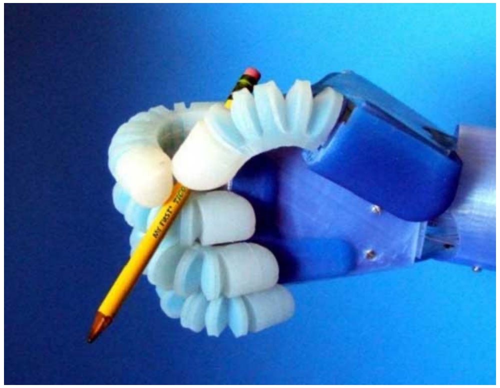
…or learn a controller directly?

# Why direct data-driven control?

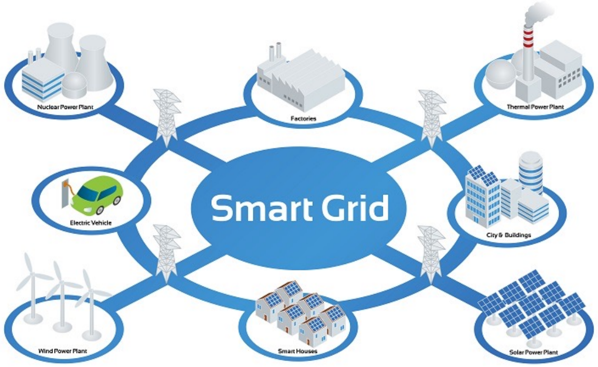
Model not conceivable




Too complex



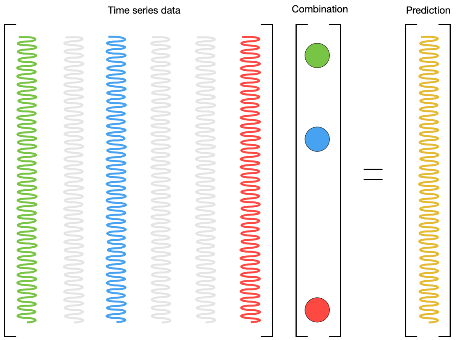
Models not available



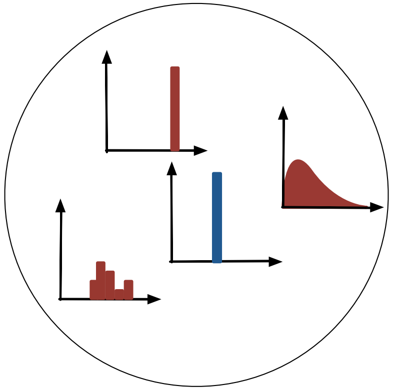

Too costly

# Outline

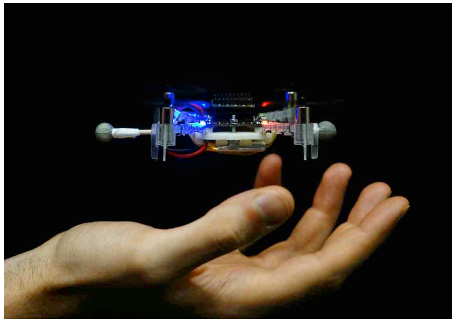
**Data-enabled Predictive Control (DeePC)**




**Robustification via Regularization**



**Applications in Robotics**



# Behavioral System Theory

The behavior is all there is!

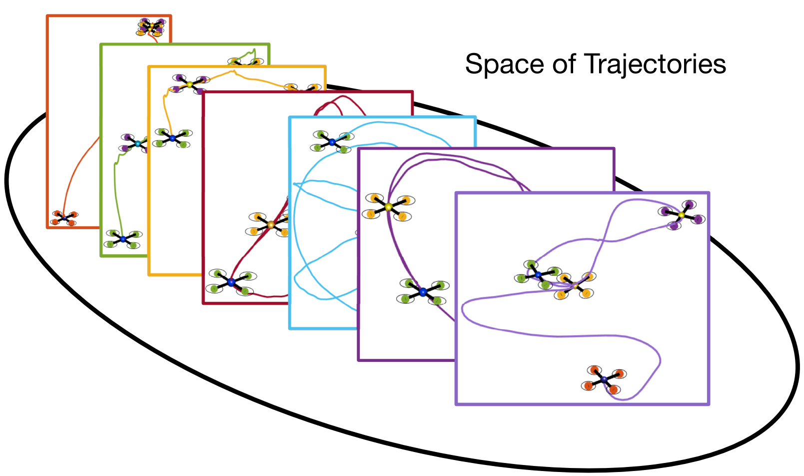
Possible to construct $\mathscr{B}$ using data?

**Jan Willems**

Introduced behavioral system theory $\sim$1980s

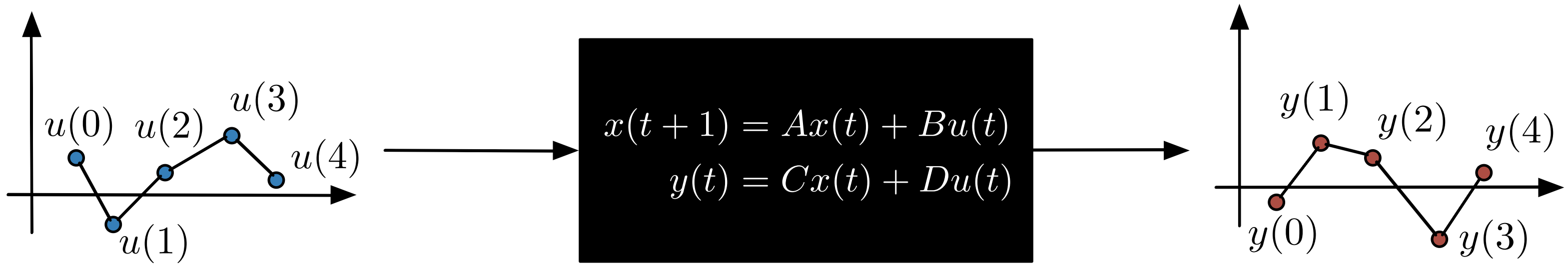
## Dynamical System

A discrete-time dynamical system is a 3-tuple $(\mathbb{Z}_{\geq 0}, \mathbb{R}^{m+p}, \mathscr{B})$ where $\mathscr{B} \subseteq (\mathbb{R}^{m+p})^{\mathbb{Z}_{\geq 0}}$ is the behavior (set of allowable trajectories).



# Fundamental Lemma

[Willems et al. '05], [Markovsky, Dörfler '20]




Given data $(u_{[0,T-1]}, y_{[0,T-1]})$ and parameters $\begin{cases} \text{state dimension } n \\ \text{trajectory length } L \end{cases}$

$$\operatorname{im} \begin{bmatrix} u(0) & u(1) & \cdots & u(T-L) \\ \vdots & \vdots & \ddots & \vdots \\ u(L-1) & u(L) & \cdots & u(T-1) \\ y(0) & y(1) & \cdots & y(T-L) \\ \vdots & \vdots & \ddots & \vdots \\ y(L-1) & y(L) & \cdots & y(T-1) \end{bmatrix} = \text{set of **all** } L\text{-length trajectories}$$

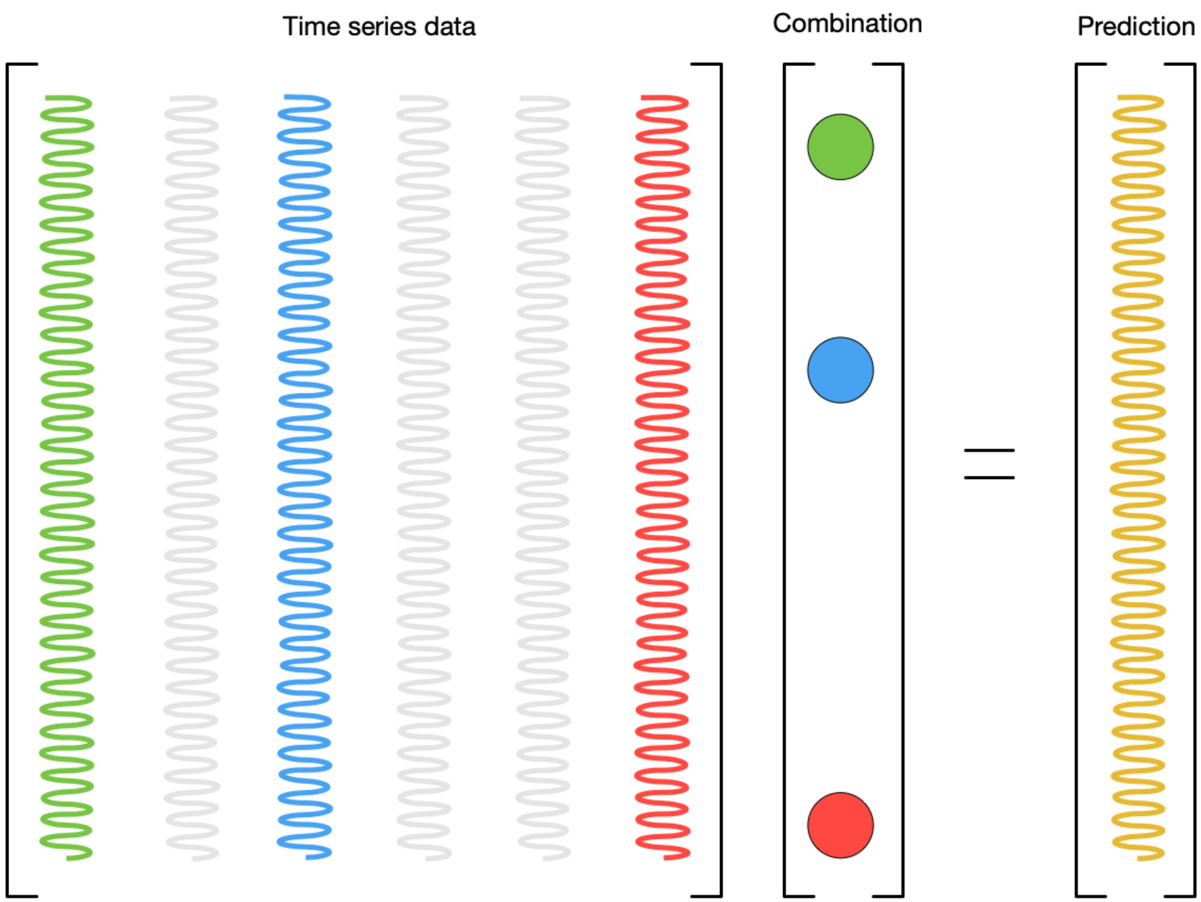
set of **all** $L$-length trajectories

if and only if trajectory matrix has rank $mL + n$ for $L$ large enough.

Any trajectory can be generated from finitely many, sufficiently rich data trajectories

# Trajectory Matrix: a predictive model

Idea: The trajectory matrix using <u>raw data</u> can serve as a non-parametric <u>predictive model</u>!

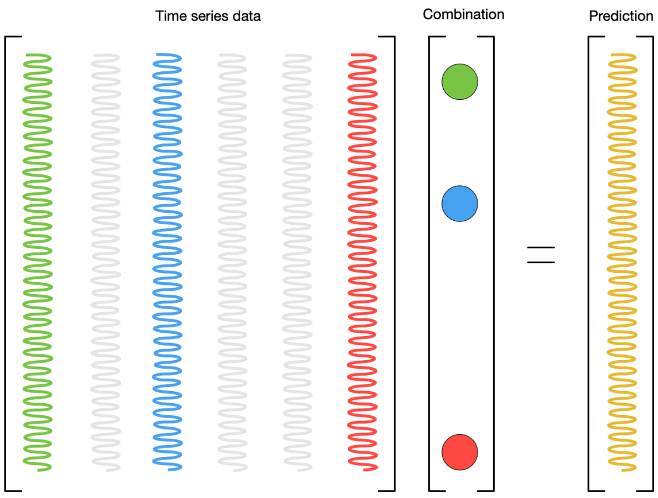


# Trajectory Prediction [Markovsky, Rapisarda '08]

Given:

- Data $(u^{\mathsf{d}}_{[0,T-1]}, y^{\mathsf{d}}_{[0,T-1]})$
- Initial trajectory $(u_{\mathsf{ini}}, y_{\mathsf{ini}}) \in \mathbb{R}^{(m+p)T_{\mathsf{ini}}}$
- Future input $u \in \mathbb{R}^{mT_{\mathsf{f}}}$

Predict the future output $y \in \mathbb{R}^{pT_{\mathsf{f}}}$.




$$\underbrace{\left[\begin{array}{cccc}
u^{\mathsf{d}}(0) & u^{\mathsf{d}}(1) & \cdots & u^{\mathsf{d}}(T-T_{\mathsf{ini}}-T_{\mathsf{f}}) \\
\vdots & \vdots & \ddots & \vdots \\
u^{\mathsf{d}}(T_{\mathsf{ini}}-1) & u^{\mathsf{d}}(T_{\mathsf{ini}}) & \cdots & u^{\mathsf{d}}(T-T_{\mathsf{f}}-1) \\
\hline
y^{\mathsf{d}}(0) & y^{\mathsf{d}}(1) & \cdots & y^{\mathsf{d}}(T-T_{\mathsf{ini}}-T_{\mathsf{f}}) \\
\vdots & \vdots & \ddots & \vdots \\
y^{\mathsf{d}}(T_{\mathsf{ini}}-1) & y^{\mathsf{d}}(T_{\mathsf{ini}}) & \cdots & y^{\mathsf{d}}(T-T_{\mathsf{f}}-1) \\
\hline
u^{\mathsf{d}}(T_{\mathsf{ini}}) & u^{\mathsf{d}}(T_{\mathsf{ini}}+1) & \cdots & u^{\mathsf{d}}(T-T_{\mathsf{f}}) \\
\vdots & \vdots & \ddots & \vdots \\
u^{\mathsf{d}}(T_{\mathsf{ini}}+T_{\mathsf{f}}-1) & u^{\mathsf{d}}(T_{\mathsf{ini}}+T_{\mathsf{f}}) & \cdots & u^{\mathsf{d}}(T-1) \\
\hline
y^{\mathsf{d}}(T_{\mathsf{ini}}) & y^{\mathsf{d}}(T_{\mathsf{ini}}+1) & \cdots & y^{\mathsf{d}}(T-T_{\mathsf{f}}) \\
\vdots & \vdots & \ddots & \vdots \\
y^{\mathsf{d}}(T_{\mathsf{ini}}+T_{\mathsf{f}}-1) & y^{\mathsf{d}}(T_{\mathsf{ini}}+T_{\mathsf{f}}) & \cdots & y^{\mathsf{d}}(T-1)
\end{array}\right]}_{=: \begin{bmatrix} U_{\mathsf{p}} \\ Y_{\mathsf{p}} \\ U_{\mathsf{f}} \\ Y_{\mathsf{f}} \end{bmatrix}}
\quad g = \begin{bmatrix} u_{\mathsf{ini}} \\ y_{\mathsf{ini}} \\ u \\ y \end{bmatrix}$$

$\left.\begin{matrix} u_{\mathsf{ini}} \\ y_{\mathsf{ini}} \end{matrix}\right\}$ set initial condition

$\left.\begin{matrix} u \\ y \end{matrix}\right\}$ prediction

Prediction unique if $T_{\mathsf{ini}}$ large enough

# Data-enabled Predictive Control

## Goal

Design a controller to:

- track reference output trajectory
- satisfy safety constraints $u(t) \in \mathcal{U},\ y(t) \in \mathcal{Y}$

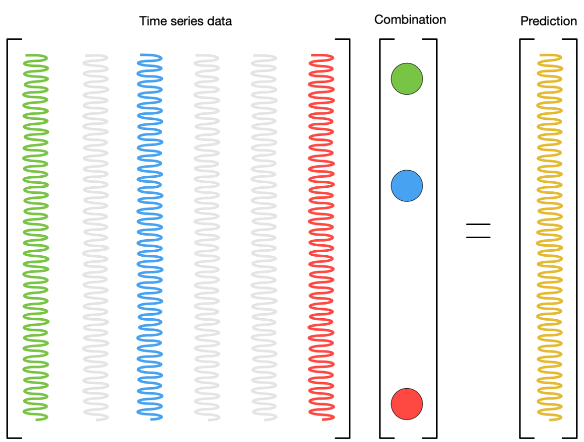
When the state space model (i.e., $A, B, C, D$ ) is unknown:

**DeePC:**

$$\min_{g,u,y} \quad \left.\sum_{k=0}^{T_{\mathsf{f}}-1} \|y_k - r_{t+k}\|_Q^2 + \|u_k\|_R^2 \right\} \text{performance}$$

$$\text{s.t.} \quad \begin{bmatrix} U_{\mathsf{p}} \\ Y_{\mathsf{p}} \\ U_{\mathsf{f}} \\ Y_{\mathsf{f}} \end{bmatrix} g = \begin{bmatrix} u_{\mathsf{ini}} \\ y_{\mathsf{ini}} \\ u \\ y \end{bmatrix} \quad \begin{array}{l} \left.\begin{array}{l} \\ \\ \end{array}\right\} \text{initial condition} \\ \left.\begin{array}{l} \\ \\ \end{array}\right\} \text{prediction} \end{array}$$

$$\left.\begin{array}{l} u_k \in \mathcal{U},\ \forall k \in \{0, \dots, T_{\mathsf{f}} - 1\} \\ y_k \in \mathcal{Y},\ \forall k \in \{0, \dots, T_{\mathsf{f}} - 1\} \end{array}\right\} \text{safety}$$



# Models vs data

**MPC:**

$$\min_{u,x,y} \sum_{k=0}^{T_{\mathsf{f}}-1} \|y_k - r_{t+k}\|_Q^2 + \|u_k\|_R^2$$

s.t.

$$x_{k+1} = Ax_k + Bu_k,\ \forall k \in \{0,\dots,T_{\mathsf{f}}-1\}$$
$$y_k = Cx_k + Du_k,\ \forall k \in \{0,\dots,T_{\mathsf{f}}-1\}$$
$$x_{k+1} = Ax_k + Bu_k,\ \forall k \in \{-T_{\mathsf{ini}},\dots,-1\}$$
$$y_k = Cx_k + Du_k,\ \forall k \in \{-T_{\mathsf{ini}},\dots,-1\}$$
$$u_k \in \mathcal{U},\ \forall k \in \{0,\dots,T_{\mathsf{f}}-1\}$$
$$y_k \in \mathcal{Y},\ \forall k \in \{0,\dots,T_{\mathsf{f}}-1\}$$

**DeePC:**

$$\min_{g,u,y} \sum_{k=0}^{T_{\mathsf{f}}-1} \|y_k - r_{t+k}\|_Q^2 + \|u_k\|_R^2$$

s.t.

$$\begin{bmatrix} U_{\mathsf{p}} \\ Y_{\mathsf{p}} \\ U_{\mathsf{f}} \\ Y_{\mathsf{f}} \end{bmatrix} g = \begin{bmatrix} u_{\mathsf{ini}} \\ y_{\mathsf{ini}} \\ u \\ y \end{bmatrix}$$
$$u_k \in \mathcal{U},\ \forall k \in \{0,\dots,T_{\mathsf{f}}-1\}$$
$$y_k \in \mathcal{Y},\ \forall k \in \{0,\dots,T_{\mathsf{f}}-1\}$$

**Theorem [Coulson et al. '19]**

MPC and DeePC have equivalent closed-loop behavior.

What about noisy data?
…Nonlinear systems?

We need a robustified approach!

# Robustify against what?

## Infeasibility

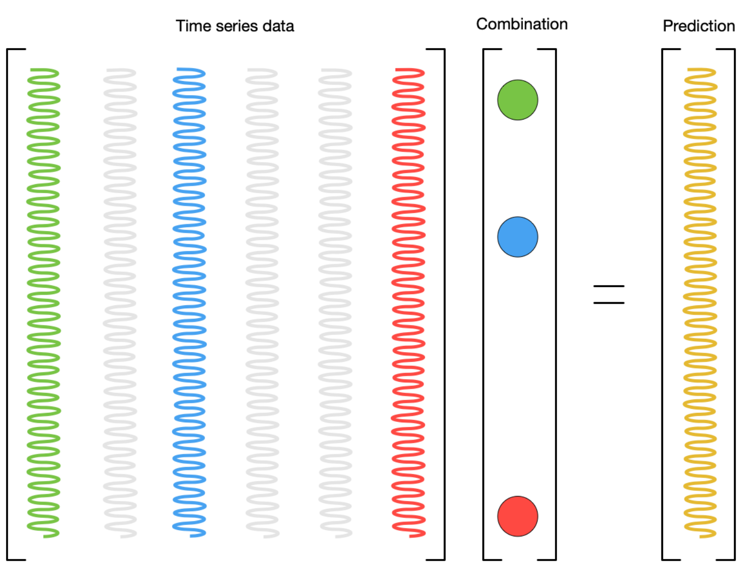
Online data $(u_{\text{ini}}, y_{\text{ini}})$ may be inconsistent with data in the trajectory matrix

$$\text{No } g \text{ such that } \begin{bmatrix} U_{\text{p}} \\ Y_{\text{p}} \\ U_{\text{f}} \\ Y_{\text{f}} \end{bmatrix} g = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \\ y \end{bmatrix} \begin{matrix} \left.\vphantom{\begin{matrix} u \\ y \end{matrix}}\right\} \text{initial condition} \\ \vphantom{\begin{matrix} u \\ y \end{matrix}} \end{matrix}$$

## Over-optimism

Trajectory matrix contains noisy data $\Longrightarrow$ Trajectory selection over-optimistic

No longer low rank $\rightarrow$




$\leftarrow$ Can predict trajectories outside of subspace of allowable trajectories robustification via regularization

# Distributionally Robust Optimization

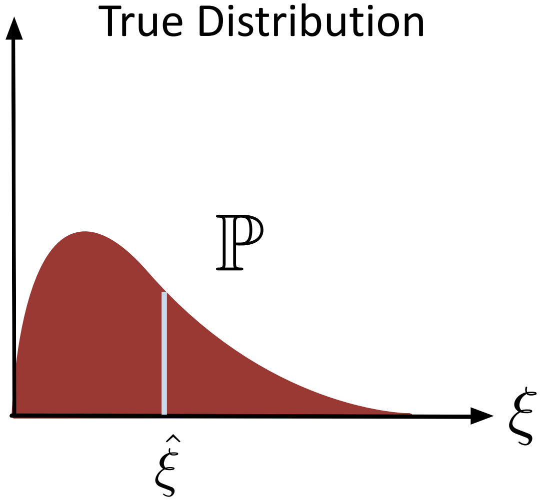
[Bertsimas et al. ‘04], [Ben-Tal et al. ‘09], [Kuhn et al. ‘19]

Objective: $\underset{g \in G}{\text{minimize}} \quad \mathbb{E}_{\mathbb{P}}[c(\xi, g)]$

$\xi$ is a random variable distributed according to unknown distribution $\mathbb{P}$

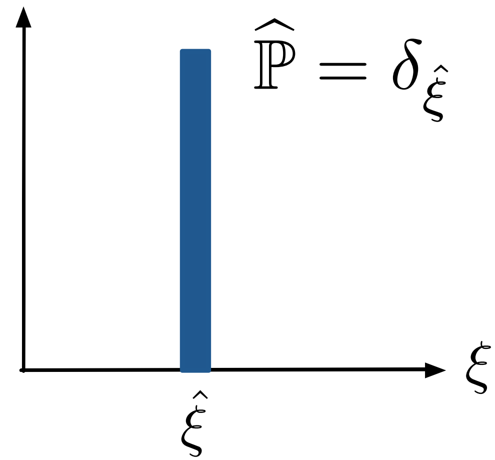
We only have access to measured data sample $\hat{\xi}$

True Distribution




$\neq$

Empirical Distribution




Solve sample average approximation problem

$$g^\star \in \underset{g \in G}{\text{argmin}} \quad \mathbb{E}_{\widehat{\mathbb{P}}}[c(\xi, g)]$$

$\longrightarrow$

Poor true expected cost

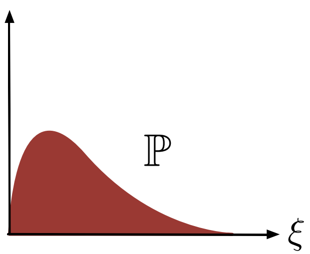
$$\mathbb{E}_{\mathbb{P}}[c(\xi, g^\star)]$$

# Wasserstein Ball

[Kuhn et al. ‘19]

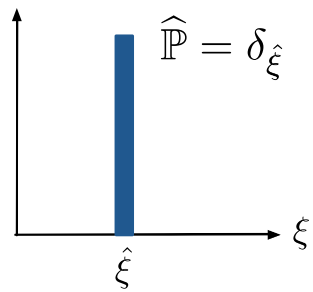
Robustify against the fact that:

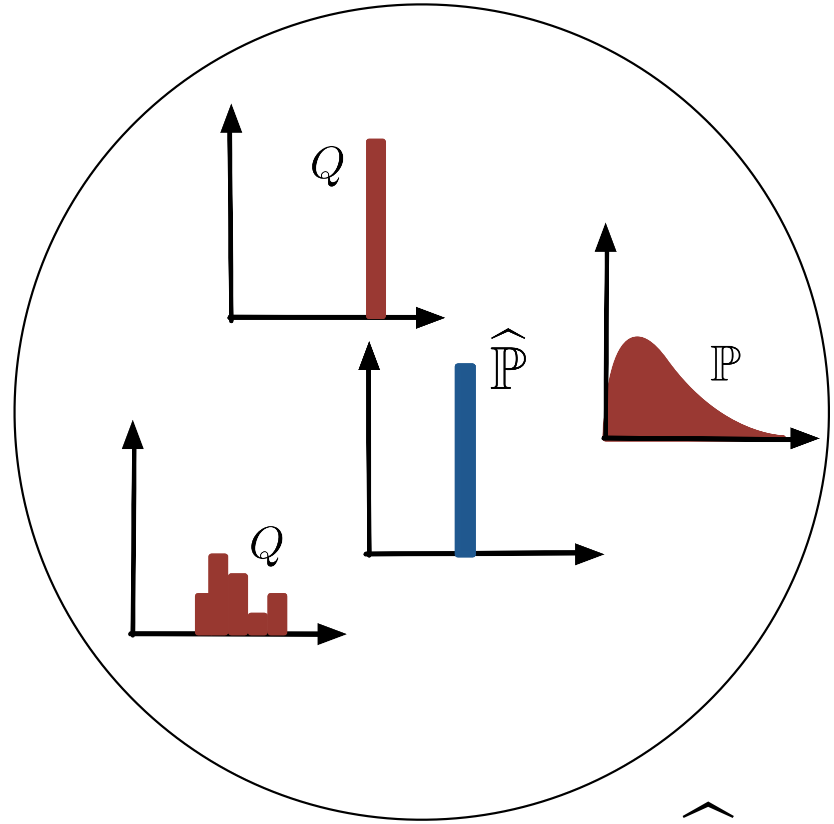
True Distribution




$\neq$

Empirical Distribution




$B_\epsilon(\widehat{\mathbb{P}})$

Contains distributions that could have generated the data

**Distributionally Robust Optimization**

$$\inf_{g \in G} \sup_{Q \in B_\epsilon(\widehat{\mathbb{P}})} \mathbb{E}_Q[c(\xi, g)]$$

where $B_\epsilon(\widehat{\mathbb{P}}) = \left\{ Q \,\middle|\, \int_\Xi \|\xi - \hat{\xi}\| Q(d\xi) \leq \epsilon \right\}$

is the Wasserstein ball.

# Abstracted DeePC

## DeePC + slack

$$\min_{g,u,y} \sum_{k=0}^{T_\mathsf{f}-1} f(u_k, y_k) + h(\sigma_y)$$

$$\text{s.t.} \quad \begin{bmatrix} U_\mathsf{p} \\ Y_\mathsf{p} \\ U_\mathsf{f} \\ Y_\mathsf{f} \end{bmatrix} g = \begin{bmatrix} u_\mathsf{ini} \\ y_\mathsf{ini} + \sigma_y \\ u \\ y \end{bmatrix}$$

$$u_k \in \mathcal{U}, \ \forall k \in \{0, \ldots, T_\mathsf{f} - 1\}$$

## Abstracted DeePC

$$\underset{g \in G}{\text{minimize}} \quad c(\hat{\xi}, g)$$

with $\hat{\xi} = \begin{bmatrix} {U_\mathsf{p}}^\top & {Y_\mathsf{p}}^\top & {U_\mathsf{f}}^\top & {Y_\mathsf{f}}^\top \end{bmatrix}^\top$

$$G = \left\{ g \,\middle|\, \begin{bmatrix} U_\mathsf{p} \\ U_\mathsf{f} \end{bmatrix} g = \begin{bmatrix} u_\mathsf{ini} \\ u \end{bmatrix}, u \in \mathcal{U}^{T_\mathsf{f}} \right\}$$

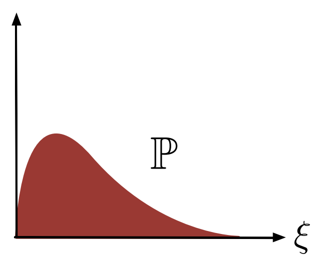
Data $\hat{\xi}$ is a particular measurement (realization) of random variable $\xi$ distributed according to unknown distribution $\mathbb{P}$.

What we care about:

$$\min_{g \in G} \mathbb{E}_\mathbb{P}[c(\xi, g)]$$

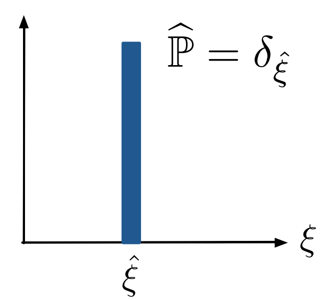
What DeePC is solving:

$$\min_{g \in G} c(\hat{\xi}, g) = \min_{g \in G} \mathbb{E}_{\widehat{\mathbb{P}}}[c(\xi, g)]$$




True Distribution

$\neq$




Empirical Distribution

# Distributionally Robust DeePC

**Theorem [Coulson et al. '21]**

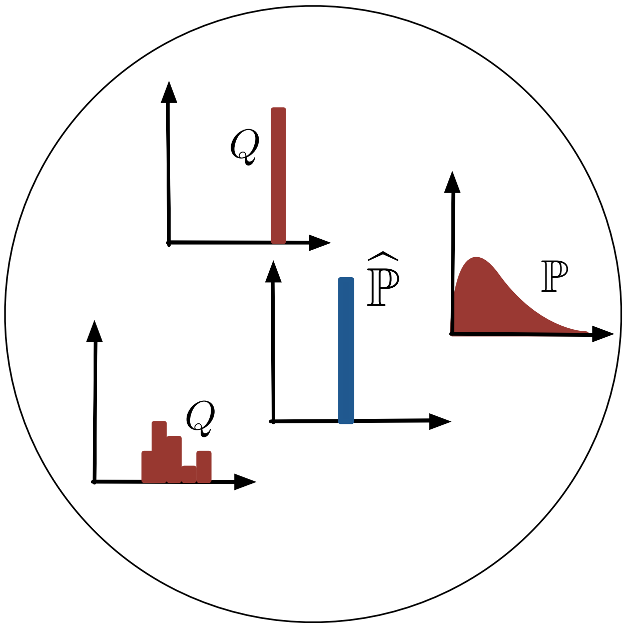
Under minor technical conditions

$$\inf_{g \in G} \sup_{Q \in B_\epsilon(\widehat{\mathbb{P}})} \mathbb{E}_Q[c(\xi, g)] = \inf_{g \in G} \underbrace{c(\hat{\xi}, g)}_{\text{nominal DeePC}} + \underbrace{\epsilon \mathsf{Lip}(c) \|g\|_*}_{\text{regularization}}$$

$p$-norm robustness $\iff$ $q$-norm regularization

$$\frac{1}{p} + \frac{1}{q} = 1$$

The Wasserstein ball contains more than just LTI systems with additive noise




Proof uses methods from [Mohajerin Esfahani, Kuhn '18]

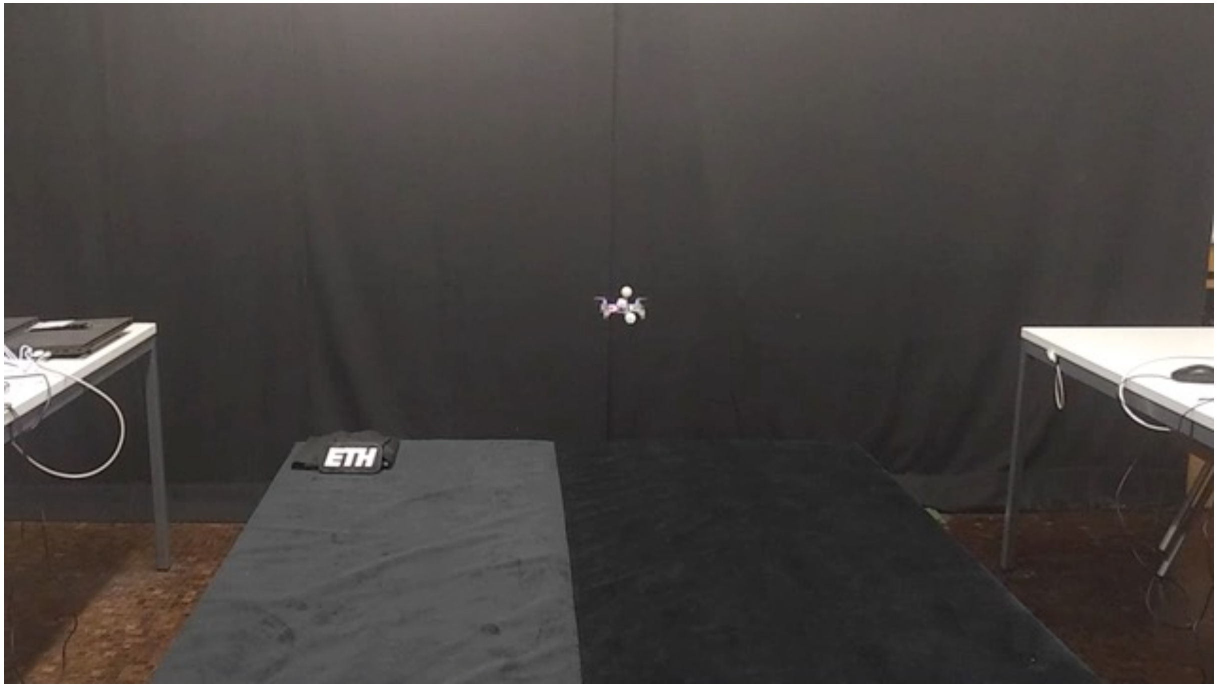
# Regularized and Robust DeePC

$$\min_{g,u,y} \quad \sum_{k=0}^{T_\mathsf{f}-1} \|y_k - r_{t+k}\|_Q^2 + \|u_k\|_R^2 + \lambda_y \|\sigma_y\|_2^2 + \lambda_g \|g\|_1$$

$$\text{s.t.} \quad \begin{bmatrix} U_\mathsf{p} \\ Y_\mathsf{p} \\ U_\mathsf{f} \\ Y_\mathsf{f} \end{bmatrix} g = \begin{bmatrix} u_\mathsf{ini} \\ y_\mathsf{ini} + \sigma_y \\ u \\ y \end{bmatrix}$$

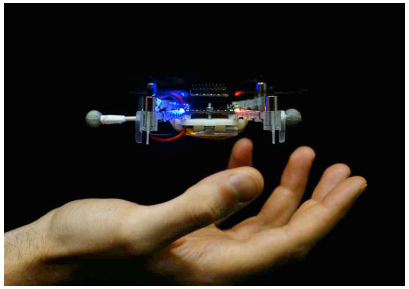
$$u_k \in \mathcal{U}, \ \forall k \in \{0, \ldots, T_\mathsf{f} - 1\}$$

$$y_k \in \mathcal{Y}, \ \forall k \in \{0, \ldots, T_\mathsf{f} - 1\}$$

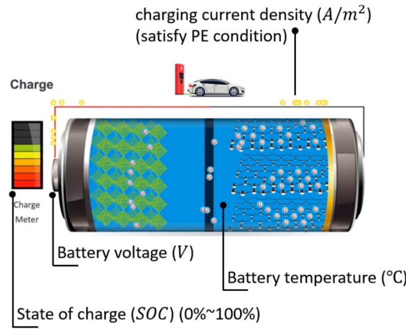



[Elokda, Coulson et al. '21]

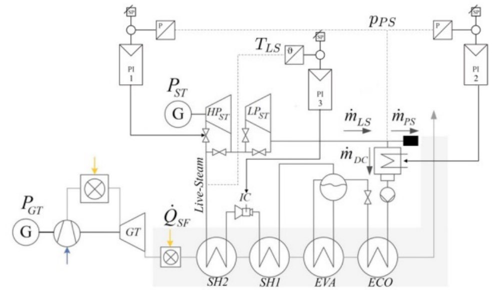
# Versatile method



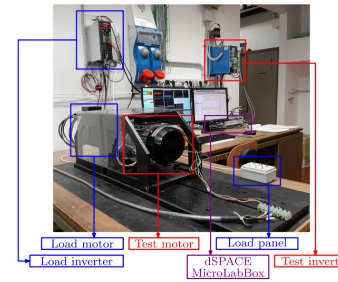
Quadcopter




Batteries [Chen at al.]



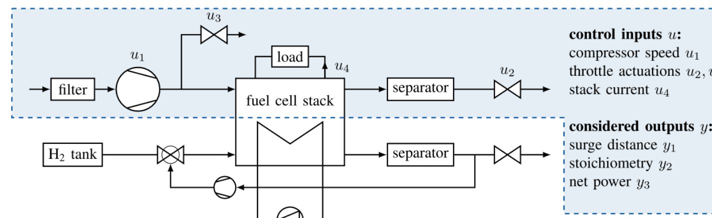

Power plant [Mahdavipour et al.]




Synchronous motor drive [Carlet et al.]

Autonomous excavator

Buildings & energy hubs [Lian et al.]

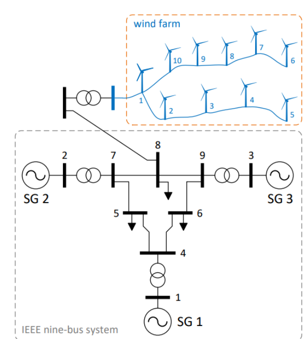



Fuel cell [Schmitt et al.]

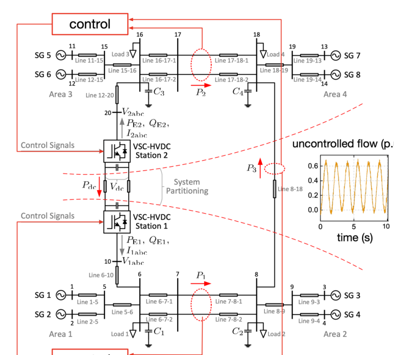
Quadruped [Fawcett et al.]

Traffic [Wang et al.]

Greenhouse [Kerkhof et al.]




Wind farm [Markovsky et al.]




Power systems

# Further results

**Distributionally Robust Chance Constrained Data-Enabled Predictive Control**

Jeremy Coulson, John Lygeros, *Fellow, IEEE*, and Florian Dörfler

- Leverage more data
- Safety through output chance constraints
- Fundamental lemma for better matrix structures

**Bridging Direct and Indirect Data-Driven Control Formulations via Regularizations and Relaxations**

Florian Dörfler, *Senior Member, IEEE*, Jeremy Coulson, *Member, IEEE*, and Ivan Markovsky, *Associate Member, IEEE*

- Regularizations via indirect data-driven control

**A Quantitative Notion of Persistency of Excitation and the Robust Fundamental Lemma**

Jeremy Coulson, Henk J. van Waarde, *Member, IEEE*, John Lygeros, *Fellow, IEEE*, and Florian Dörfler, *Senior Member, IEEE*

- Quantitative fundamental lemma

# Summary

- Matrix of time-series data is a predictive model
- Robustification = regularization
- Successful deployment on many experiments and simulations

Open & ongoing:

- Real-time adaptation using streaming data
- Incorporating prior (physical) knowledge
- Large scale integration (distributed control)