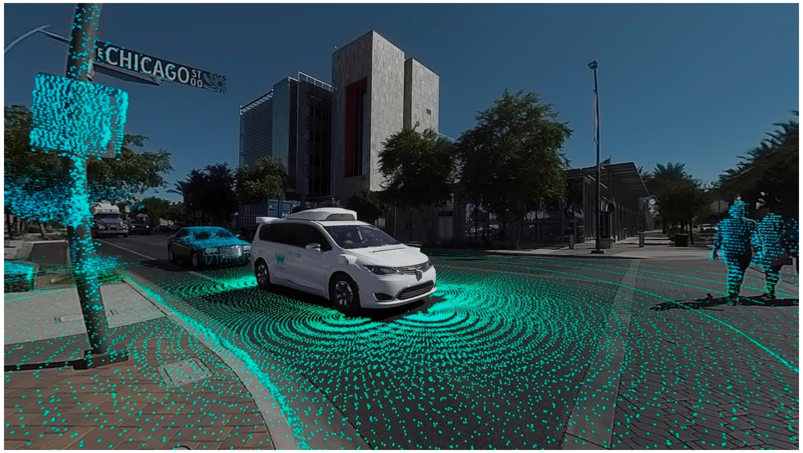
Data-enabled Predictive Control

Regularization and Robustness

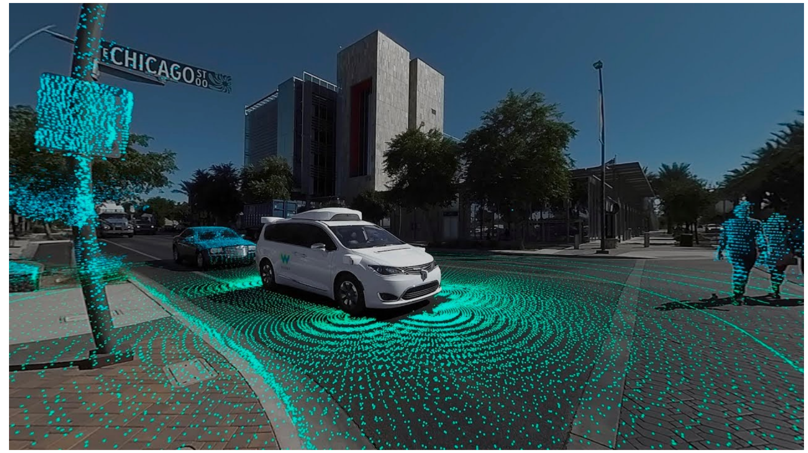
Jeremy Coulson (joint work with Florian Dörfler & John Lygeros)

IFAC 2023

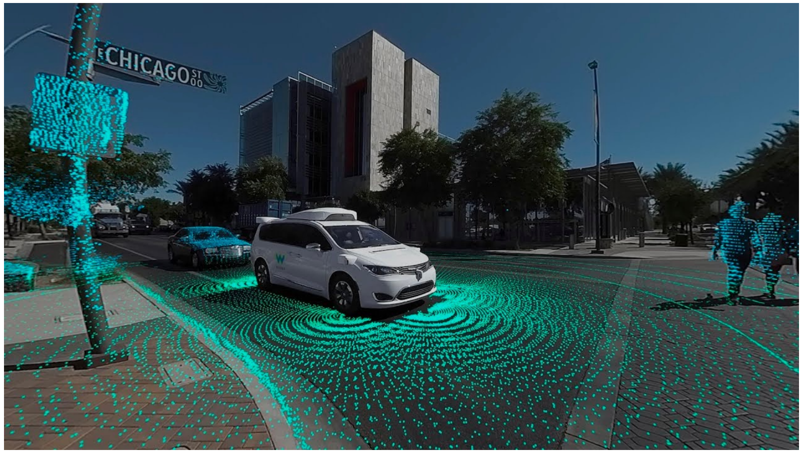
Living in a data-rich world



Living in a data-rich world

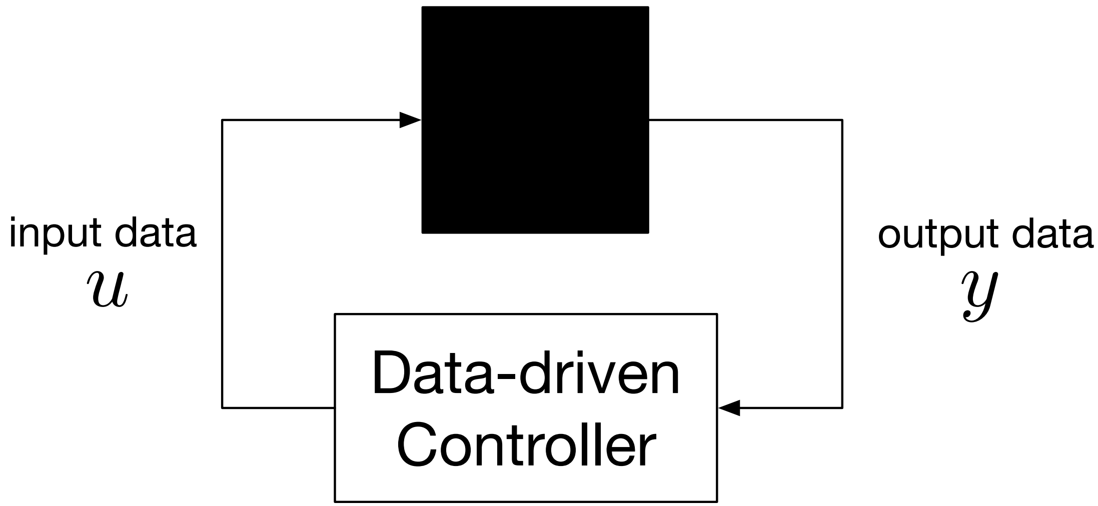


Living in a data-rich world

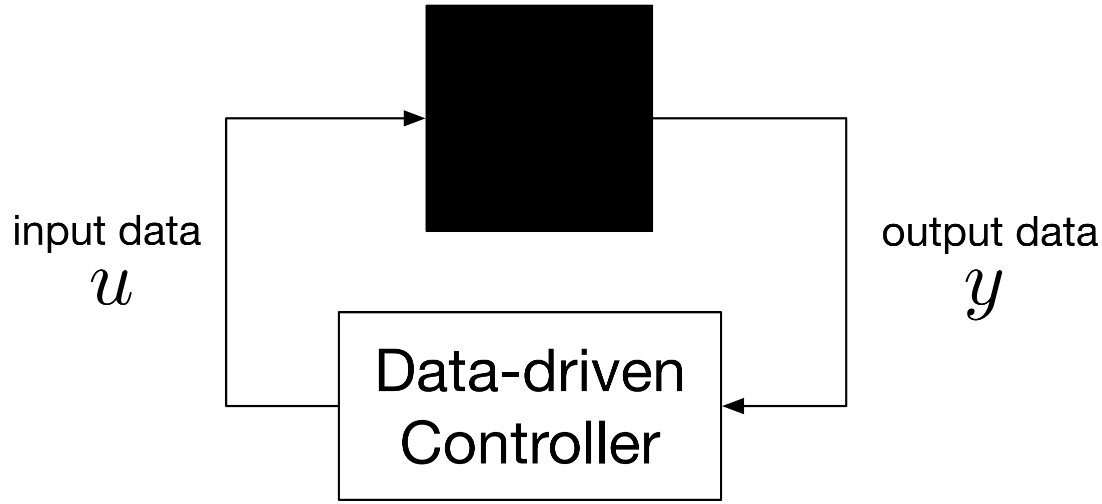


safety, robustness, reliability

Data-driven Control

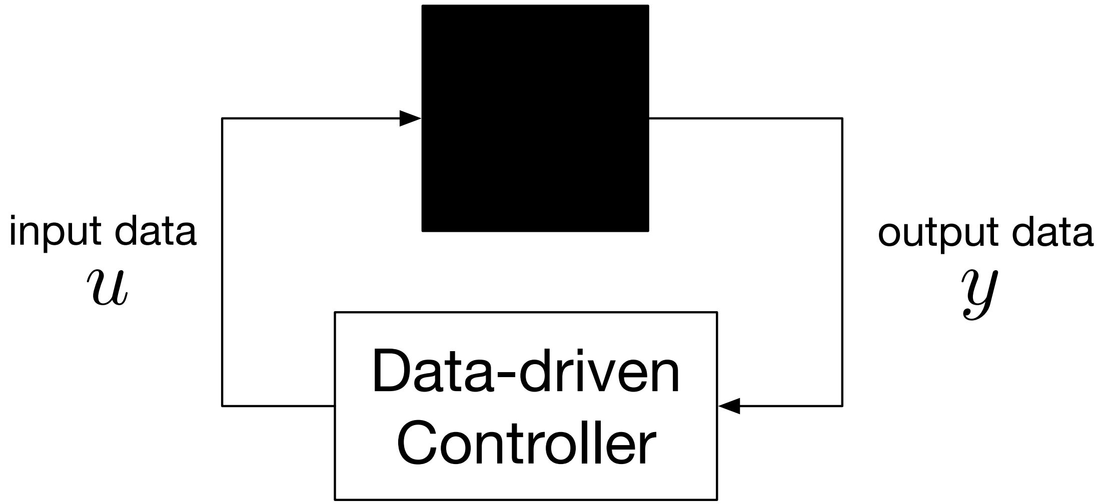


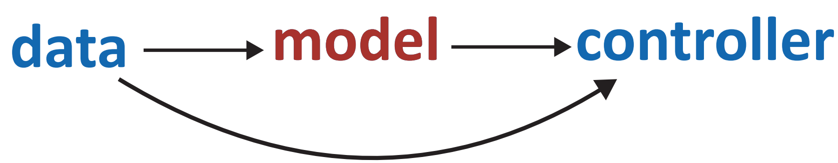
Data-driven Control



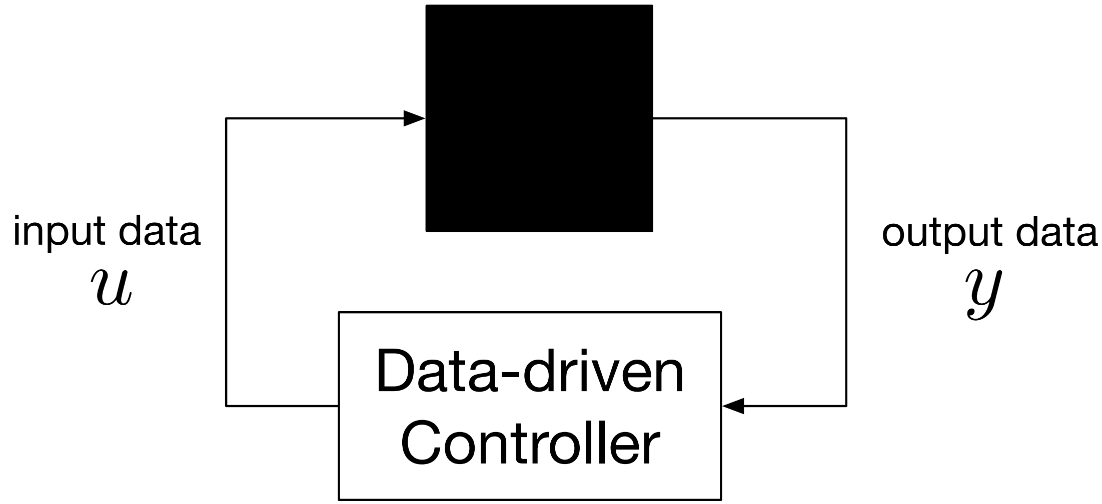
data $\longrightarrow$ **model** $\longrightarrow$ **controller**

Data-driven Control



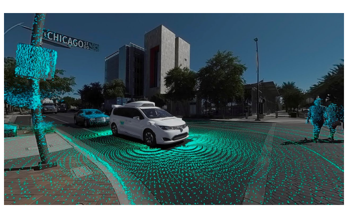


Data-driven Control



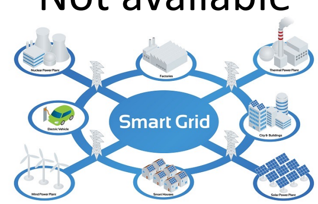
data → **model** → **controller**

Models not conceivable



Too complex

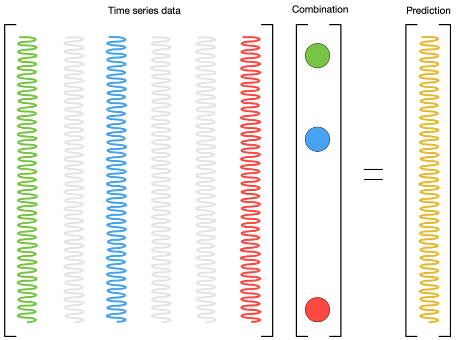
Not available



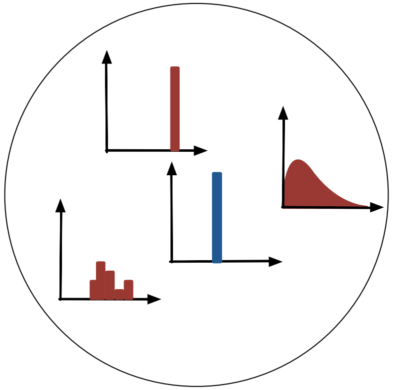
Too costly

Outline

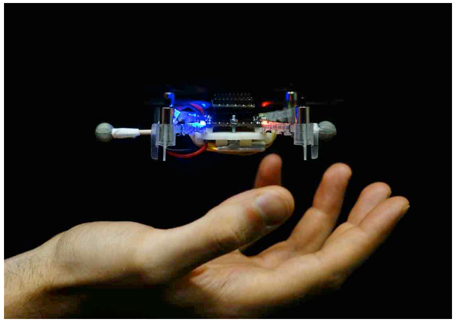
Data-enabled Predictive Control (DeePC)



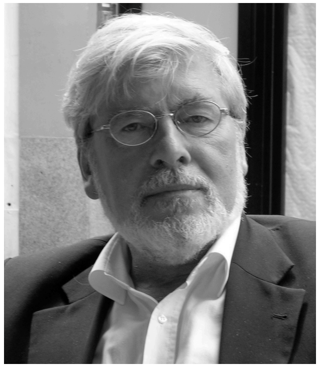
Robustification via Regularization



Applications in Robotics



Behavioral System Theory

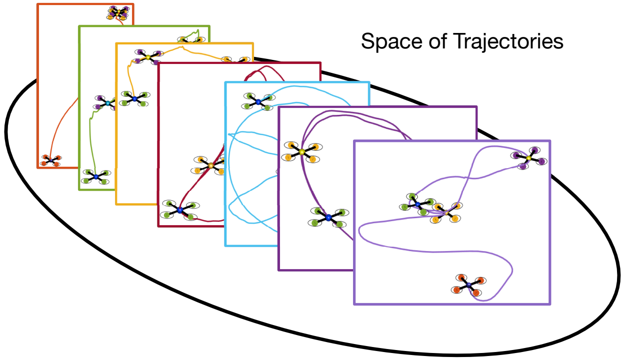


Jan Willems

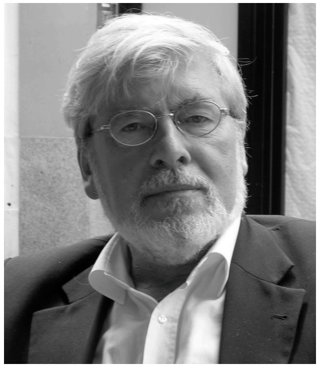
Introduced behavioral system theory $\sim$ 1980s

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).



Behavioral System Theory

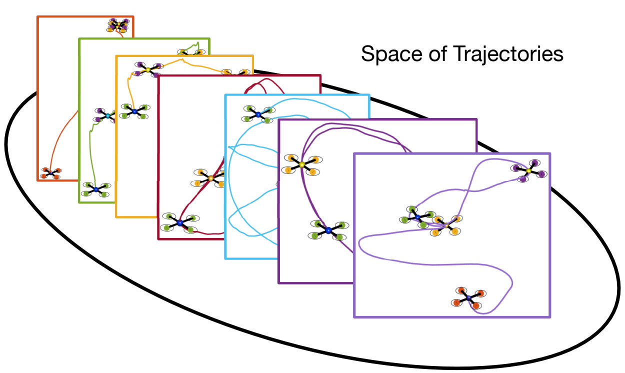


Jan Willems

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).



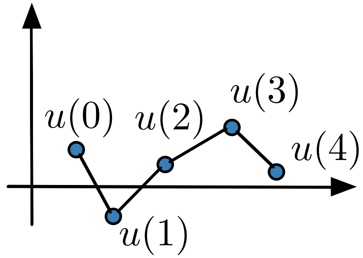
LTI system

=

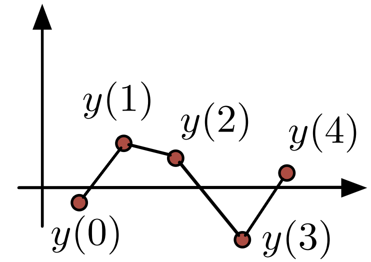
time-shift invariant subspace

Fundamental Lemma

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



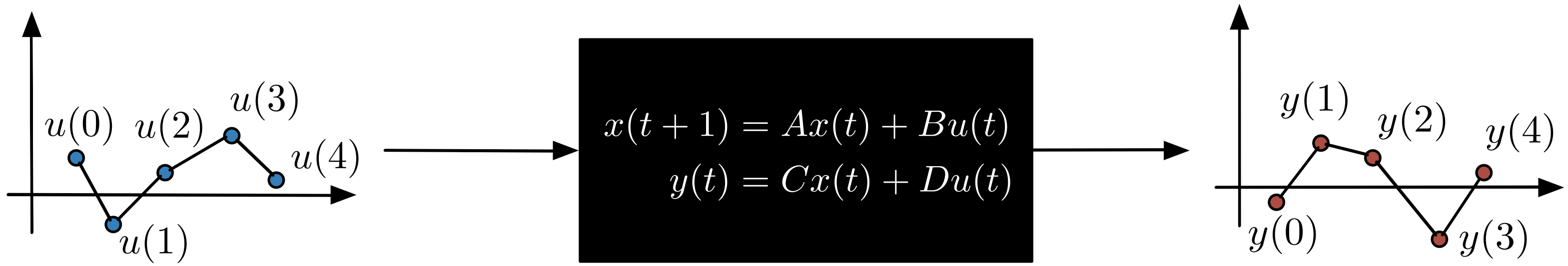
$$
\begin{aligned}
x(t+1) &= Ax(t) + Bu(t)\\
y(t) &= Cx(t) + Du(t)
\end{aligned}
$$



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

Fundamental Lemma

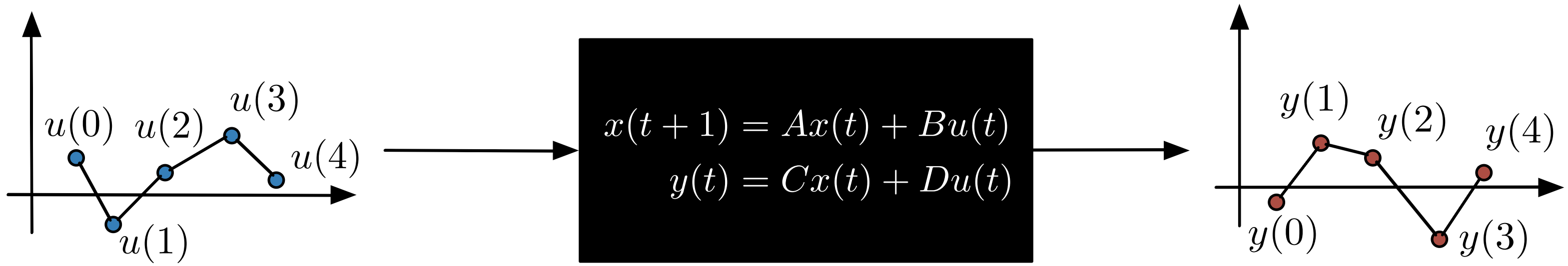
[Willems et al. '05], [Markovsky, Dörfler '20]



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

$$\text{im} \begin{bmatrix} u(0) & u(1) & \cdots & u(T-L) \\ \vdots & \vdots & \ddots & \vdots \\ u(L-1) & u(L) & \cdots & u(T-1) \\ y(0) & y(1) & \cdots & y(T-L) \\ \vdots & \vdots & \ddots & \vdots \\ y(L-1) & y(L) & \cdots & y(T-1) \end{bmatrix}$$

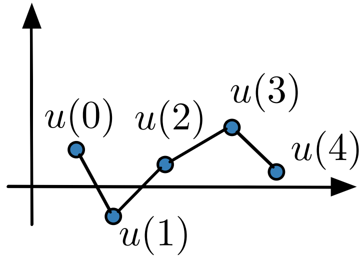
Fundamental Lemma [Willems et al. '05], [Markovsky, Dörfler '20]



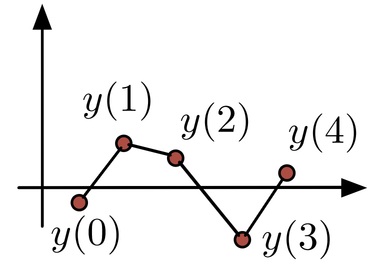
Given data $(u_{[0,T-1]}, y_{[0,T-1]})$ and parameters $\begin{cases} \text{state dimension } n \\ \text{lag } \ell \\ \text{trajectory length } L \end{cases}$

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

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



$$
\begin{aligned}
x(t+1) &= Ax(t) + Bu(t) \\
y(t) &= Cx(t) + Du(t)
\end{aligned}
$$



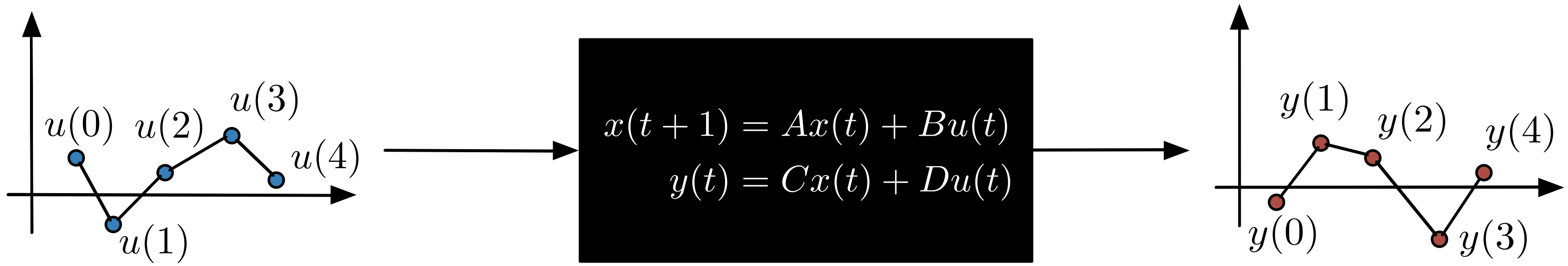
Given data $(u_{[0,T-1]}, y_{[0,T-1]})$ and parameters $\begin{cases} \text{state dimension } n \\ \text{lag } \ell \\ \text{trajectory length } L \end{cases}$

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

if and only if trajectory matrix has rank $mL + n$ for $L \geq \ell$.

Fundamental Lemma

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



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

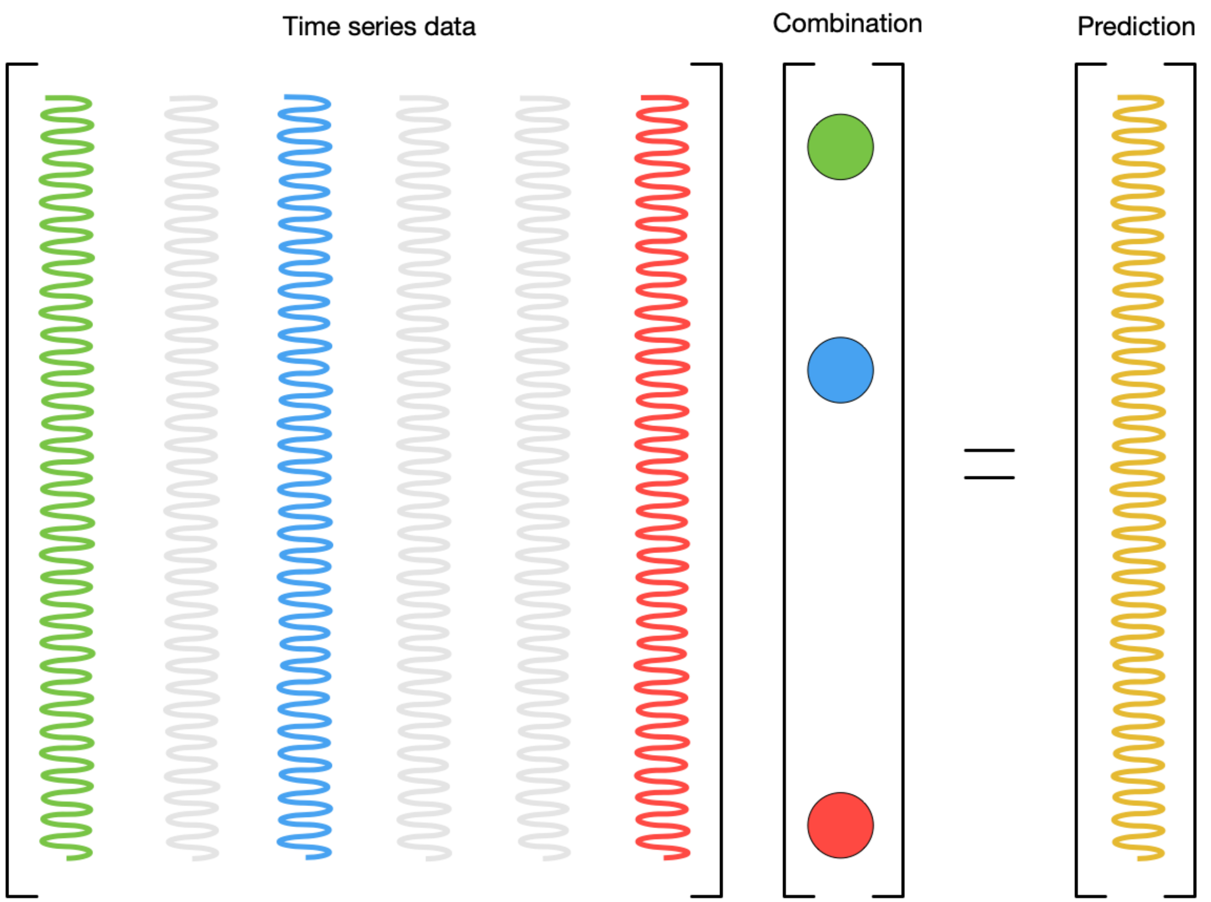
$$\operatorname{im} \begin{bmatrix} u(0) & u(1) & \cdots & u(T-L) \\ \vdots & \vdots & \ddots & \vdots \\ u(L-1) & u(L) & \cdots & u(T-1) \\ y(0) & y(1) & \cdots & y(T-L) \\ \vdots & \vdots & \ddots & \vdots \\ y(L-1) & y(L) & \cdots & y(T-1) \end{bmatrix} = \text{set of } \textbf{all } L\text{-length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

if and only if trajectory matrix has rank $mL + n$ for $L \geq \ell$.

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

Trajectory Matrix: a predictive model

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

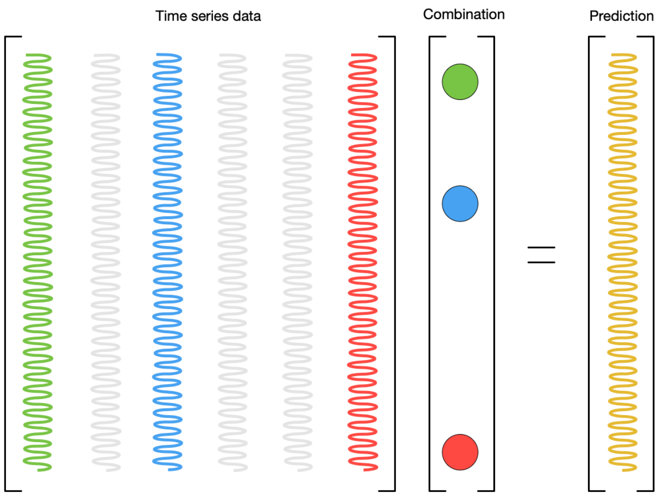


Trajectory Prediction [Markovsky, Rapisarda '08]

Given:

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

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

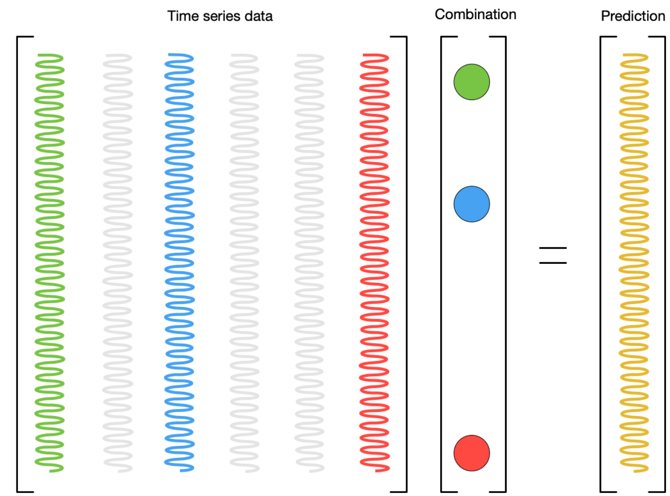


Trajectory Prediction [Markovsky, Rapisarda ‘08]

Given:

- Data $\left(u^{\mathsf{d}}_{[0,T-1]}, y^{\mathsf{d}}_{[0,T-1]}\right)$
- 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 $y \in \mathbb{R}^{pT_{\mathsf{f}}}$.



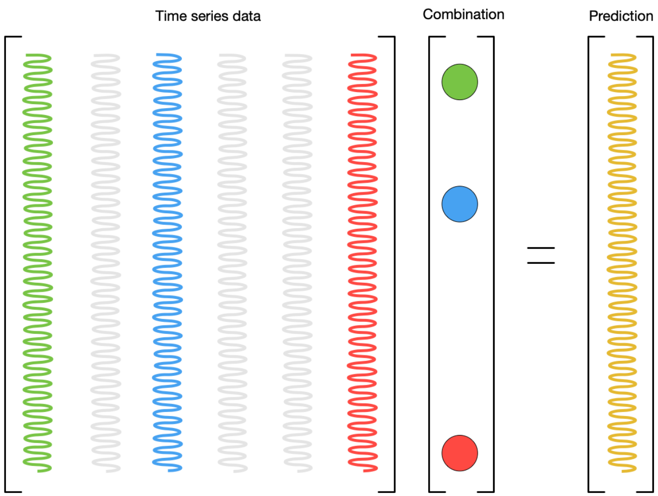
$$\underbrace{\left[\begin{array}{cccc}
u^{\mathsf{d}}(0) & u^{\mathsf{d}}(1) & \cdots & u^{\mathsf{d}}(T-T_{\mathsf{ini}}-T_{\mathsf{f}}) \\
\vdots & \vdots & \ddots & \vdots \\
u^{\mathsf{d}}(T_{\mathsf{ini}}-1) & u^{\mathsf{d}}(T_{\mathsf{ini}}) & \cdots & u^{\mathsf{d}}(T-T_{\mathsf{f}}-1) \\
\hline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

Trajectory Prediction [Markovsky, Rapisarda '08]

Given:

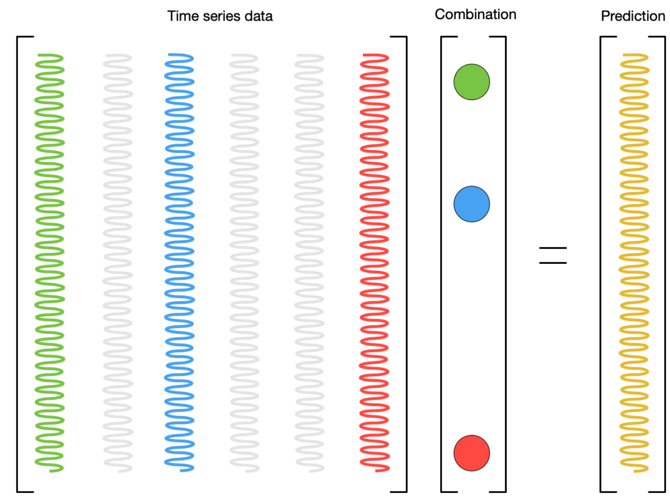
- Data $(u^{\mathsf{d}}_{[0,T-1]}, y^{\mathsf{d}}_{[0,T-1]})$
- Initial trajectory $(u_{\mathsf{ini}}, y_{\mathsf{ini}}) \in \mathbb{R}^{(m+p)T_{\mathsf{ini}}}$
- Future input $u \in \mathbb{R}^{mT_{\mathsf{f}}}$

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



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

Trajectory Prediction [Markovsky, Rapisarda '08]



Given:

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

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

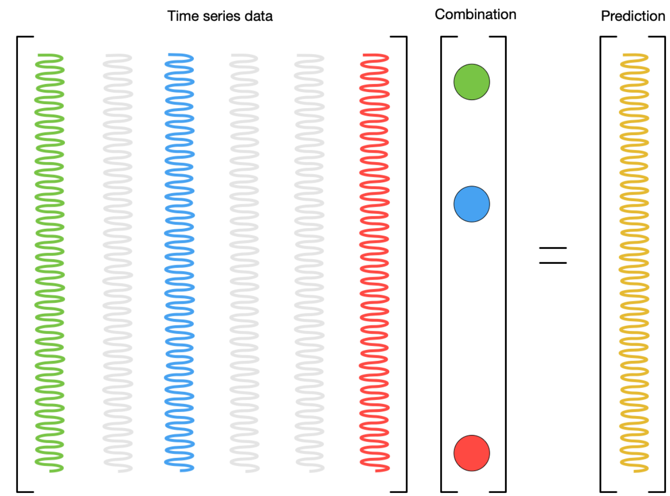
$$\underbrace{\left[\begin{array}{cccc}
u^{\mathsf{d}}(0) & u^{\mathsf{d}}(1) & \cdots & u^{\mathsf{d}}(T-T_{\mathsf{ini}}-T_{\mathsf{f}}) \\
\vdots & \vdots & \ddots & \vdots \\
u^{\mathsf{d}}(T_{\mathsf{ini}}-1) & u^{\mathsf{d}}(T_{\mathsf{ini}}) & \cdots & u^{\mathsf{d}}(T-T_{\mathsf{f}}-1) \\
\hline
y^{\mathsf{d}}(0) & y^{\mathsf{d}}(1) & \cdots & y^{\mathsf{d}}(T-T_{\mathsf{ini}}-T_{\mathsf{f}}) \\
\vdots & \vdots & \ddots & \vdots \\
y^{\mathsf{d}}(T_{\mathsf{ini}}-1) & y^{\mathsf{d}}(T_{\mathsf{ini}}) & \cdots & y^{\mathsf{d}}(T-T_{\mathsf{f}}-1) \\
\hline
u^{\mathsf{d}}(T_{\mathsf{ini}}) & u^{\mathsf{d}}(T_{\mathsf{ini}}+1) & \cdots & u^{\mathsf{d}}(T-T_{\mathsf{f}}) \\
\vdots & \vdots & \ddots & \vdots \\
u^{\mathsf{d}}(T_{\mathsf{ini}}+T_{\mathsf{f}}-1) & u^{\mathsf{d}}(T_{\mathsf{ini}}+T_{\mathsf{f}}) & \cdots & u^{\mathsf{d}}(T-1) \\
\hline
y^{\mathsf{d}}(T_{\mathsf{ini}}) & y^{\mathsf{d}}(T_{\mathsf{ini}}+1) & \cdots & y^{\mathsf{d}}(T-T_{\mathsf{f}}) \\
\vdots & \vdots & \ddots & \vdots \\
y^{\mathsf{d}}(T_{\mathsf{ini}}+T_{\mathsf{f}}-1) & y^{\mathsf{d}}(T_{\mathsf{ini}}+T_{\mathsf{f}}) & \cdots & y^{\mathsf{d}}(T-1)
\end{array}\right]}_{=: \begin{bmatrix} U_{\mathsf{p}} \\ Y_{\mathsf{p}} \\ U_{\mathsf{f}} \\ Y_{\mathsf{f}} \end{bmatrix}}
\quad g = \begin{bmatrix} u_{\mathsf{ini}} \\ y_{\mathsf{ini}} \\ \color{blue}{u} \\ \color{red}{y} \end{bmatrix}
\begin{array}{l} \left.\vphantom{\begin{matrix} a \\ a \end{matrix}}\right\} \text{set initial condition} \\ \left.\vphantom{\begin{matrix} a \\ a \end{matrix}}\right\} \text{prediction} \end{array}$$

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 $y \in \mathbb{R}^{pT_{\mathsf{f}}}$.



$$\underbrace{\left[\begin{array}{cccc}
u^{\mathsf{d}}(0) & u^{\mathsf{d}}(1) & \cdots & u^{\mathsf{d}}(T-T_{\mathsf{ini}}-T_{\mathsf{f}}) \\
\vdots & \vdots & \ddots & \vdots \\
u^{\mathsf{d}}(T_{\mathsf{ini}}-1) & u^{\mathsf{d}}(T_{\mathsf{ini}}) & \cdots & u^{\mathsf{d}}(T-T_{\mathsf{f}}-1) \\ \hline
y^{\mathsf{d}}(0) & y^{\mathsf{d}}(1) & \cdots & y^{\mathsf{d}}(T-T_{\mathsf{ini}}-T_{\mathsf{f}}) \\
\vdots & \vdots & \ddots & \vdots \\
y^{\mathsf{d}}(T_{\mathsf{ini}}-1) & y^{\mathsf{d}}(T_{\mathsf{ini}}) & \cdots & y^{\mathsf{d}}(T-T_{\mathsf{f}}-1) \\ \hline
u^{\mathsf{d}}(T_{\mathsf{ini}}) & u^{\mathsf{d}}(T_{\mathsf{ini}}+1) & \cdots & u^{\mathsf{d}}(T-T_{\mathsf{f}}) \\
\vdots & \vdots & \ddots & \vdots \\
u^{\mathsf{d}}(T_{\mathsf{ini}}+T_{\mathsf{f}}-1) & u^{\mathsf{d}}(T_{\mathsf{ini}}+T_{\mathsf{f}}) & \cdots & u^{\mathsf{d}}(T-1) \\ \hline
y^{\mathsf{d}}(T_{\mathsf{ini}}) & y^{\mathsf{d}}(T_{\mathsf{ini}}+1) & \cdots & y^{\mathsf{d}}(T-T_{\mathsf{f}}) \\
\vdots & \vdots & \ddots & \vdots \\
y^{\mathsf{d}}(T_{\mathsf{ini}}+T_{\mathsf{f}}-1) & y^{\mathsf{d}}(T_{\mathsf{ini}}+T_{\mathsf{f}}) & \cdots & y^{\mathsf{d}}(T-1)
\end{array}\right]}_{=:\begin{bmatrix} U_{\mathsf{p}} \\ Y_{\mathsf{p}} \\ U_{\mathsf{f}} \\ Y_{\mathsf{f}} \end{bmatrix}} g = \begin{bmatrix} u_{\mathsf{ini}} \\ y_{\mathsf{ini}} \\ u \\ y \end{bmatrix}
\begin{array}{l} \left.\vphantom{\begin{matrix} a \\ a \end{matrix}}\right\} \text{set initial condition} \\ \left.\vphantom{\begin{matrix} a \\ a \end{matrix}}\right\} \text{prediction} \end{array}$$

Prediction unique if $T_{\mathsf{ini}} \geq \ell$

Data-enabled Predictive Control

Goal

Using data, design a controller to:

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

Data-enabled Predictive Control

Goal

Using data, design a controller to:

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

DeePC:

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

performance

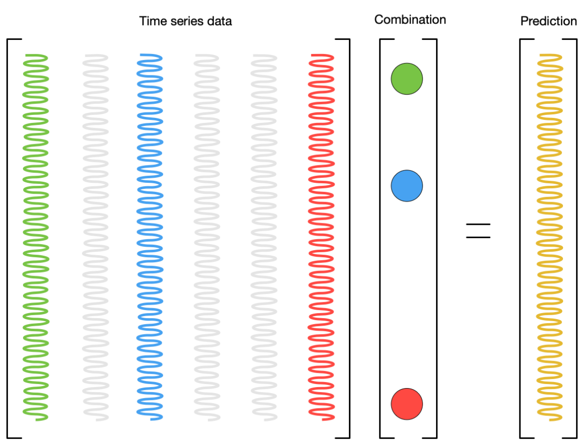
$$\text{s.t.} \quad \begin{bmatrix} U_{\mathsf{p}} \\ Y_{\mathsf{p}} \\ U_{\mathsf{f}} \\ Y_{\mathsf{f}} \end{bmatrix} g = \begin{bmatrix} u_{\mathsf{ini}} \\ y_{\mathsf{ini}} \\ u \\ y \end{bmatrix}$$

initial condition

prediction

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

safety



Models vs data

MPC:

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

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:

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

s.t.

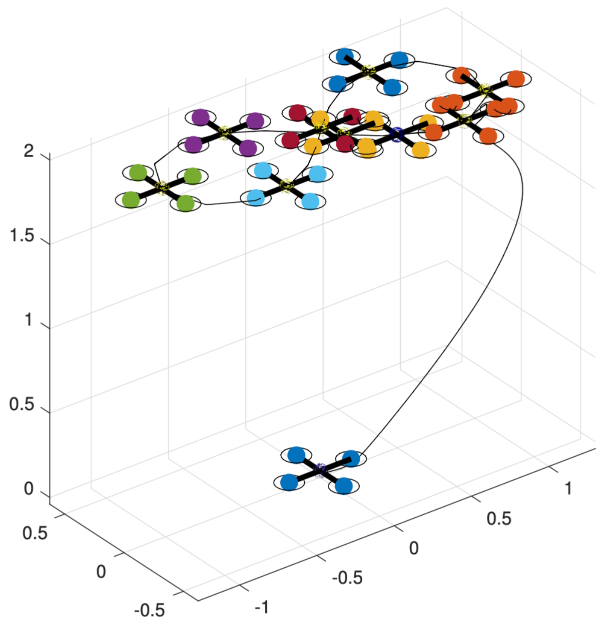
$$\begin{bmatrix} U_{\mathsf{p}} \\ Y_{\mathsf{p}} \\ U_{\mathsf{f}} \\ Y_{\mathsf{f}} \end{bmatrix} g = \begin{bmatrix} u_{\mathsf{ini}} \\ y_{\mathsf{ini}} \\ u \\ y \end{bmatrix}$$
$$u_k \in \mathcal{U}, \ \forall k \in \{0, \dots, T_{\mathsf{f}} - 1\}$$
$$y_k \in \mathcal{Y}, \ \forall k \in \{0,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

Predictive model and state estimation in MPC is replaced by <u>raw data</u> in DeePC!

Consistent for deterministic LTI systems

Theorem [Coulson et al. '19]

MPC and DeePC have equivalent closed-loop behavior.



Stability and recursive feasibility proven by [Berberich et al. '19]

Beyond Deterministic LTI

What about noisy data?
…Nonlinear systems?

Beyond Deterministic LTI

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

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

Robustify against what?

Infeasibility

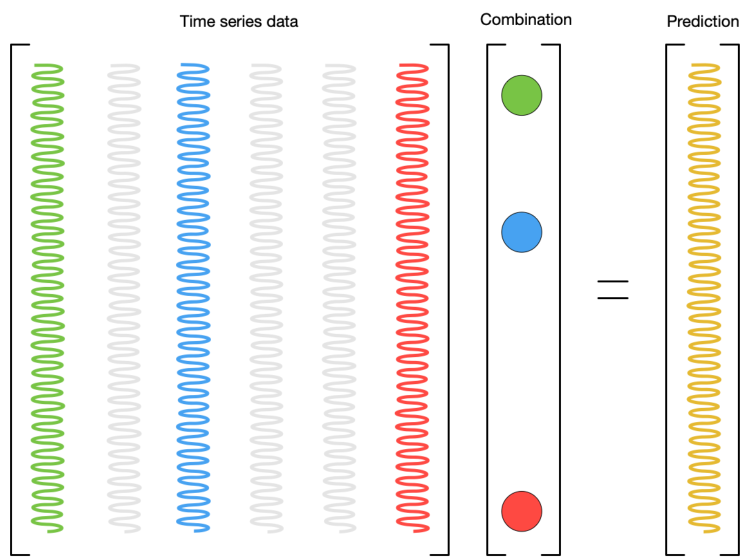
Online data $(u_{\text{ini}}, y_{\text{ini}})$ may be inconsistent with data in the trajectory matrix

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

Over-optimism

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

No longer low rank $\rightarrow$



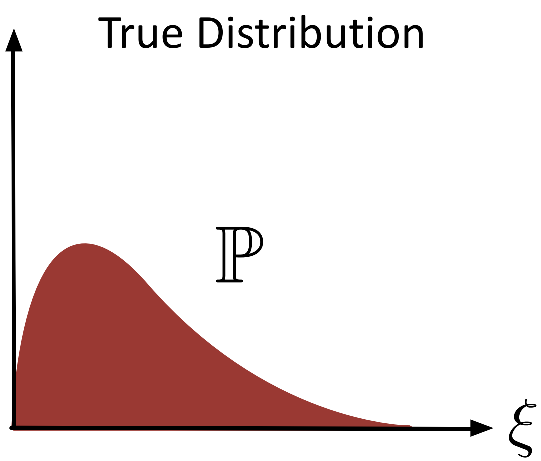
$\leftarrow$ Can predict trajectories outside of subspace of allowable trajectories robustification via regularization

Distributionally Robust Optimization

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

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

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



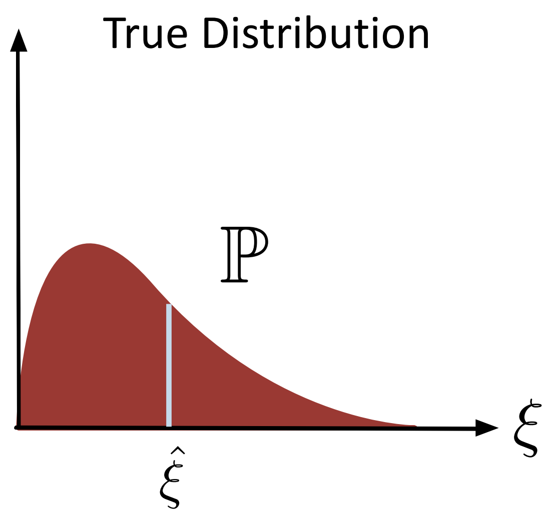
Distributionally Robust Optimization

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

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

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

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



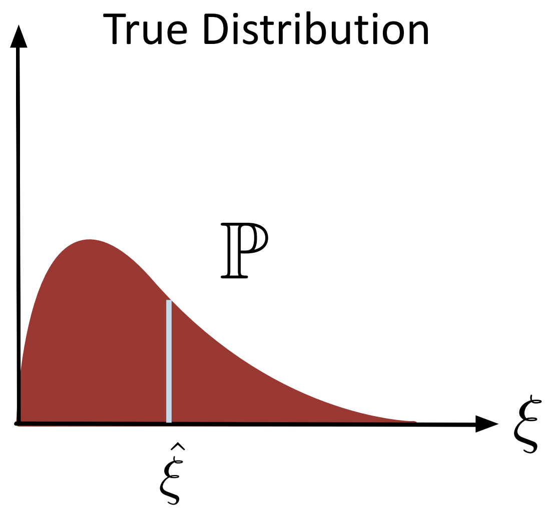
Distributionally Robust Optimization

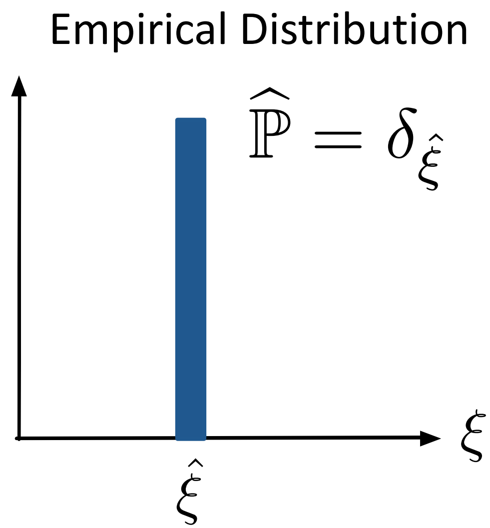
[Bertsimas et al. '04], [Ben-Tal et al. '09], [Kuhn et al. '19]

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

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

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





Distributionally Robust Optimization

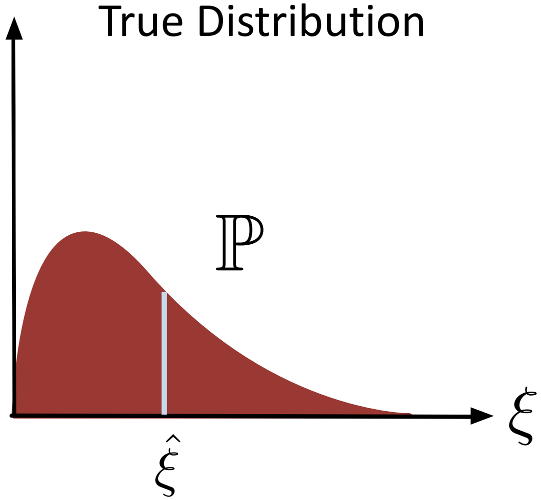
[Bertsimas et al. ‘04], [Ben-Tal et al. ‘09], [Kuhn et al. ‘19]

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

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

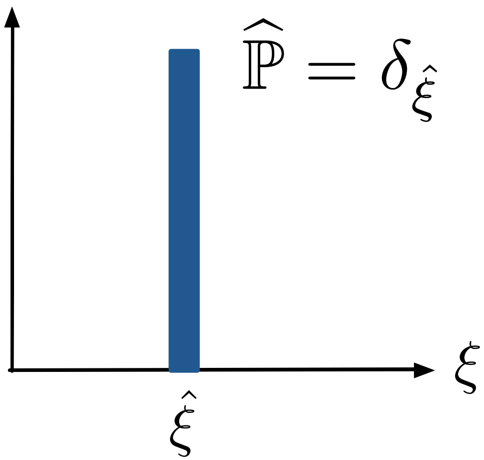
We only have access to measured data sample $\hat{\xi}$

True Distribution



$\neq$

Empirical Distribution



Distributionally Robust Optimization

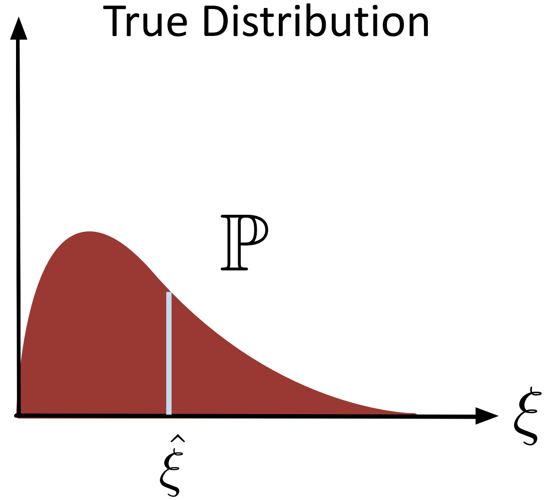
[Bertsimas et al. '04], [Ben-Tal et al. '09], [Kuhn et al. '19]

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

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

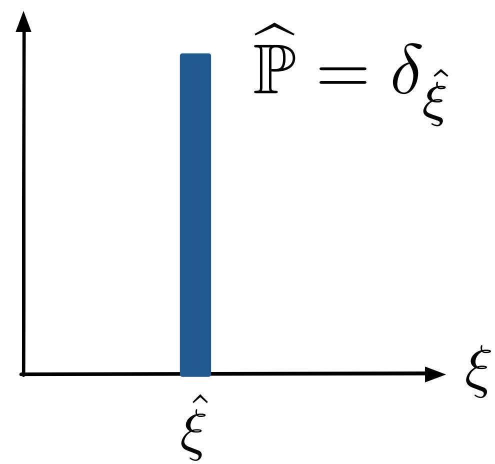
We only have access to measured data sample $\hat{\xi}$

True Distribution



$\neq$

Empirical Distribution



Solve sample average approximation problem

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

Distributionally Robust Optimization

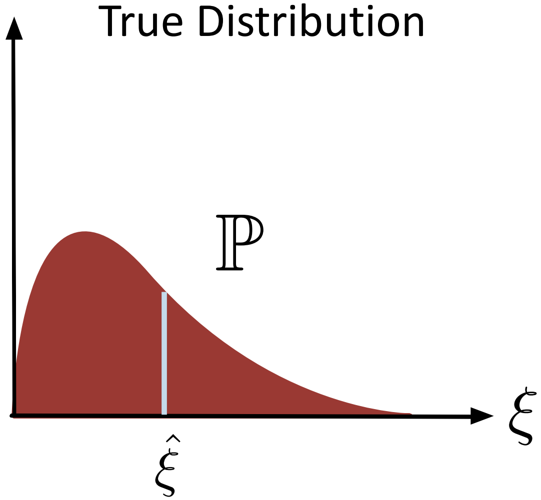
[Bertsimas et al. '04], [Ben-Tal et al. '09], [Kuhn et al. '19]

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

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

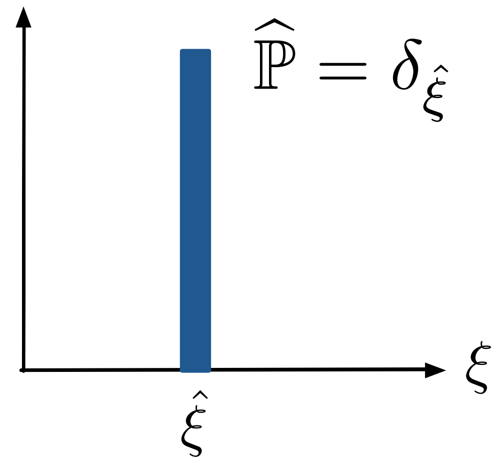
We only have access to measured data sample $\hat{\xi}$

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

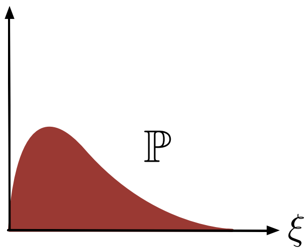
$$\mathbb{E}_{\mathbb{P}}[c(\xi, g^\star)]$$

Wasserstein Ball

[Kuhn et al. '19]

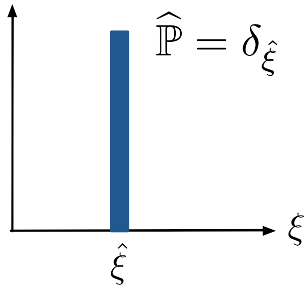
Robustify against the fact that:

True Distribution



$\neq$

Empirical Distribution

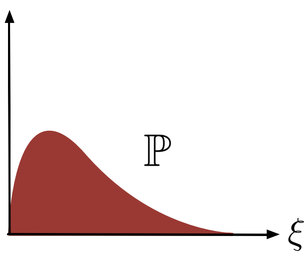


Wasserstein Ball

[Kuhn et al. ‘19]

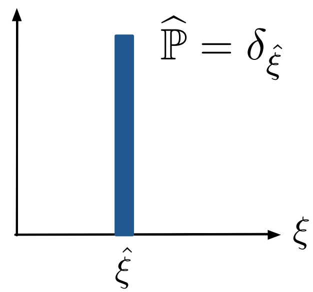
Robustify against the fact that:

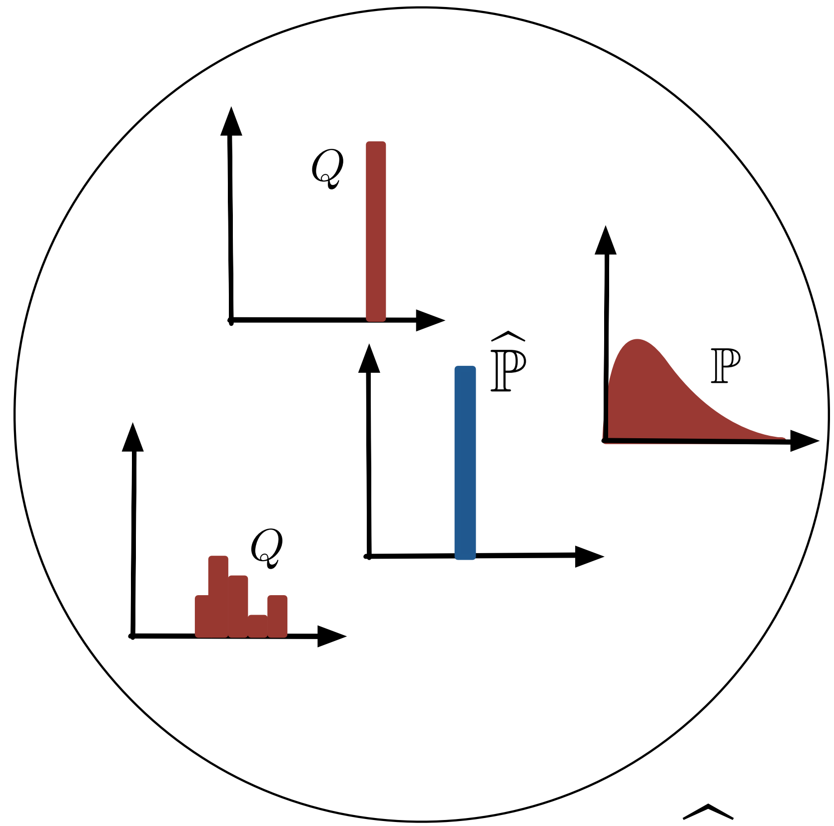
True Distribution



$\neq$

Empirical Distribution





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

Distributionally Robust Optimization

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

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

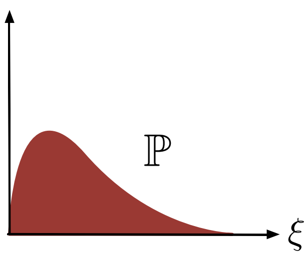
is the Wasserstein ball.

Wasserstein Ball

[Kuhn et al. '19]

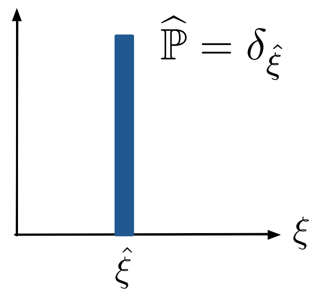
Robustify against the fact that:

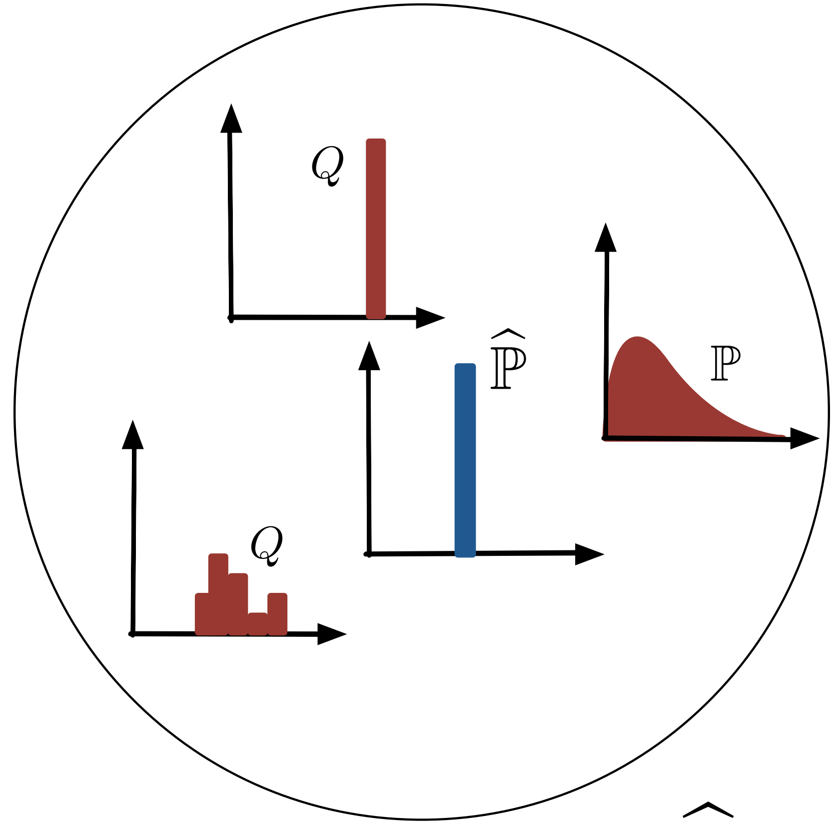
True Distribution



$\neq$

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

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

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

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

Abstracted DeePC

DeePC + slack

$$\begin{aligned}\min_{g,u,y} \quad & \sum_{k=0}^{T_{\mathsf{f}}-1} f(u_k, y_k) + h(\sigma_y) \\ \text{s.t.} \quad & \begin{bmatrix} U_{\mathsf{p}} \\ Y_{\mathsf{p}} \\ U_{\mathsf{f}} \\ Y_{\mathsf{f}} \end{bmatrix} g = \begin{bmatrix} u_{\mathsf{ini}} \\ y_{\mathsf{ini}} + \sigma_y \\ u \\ y \end{bmatrix} \\ & u_k \in \mathcal{U}, \ \forall k \in \{0, \ldots, T_{\mathsf{f}} - 1\}\end{aligned}$$

Abstracted DeePC

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

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

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

Abstracted DeePC

DeePC + slack

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

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

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

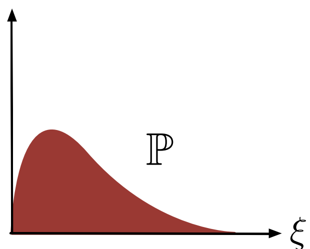
Abstracted DeePC

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

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

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

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



True Distribution

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

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

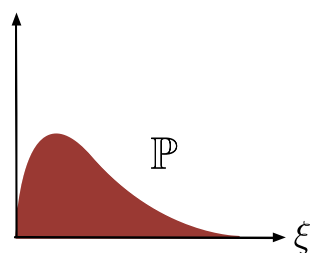
Data $\hat{\xi}$ is a particular measurement (realization) of random variable ξ distributed according to unknown distribution $\mathbb{P}$.

What we care about:

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

What DeePC is solving:

$$\min_{g \in G} \ c(\hat{\xi}, g)$$



True Distribution

Abstracted DeePC

DeePC + slack

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

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

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

Abstracted DeePC

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

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

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

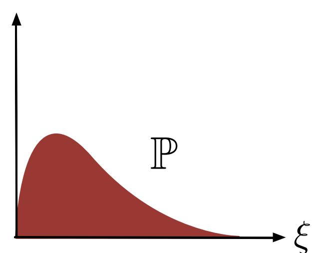
Data $\hat{\xi}$ is a particular measurement (realization) of random variable ξ distributed according to unknown distribution $\mathbb{P}$.

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

Abstracted DeePC

DeePC + slack

$$
\begin{aligned}
\min_{g,u,y} \quad & \sum_{k=0}^{T_{\mathsf{f}}-1} f(u_k, y_k) + h(\sigma_y) \\
\text{s.t.} \quad & \begin{bmatrix} U_{\mathsf{p}} \\ Y_{\mathsf{p}} \\ U_{\mathsf{f}} \\ Y_{\mathsf{f}} \end{bmatrix} g = \begin{bmatrix} u_{\mathsf{ini}} \\ y_{\mathsf{ini}} + \sigma_y \\ u \\ y \end{bmatrix} \\
& u_k \in \mathcal{U}, \ \forall k \in \{0, \ldots, T_{\mathsf{f}} - 1\}
\end{aligned}
$$

Abstracted DeePC

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

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

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

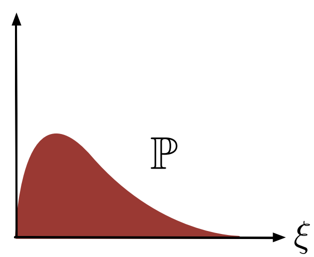
Data $\hat{\xi}$ is a particular measurement (realization) of random variable ξ distributed according to unknown distribution $\mathbb{P}$.

What we care about:

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

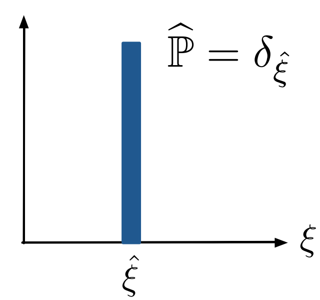
What DeePC is solving:

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



True Distribution

$\neq$



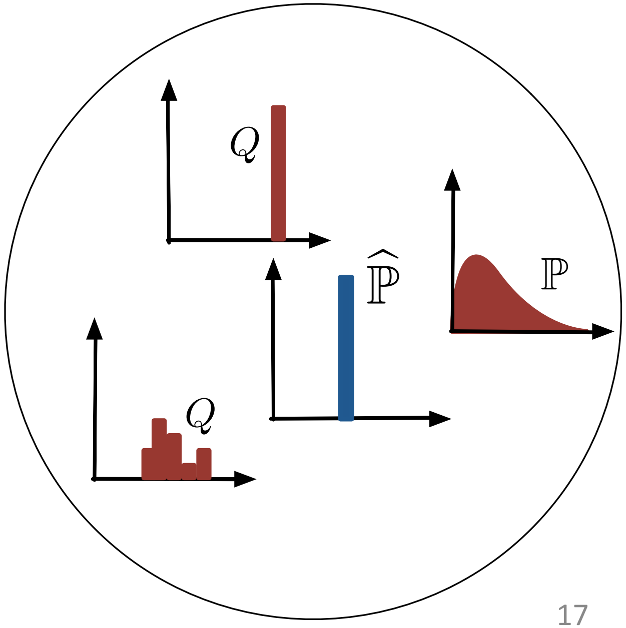
Empirical Distribution

Distributionally Robust DeePC

Theorem [Coulson et al. '21]

Under minor technical conditions

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



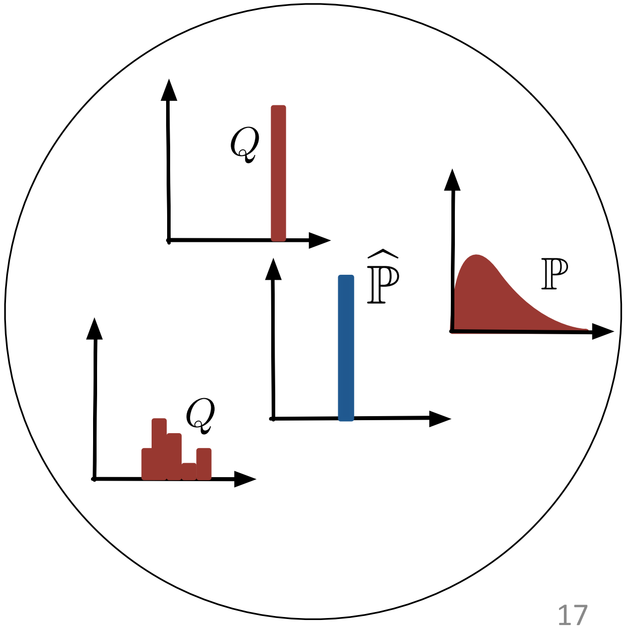
Proof uses methods from [Mohajerin Esfahani, Kuhn '18]

Distributionally Robust DeePC

Theorem [Coulson et al. '21]

Under minor technical conditions

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



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

Distributionally Robust DeePC

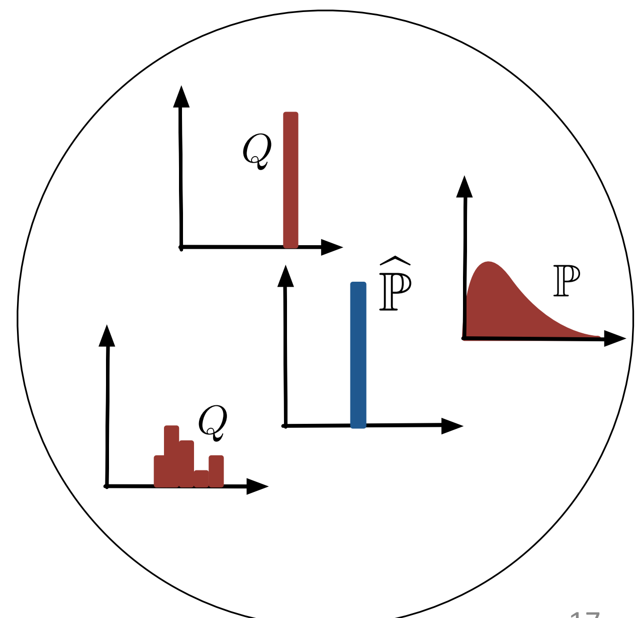
Theorem [Coulson et al. '21]

Under minor technical conditions

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

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

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



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

Distributionally Robust DeePC

Theorem [Coulson et al. '21]

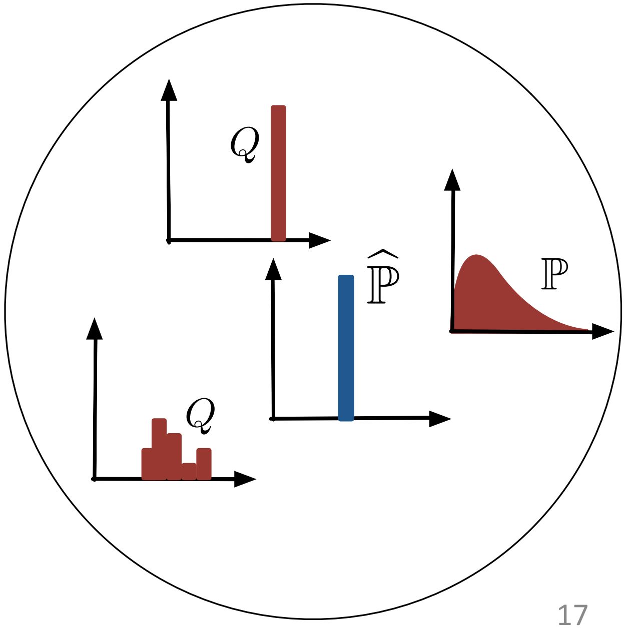
Under minor technical conditions

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

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

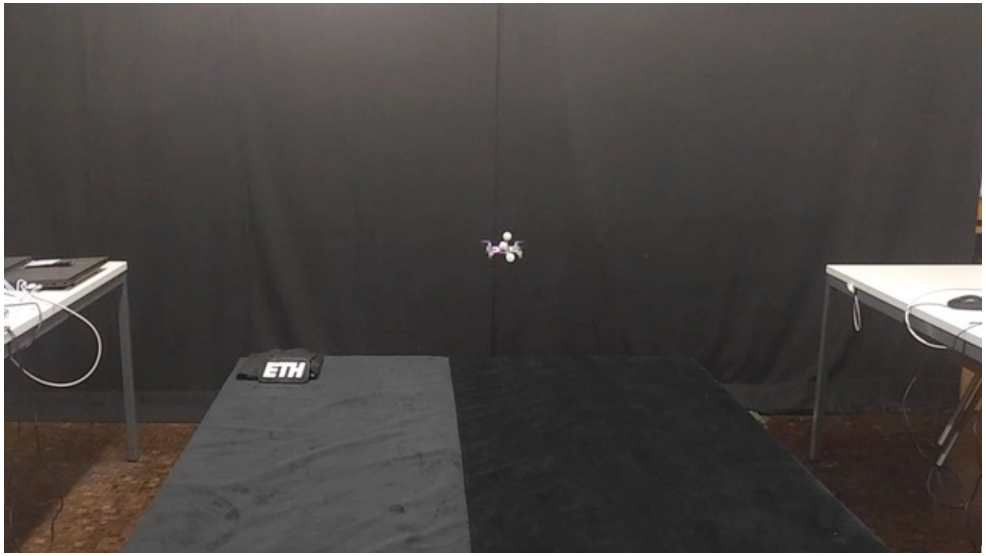
$$\frac{1}{p}+\frac{1}{q}=1$$

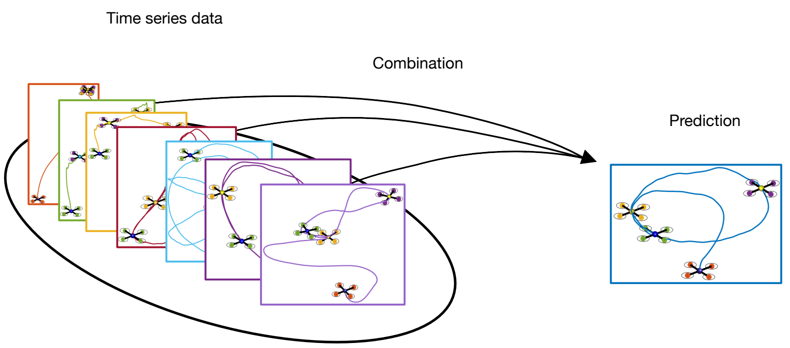
The Wasserstein ball contains more than just LTI systems with additive noise

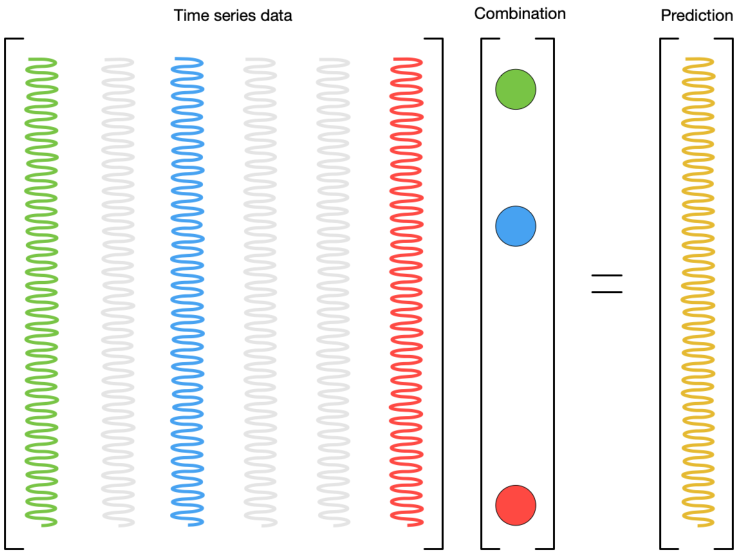


Proof uses methods from [Mohajerin Esfahani, Kuhn '18]

Into the real world [Elokda, Coulson et al. '21]



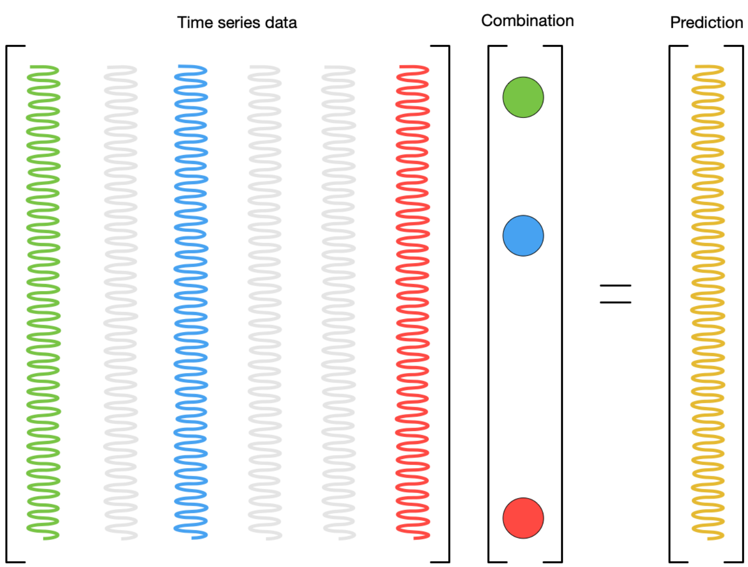


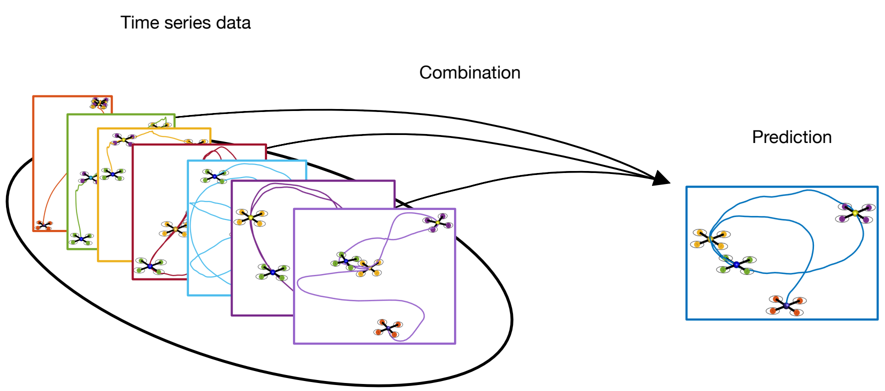


Summary

- Matrix of time-series data is a predictive model
- DeePC equivalent to MPC for deterministic LTI
- Distributionally robustify DeePC through regularizations to extend beyond deterministic LTI
- Successful deployment on many experiments and simulations

Future work: online + adaptive guarantees





Thanks!

UNIVERSITY OF WISCONSIN–MADISON

jeremy.coulson@wisc.edu