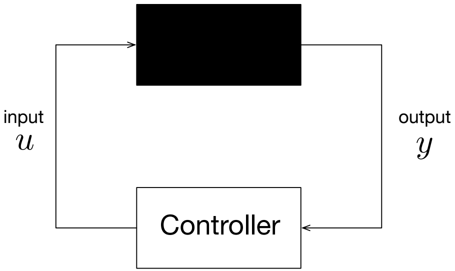
Regularized & Distributionally Robust Data-enabled Predictive Control

Jeremy Coulson, John Lygeros, Florian Dörfler

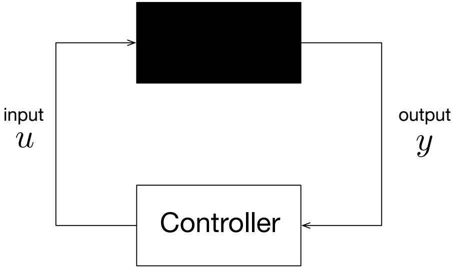
Learning Sparse Models Workshop

What can one do with a black box?



Question: How should I design a controller?

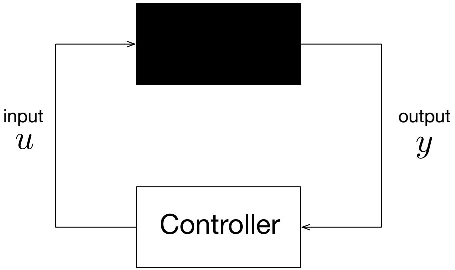
What can one do with a black box?



Question: How should I design a controller?

collect data

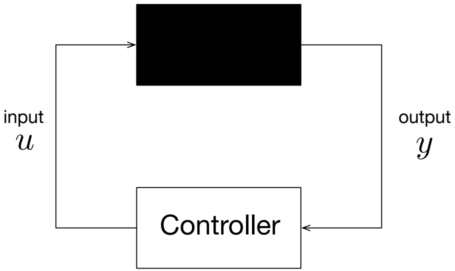
What can one do with a black box?



Question: How should I design a controller?

collect data $\longrightarrow$ **identify model**

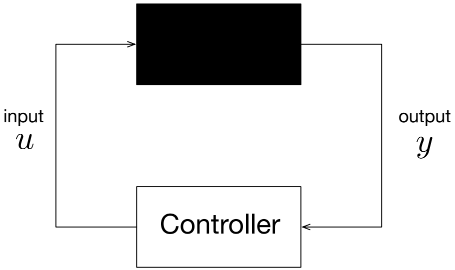
What can one do with a black box?



Question: How should I design a controller?

collect data $\longrightarrow$ **identify model** $\longrightarrow$ **design controller**

What can one do with a black box?

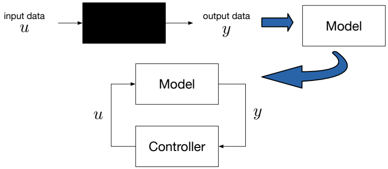


Question: How should I design a controller?

collect data $\longrightarrow$ **identify model** $\longrightarrow$ **design controller**

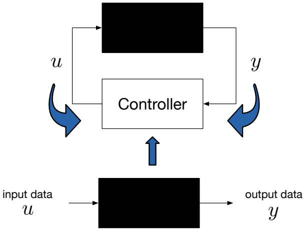
“Why learn a model if we only care about control?”

Direct vs. Indirect Data-driven Control



Indirect data-driven control

- ✓ Quantify uncertainty + design robust controller
- ✗ Sys ID very expensive
- ✗ Sys ID seeks best model that fits data...not best for control



Direct data-driven control

- ✓ Impressive recent theoretical & practical advances
- ✗ Often requires a lot of data and brute-force computation
- ✗ Not suitable for real-time safety critical system

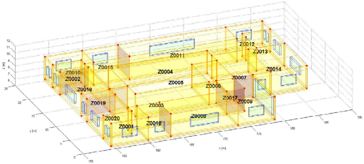
Why direct data-driven control?

Question: When should one use direct data-driven control?

Why direct data-driven control?

Question: When should one use direct data-driven control?

- First-principle models **not conceivable** (e.g., human-in-the-loop, biology)
- Models **too complex** for control design (e.g., fluids, building automation)
- Thorough modelling **too costly** (e.g., robotics)
- Often **easier to learn control policies** directly from data (e.g., PID)



Outline

1. DeePC (Basic Idea):

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

Jeremy Coulson John Lygeros Florian Dörfler

2. Distributionally Robust DeePC:

Distributionally Robust Chance Constrained Data-enabled Predictive Control

Jeremy Coulson John Lygeros Florian Dörfler

3. Application:

Data-Enabled Predictive Control for Quadcopters

Ezzat Elokda | Jeremy Coulson* | Paul N. Beuchat | John Lygeros | Florian Dörfler

Problem Statement

Consider the **controllable** LTI system

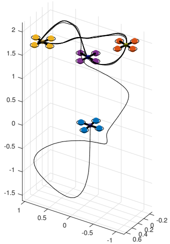
$$\begin{cases} x(t+1) = Ax(t) + Bu(t) & t \in \mathbb{Z}_{\geq 0} \\ y(t) = Cx(t) + Du(t), \end{cases}$$

where

- $x(t) \in \mathbb{R}^n$ is the **state**
- $u(t) \in \mathbb{R}^m$ is the **control input**
- $y(t) \in \mathbb{R}^p$ is the **output**
- $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$, $D \in \mathbb{R}^{p \times m}$ are **unknown**

Problem Statement

Consider the **controllable** LTI system



$$\begin{cases} x(t+1) = Ax(t) + Bu(t) \quad t \in \mathbb{Z}_{\geq 0} \\ y(t) = Cx(t) + Du(t), \end{cases}$$

where

- $x(t) \in \mathbb{R}^n$ is the **state**
- $u(t) \in \mathbb{R}^m$ is the **control input**
- $y(t) \in \mathbb{R}^p$ is the **output**
- $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$, $D \in \mathbb{R}^{p \times m}$ are **unknown**

Goal: design controller to

- **track** a reference output trajectory $r = (r_0, r_1, \dots) \in (\mathbb{R}^p)^{\mathbb{Z}_{\geq 0}}$
- **satisfy input/output constraints** $u(t) \in \mathcal{U} \subseteq \mathbb{R}^m$, $y(t) \in \mathcal{Y} \subseteq \mathbb{R}^p \ \forall t$

Behavioural System Theory

Jan Willems

Introduced behavioural system theory $\sim$1979

"The behaviour is all there is"

Behavioural System Theory

Jan Willems
Introduced behavioural system theory $\sim$1979
"The behaviour is all there is"

- LTI system defined by its **"behaviour"**

$$\mathscr{B} \subseteq \left(\mathbb{R}^{m+p}\right)^{\mathbb{Z}_{\geq 0}}$$

- $\mathscr{B}$ is subspace containing **trajectories** $(u, y) = (u_0, y_0, u_1, y_1, \dots)$.
- The set of **truncated trajectories** is

$$\mathscr{B}_T = \text{restriction of } \mathscr{B} \text{ to } t \in [0, T].$$

Persistency of Excitation

Definition

Let $T, T_\mathrm{f} \in \mathbb{Z}_{\geq 1}$ such that $T \geq T_\mathrm{f}$. The signal $u = \mathrm{col}(u_1, \ldots, u_T) \in \mathbb{R}^{Tm}$ is **persistently exciting of order T_f** if the Hankel matrix

$$\mathscr{H}_{T_\mathrm{f}}(u) \triangleq \begin{pmatrix} u_1 & u_2 & \cdots & u_{T-T_\mathrm{f}+1} \\ u_2 & u_3 & \cdots & u_{T-T_\mathrm{f}+2} \\ \vdots & \vdots & \ddots & \vdots \\ u_{T_\mathrm{f}} & u_{T_\mathrm{f}+1} & \cdots & u_T \end{pmatrix}$$

is of full row rank.

"Signal is sufficiently rich and long ($T - T_\mathrm{f} + 1 \geq T_\mathrm{f} m$)"

Fundamental Lemma

Lemma (Fundamental Lemma, Willems, et al, 2005)

Let $T, T_{\mathrm{f}} \in \mathbb{Z}_{\geq 1}$. *Consider*

- *controllable discrete-time LTI system* $\mathscr{B}$
- *Data trajectory* $\mathrm{col}(\hat{u}, \hat{y}) \in \mathscr{B}_T$ *such that*
- $\hat{u}$ *persistently exciting of order* $T_{\mathrm{f}} + n$ *(*n *is #states)*

Then

$$\mathrm{colspan}\left(\mathscr{H}_{T_{\mathrm{f}}}\begin{pmatrix}\hat{u}\\ \hat{y}\end{pmatrix}\right) = \mathscr{B}_{T_{\mathrm{f}}}.$$

"All trajectories can be reconstructed from finitely many, sufficiently rich previous trajectories"

Fundamental Lemma

Lemma (Fundamental Lemma, Willems, et al, 2005)

Let $T, T_{\mathrm{f}} \in \mathbb{Z}_{\geq 1}$. *Consider*

- *controllable discrete-time LTI system* $\mathscr{B}$
- *Data trajectory* $\operatorname{col}(\hat{u}, \hat{y}) \in \mathscr{B}_T$ *such that*
- $\hat{u}$ *persistently exciting of order* $T_{\mathrm{f}} + n$ *(*n *is #states)*

Then

$$\operatorname{colspan}\left(\mathscr{H}_{T_{\mathrm{f}}}\begin{pmatrix}\hat{u}\\ \hat{y}\end{pmatrix}\right) = \mathscr{B}_{T_{\mathrm{f}}}.$$

"All trajectories can be reconstructed from finitely many, sufficiently rich previous trajectories"

Idea: The Hankel matrix using **raw data** can serve as a **predictive model**!

Hankel Matrix Example

Assume col$(\hat{u}, \hat{y}) = (\hat{u}_1, \hat{y}_1, \ldots, \hat{u}_T, \hat{y}_T) \in \mathscr{B}_T$ and $\hat{u}$ persistently exciting of order $T_{\mathrm{f}} + n$.

Hankel Matrix Example

Assume col$(\hat{u}, \hat{y}) = (\hat{u}_1, \hat{y}_1, \ldots, \hat{u}_T, \hat{y}_T) \in \mathscr{B}_T$ and $\hat{u}$ persistently exciting of order $T_{\mathrm{f}} + n$.

$$\mathscr{H}_{T_{\mathrm{f}}} \begin{pmatrix} \hat{u} \\ \hat{y} \end{pmatrix}$$

Hankel Matrix Example

Assume col$(\hat{u}, \hat{y}) = (\hat{u}_1, \hat{y}_1, \ldots, \hat{u}_T, \hat{y}_T) \in \mathscr{B}_T$ and $\hat{u}$ persistently exciting of order $T_{\mathrm{f}} + n$.

$$\mathscr{H}_{T_{\mathrm{f}}} \begin{pmatrix} \hat{u} \\ \hat{y} \end{pmatrix} g$$

Hankel Matrix Example

Assume col$(\hat{u}, \hat{y}) = (\hat{u}_1, \hat{y}_1, \ldots, \hat{u}_T, \hat{y}_T) \in \mathscr{B}_T$ and $\hat{u}$ persistently exciting of order $T_{\mathrm{f}} + n$.

$$\mathscr{H}_{T_{\mathrm{f}}} \begin{pmatrix} \hat{u} \\ \hat{y} \end{pmatrix} g = \begin{pmatrix} u \\ y \end{pmatrix}$$

Hankel Matrix Example

Assume $\mathrm{col}(\hat{u}, \hat{y}) = (\hat{u}_1, \hat{y}_1, \ldots, \hat{u}_T, \hat{y}_T) \in \mathscr{B}_T$ and $\hat{u}$ persistently exciting of order $T_\mathrm{f} + n$.

$$\mathscr{H}_{T_\mathrm{f}} \begin{pmatrix} \hat{u} \\ \hat{y} \end{pmatrix} g = \begin{pmatrix} u \\ y \end{pmatrix}$$

- Given input $u = (u_1, \ldots, u_{T_\mathrm{f}})$, **predict** output $y = (y_1, \ldots, y_{T_\mathrm{f}})$

Hankel Matrix Example

Assume col$(\hat{u}, \hat{y}) = (\hat{u}_1, \hat{y}_1, \ldots, \hat{u}_T, \hat{y}_T) \in \mathscr{B}_T$ and $\hat{u}$ persistently exciting of order $T_\mathrm{f} + n$.

$$\mathscr{H}_{T_\mathrm{f}} \begin{pmatrix} \hat{u} \\ \hat{y} \end{pmatrix} g = \begin{pmatrix} u \\ y \end{pmatrix}$$

- Given input $u = (u_1, \ldots, u_{T_\mathrm{f}})$, **predict** output $y = (y_1, \ldots, y_{T_\mathrm{f}})$

Issue: Predicted output not unique!

Hankel Matrix Example ctd.

Assume col$(\hat{u}, \hat{y}) = (\hat{u}_1, \hat{y}_1, \ldots, \hat{u}_T, \hat{y}_T) \in \mathscr{B}_T$ and $\hat{u}$ persistently exciting of order $T_{\text{ini}} + T_{\text{f}} + n$ and $(\hat{u}_{\text{ini}}, \hat{y}_{\text{ini}}) \in \mathscr{B}_{T_{\text{ini}}}$.

$$\mathscr{H}_{T_{\text{ini}}+T_{\text{f}}} \begin{pmatrix} \hat{u} \\ \hat{y} \end{pmatrix} g = \begin{pmatrix} \hat{u}_{\text{ini}} \\ \hat{y}_{\text{ini}} \\ u \\ y \end{pmatrix}$$

- Given input $u = (u_1, \ldots, u_{T_{\text{f}}})$, **predict** output $y = (y_1, \ldots, y_{T_{\text{f}}})$

Hankel Matrix Example ctd.

Assume col$(\hat{u}, \hat{y}) = (\hat{u}_1, \hat{y}_1, \ldots, \hat{u}_T, \hat{y}_T) \in \mathscr{B}_T$ and $\hat{u}$ persistently exciting of order $T_{\text{ini}} + T_{\text{f}} + n$ and $(\hat{u}_{\text{ini}}, \hat{y}_{\text{ini}}) \in \mathscr{B}_{T_{\text{ini}}}$.

$$\mathscr{H}_{T_{\text{ini}}+T_{\text{f}}} \begin{pmatrix} \hat{u} \\ \hat{y} \end{pmatrix} g = \begin{pmatrix} \hat{u}_{\text{ini}} \\ \hat{y}_{\text{ini}} \\ u \\ y \end{pmatrix} \begin{matrix} \left.\begin{matrix} \\ \\ \end{matrix}\right\} \text{set initial condition} \\ \left.\begin{matrix} \\ \\ \end{matrix}\right\} \text{prediction} \end{matrix}$$

- Given input $u = (u_1, \ldots, u_{T_{\text{f}}})$, **predict** output $y = (y_1, \ldots, y_{T_{\text{f}}})$

Hankel Matrix Example ctd.

Assume col$(\hat{u}, \hat{y}) = (\hat{u}_1, \hat{y}_1, \ldots, \hat{u}_T, \hat{y}_T) \in \mathscr{B}_T$ and $\hat{u}$ persistently exciting of order $T_{\text{ini}} + T_{\text{f}} + n$ and $(\hat{u}_{\text{ini}}, \hat{y}_{\text{ini}}) \in \mathscr{B}_{T_{\text{ini}}}$.

$$\mathscr{H}_{T_{\text{ini}}+T_{\text{f}}} \begin{pmatrix} \hat{u} \\ \hat{y} \end{pmatrix} g = \begin{pmatrix} \hat{u}_{\text{ini}} \\ \hat{y}_{\text{ini}} \\ u \\ y \end{pmatrix} \begin{matrix} \left.\vphantom{\begin{matrix} a \\ b \end{matrix}}\right\} \textbf{set initial condition} \\ \left.\vphantom{\begin{matrix} a \\ b \end{matrix}}\right\} \textbf{prediction} \end{matrix}$$

- Given input $u = (u_1, \ldots, u_{T_{\text{f}}})$, **predict** output $y = (y_1, \ldots, y_{T_{\text{f}}})$

When $T_{\text{ini}} \geq$ **lag** of system, the predicted output is **unique**[1].

[1] I. Markovsky and P. Rapisarda, 2008

Model Predictive Control

Goal: design controller to

- **track** a reference output trajectory $r = (r_0, r_1, \dots) \in (\mathbb{R}^p)^{\mathbb{Z}_{\geq 0}}$
- **satisfy input/output constraints** $u(t) \in \mathcal{U} \subseteq \mathbb{R}^m$, $y(t) \in \mathcal{Y} \subseteq \mathbb{R}^p \ \forall t$

When state space model (i.e., A, B, C, D) **known**:

Model Predictive Control

Goal: design controller to

- **track** a reference output trajectory $r = (r_0, r_1, \dots) \in (\mathbb{R}^p)^{\mathbb{Z}_{\geq 0}}$
- **satisfy input/output constraints** $u(t) \in \mathcal{U} \subseteq \mathbb{R}^m$, $y(t) \in \mathcal{Y} \subseteq \mathbb{R}^p \; \forall t$

When state space model (i.e., A, B, C, D) **known**:
MPC:

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

Data-enabled Predictive Control Algorithm

Goal: design controller to

- **track** a reference output trajectory $r = (r_0, r_1, \dots) \in (\mathbb{R}^p)^{\mathbb{Z}_{\geq 0}}$
- **satisfy input/output constraints** $u(t) \in \mathcal{U} \subseteq \mathbb{R}^m$, $y(t) \in \mathcal{Y} \subseteq \mathbb{R}^p \ \forall t$

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

Data-enabled Predictive Control Algorithm

Goal: design controller to

- **track** a reference output trajectory $r = (r_0, r_1, \dots) \in (\mathbb{R}^p)^{\mathbb{Z}_{\geq 0}}$
- **satisfy input/output constraints** $u(t) \in \mathcal{U} \subseteq \mathbb{R}^m$, $y(t) \in \mathcal{Y} \subseteq \mathbb{R}^p \ \forall t$

When state space model (i.e., A, B, C, D) **unknown**:
DeePC:

$$\begin{aligned}
\underset{g,u,y}{\text{minimize}} \quad & \sum_{k=0}^{T_\mathrm{f}-1} \left(\|y_k - r_{t+k}\|_Q^2 + \|u_k\|_R^2 \right) \\
\text{subject to} \quad & \mathscr{H}_{T_\mathrm{ini}+T_\mathrm{f}} \begin{pmatrix} \hat{u} \\ \hat{y} \end{pmatrix} g = \begin{pmatrix} \hat{u}_\mathrm{ini} \\ \hat{y}_\mathrm{ini} \\ u \\ y \end{pmatrix}, \\
& u_k \in \mathcal{U}, \ \forall k \in \{0, \dots, T_\mathrm{f} - 1\}, \\
& y_k \in \mathcal{Y}, \ \forall k \in \{0, \dots, T_\mathrm{f} - 1\}.
\end{aligned}$$

What's the difference?

MPC:

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

DeePC:

$$
\begin{aligned}
\underset{g,u,y}{\text{minimize}} \quad & \sum_{k=0}^{T_{\mathrm{f}}-1} \left(\left\| y_k - r_{t+k} \right\|_Q^2 + \|u_k\|_R^2 \right) \\
\text{subject to} \quad & \mathscr{H}_{T_{\mathrm{ini}}+T_{\mathrm{f}}} \begin{pmatrix} \hat{u} \\ \hat{y} \end{pmatrix} g = \begin{pmatrix} \hat{u}_{\mathrm{ini}} \\ \hat{y}_{\mathrm{ini}} \\ u \\ y \end{pmatrix}, \\
& u_k \in \mathcal{U}, \ \forall k \in \{0, \dots, T_{\mathrm{f}} - 1\}, \\
& y_k \in \mathcal{Y}, \ \forall k \in \{0, \dots, T_{\mathrm{f}} - 1\}.
\end{aligned}
$$

Predictive model and state estimation in MPC is replaced by raw data in a Hankel matrix in DeePC.

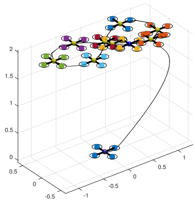
Consistent for Deterministic LTI systems

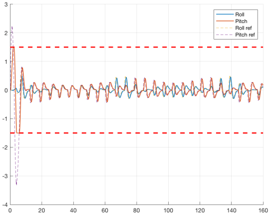
Theorem

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

Corollary

If $\mathcal{U}$, $\mathcal{Y}$ are ***convex****, then* ***closed-loop trajectories coincide***.





"MPC and DeePC have equivalent closed loop behaviour"

Beyond Deterministic LTI

What about noisy data?
...Nonlinear systems?

Beyond Deterministic LTI

What about noisy data?
...Nonlinear systems?

We need a robustified approach!

Regularizations

1. Online data $(\hat{u}_{\text{ini}}, \hat{y}_{\text{ini}})$ inconsistent with data in Hankel matrix
2. Offline data $\mathscr{H}_{T_{\text{ini}}+T_{\text{f}}}\begin{pmatrix}\hat{u}\\ \hat{y}\end{pmatrix}$ noisy $\implies$ data matrix full rank (can predict anything)

$$\begin{aligned}\underset{g,u,y,\sigma_y}{\text{minimize}} \quad & \sum_{k=0}^{T_{\text{f}}-1}\left(\left\|y_k - r_{t+k}\right\|_Q^2 + \|u_k\|_R^2\right) + \lambda_y\|\sigma_y\|_p + \lambda_g\|g\|_1 \\ \text{subject to} \quad & \mathscr{H}_{T_{\text{ini}}+T_{\text{f}}}\begin{pmatrix}\hat{u}\\ \hat{y}\end{pmatrix} g = \begin{pmatrix}\hat{u}_{\text{ini}}\\ \hat{y}_{\text{ini}}\\ u\\ y\end{pmatrix} + \begin{pmatrix}0\\ \sigma_y\\ 0\\ 0\end{pmatrix}, \\ & u_k \in \mathcal{U},\ \forall k \in \{0,\ldots,T_{\text{f}}-1\}, \\ & y_k \in \mathcal{Y},\ \forall k \in \{0,\ldots,T_{\text{f}}-1\}.\end{aligned}$$

Regularizations

1. Online data $(\hat{u}_{\text{ini}}, \hat{y}_{\text{ini}})$ inconsistent with data in Hankel matrix
2. Offline data $\mathscr{H}_{T_{\text{ini}}+T_{\text{f}}}\begin{pmatrix}\hat{u}\\ \hat{y}\end{pmatrix}$ noisy $\implies$ data matrix full rank (can predict anything)

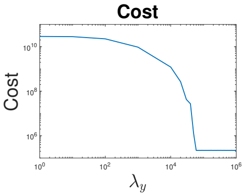
$$
\begin{aligned}
\underset{g,u,y,\sigma_y}{\text{minimize}} \quad & \sum_{k=0}^{T_{\text{f}}-1}\left(\left\|y_k - r_{t+k}\right\|_Q^2 + \|u_k\|_R^2\right) + \lambda_y\|\sigma_y\|_p + \lambda_g\|g\|_1 \\
\text{subject to} \quad & \mathscr{H}_{T_{\text{ini}}+T_{\text{f}}}\begin{pmatrix}\hat{u}\\ \hat{y}\end{pmatrix} g = \begin{pmatrix}\hat{u}_{\text{ini}}\\ \hat{y}_{\text{ini}}\\ u \\ y\end{pmatrix} + \begin{pmatrix}0\\ \sigma_y\\ 0\\ 0\end{pmatrix}, \\
& u_k \in \mathcal{U},\ \forall k \in \{0,\dots,T_{\text{f}}-1\}, \\
& y_k \in \mathcal{Y},\ \forall k \in \{0,\dots,T_{\text{f}}-1\}.
\end{aligned}
$$

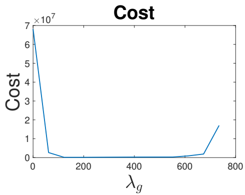
1-norm promotes sparsity $\iff$ sparse selection of **motion primitives**

Regularizations

1. Online data $(\hat{u}_{\text{ini}}, \hat{y}_{\text{ini}})$ inconsistent with data in Hankel matrix
2. Offline data $\mathscr{H}_{T_{\text{ini}}+T_{\text{f}}}\begin{pmatrix}\hat{u}\\ \hat{y}\end{pmatrix}$ noisy $\implies$ data matrix full rank (can predict anything)

$$\begin{aligned}
\underset{g,u,y,\sigma_y}{\text{minimize}} \quad & \sum_{k=0}^{T_{\text{f}}-1}\left(\left\|y_k - r_{t+k}\right\|_Q^2 + \|u_k\|_R^2\right) + \lambda_y\|\sigma_y\|_p + \lambda_g\|g\|_1 \\
\text{subject to} \quad & \mathscr{H}_{T_{\text{ini}}+T_{\text{f}}}\begin{pmatrix}\hat{u}\\ \hat{y}\end{pmatrix} g = \begin{pmatrix}\hat{u}_{\text{ini}}\\ \hat{y}_{\text{ini}}\\ u\\ y\end{pmatrix} + \begin{pmatrix}0\\ \sigma_y\\ 0\\ 0\end{pmatrix}, \\
& u_k \in \mathcal{U},\ \forall k \in \{0,\ldots,T_{\text{f}}-1\}, \\
& y_k \in \mathcal{Y},\ \forall k \in \{0,\ldots,T_{\text{f}}-1\}.
\end{aligned}$$





1-norm promotes sparsity $\iff$ sparse selection of **motion primitives**

Nonlinear Systems

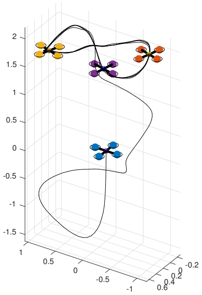
Idea: Can lift nonlinear system to large/infinite-dimensional bi-/linear system (e.g., Carleman, Koopman, Volterra)

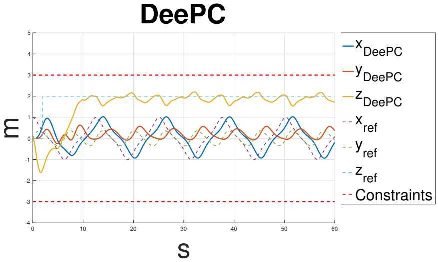
Build **larger Hankel matrix** and let **1-norm regularization** pick features

$$\underbrace{\begin{pmatrix} \hat{u}_1 & \hat{u}_2 & \cdots \\ \vdots & \vdots & \ddots \\ \hat{u}_{T_{\text{ini}}+T_{\text{f}}} & \hat{u}_{T_{\text{ini}}+T_{\text{f}}+1} & \cdots \\ \hat{y}_1 & \hat{y}_2 & \cdots \\ \vdots & \vdots & \ddots \\ \hat{y}_{T_{\text{ini}}+T_{\text{f}}} & \hat{y}_{T_{\text{ini}}+T_{\text{f}}+1} & \cdots \end{pmatrix}}_{\text{more data = more columns}}$$

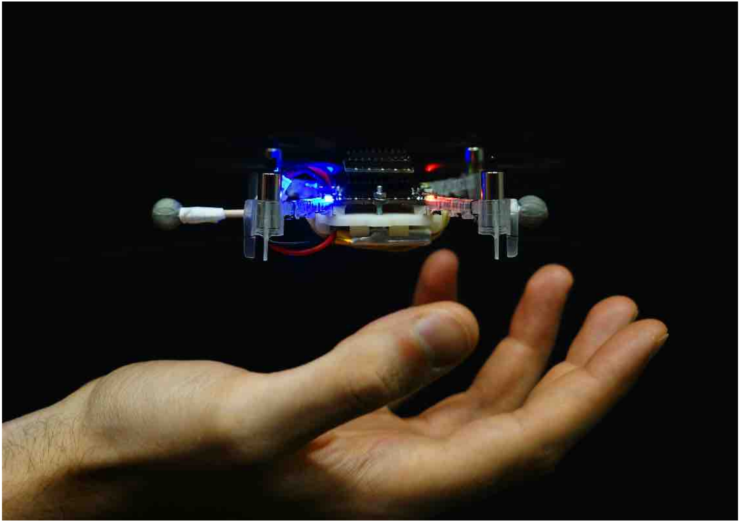
Nonlinear Case Study

Setup: nonlinear stochastic quadcopter model with full state info
DeePC: Nominal DeePC + σ_y slack + 1-norm regularization for g
+ more columns





Real-world Experiment



Why does it work so well?

Why does it work so well?

Time for some theory!

Distributionally Robust DeePC

DeePC + σ_y slack:

$$\underset{g,u,y}{\text{minimize}} \quad \sum_{k=0}^{T_\mathrm{f}-1} f(u_k, y_k) + \lambda_y \|\sigma_y\|_p$$

$$\text{subject to} \quad \mathscr{H}_{T_\mathrm{ini}+T_\mathrm{f}} \begin{pmatrix} \hat{u} \\ \hat{y} \end{pmatrix} g = \begin{pmatrix} \hat{u}_\mathrm{ini} \\ \hat{y}_\mathrm{ini} \\ u \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ \sigma_y \\ 0 \\ 0 \end{pmatrix},$$

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

Abstracted DeePC:

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

with $\hat{\xi} = \mathscr{H}_{T_\mathrm{ini}+T_\mathrm{f}}(\hat{y})$ and

$$G = \left\{ g \;\middle|\; \mathscr{H}_{T_\mathrm{ini}+T_\mathrm{f}}(\hat{u}) g = \begin{pmatrix} \hat{u}_\mathrm{ini} \\ u \end{pmatrix}, u \in \mathcal{U}^{T_\mathrm{f}} \right\}.$$

- ξ is a **random variable** distributed according to unknown distribution $\mathbb{P}$
- $\hat{\xi}$ is a particular **measurement** of random variable ξ

We use $\hat{\cdot}$ to denote measured (thus possibly noisy) data.

Abstracted DeePC

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

where $\widehat{\mathbb{P}} = \delta_{\hat{\xi}}$ is the **empirical distribution** of ξ (approximation for true data generating distribution $\mathbb{P}$).

Abstracted DeePC

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

where $\widehat{\mathbb{P}} = \delta_{\hat{\xi}}$ is the **empirical distribution** of ξ (approximation for true data generating distribution $\mathbb{P}$).

Solution has poor **out-of-sample performance** $\mathbb{E}_{\mathbb{P}}[c(\xi, g^\star)]$ where $\mathbb{P}$ is true distribution of ξ and $g^\star$ solution to above.

Abstracted DeePC

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

where $\widehat{\mathbb{P}} = \delta_{\hat{\xi}}$ is the **empirical distribution** of ξ (approximation for true data generating distribution $\mathbb{P}$).

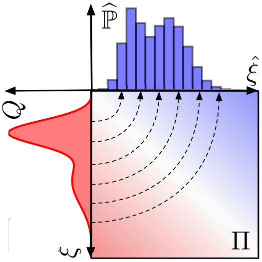
Solution has poor **out-of-sample performance** $\mathbb{E}_{\mathbb{P}}[c(\xi, g^\star)]$ where $\mathbb{P}$ is true distribution of ξ and $g^\star$ solution to above.

Distributionally Robust DeePC

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

where the ambiguity set is the **Wasserstein ball**

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



Theorem

Under minor technical conditions

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

Hence,

$$p\text{-norm robustness} \iff q\text{-norm regularization}$$

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

Note that the Wasserstein ball contains **more than just LTI systems with additive noise.**

Proof uses methods from [Mohajerin Esfahani and Kuhn, 2018].

Further Improvements?

1. How to leverage more data?
2. How to include output constraints?
3. Am I stuck with the Hankel matrix structure?

Leveraging more data

- Collect **many Hankel matrices** $\hat{\xi}^{(i)} = \mathscr{H}^{(i)}_{T_{\text{ini}}+T_{\text{f}}}(\hat{y}^{(i)})$, $i \in \{1, \ldots, N\}$.
- Result is **“better” empirical distribution** $\widehat{\mathbb{P}} = \frac{1}{N}\sum_{i=1}^{N} \delta_{\hat{\xi}^{(i)}}$.
- Use **measure concentration** to decrease size of $B_\epsilon(\widehat{\mathbb{P}})$.

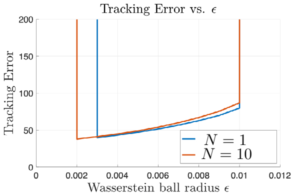
Leveraging more data

- Collect **many Hankel matrices** $\hat{\xi}^{(i)} = \mathscr{H}^{(i)}_{T_{\text{ini}}+T_{\text{f}}}(\hat{y}^{(i)})$, $i \in \{1, \ldots, N\}$.
- Result is **“better” empirical distribution** $\widehat{\mathbb{P}} = \frac{1}{N}\sum_{i=1}^{N} \delta_{\hat{\xi}^{(i)}}$.
- Use **measure concentration** to decrease size of $B_\epsilon(\widehat{\mathbb{P}})$.

Theorem

Let $\epsilon \sim \frac{1}{N}^{1/\dim \xi}$. Then with high probability

$$\underbrace{\mathbb{E}_{\mathbb{P}}[c(\xi, g)]}_{\text{true out-of-sample performance}} \leq \underbrace{\frac{1}{N}\sum_{i=1}^{N} c(\hat{\xi}^{(i)}, g)}_{\text{sample average cost}} + \epsilon \text{Lip}(c)\|g\|_*$$



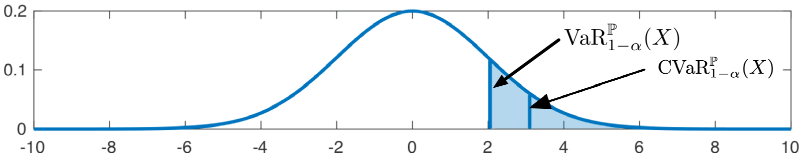
Distributionally Robust Chance Constraints

- Want future trajectories to be **constrained** in $\mathcal{Y} = \{y \mid h(y) \leq 0\}$.
- Relax to **chance-constraint** $\mathbb{P}(h(y) \leq 0) \geq 1 - \alpha \iff \text{VaR}^{\mathbb{P}}_{1-\alpha}(h(y)) \leq 0$.
- **Convex relaxation** $\text{CVaR}^{\mathbb{P}}_{1-\alpha}(h(y)) \leq 0$.

Distributionally Robust CVaR Constraint

$$\sup_{Q \in B_\epsilon(\widehat{\mathbb{P}})} \text{CVaR}^{Q}_{1-\alpha}(h(y))$$

$\iff$ sample average constraint + regularization + tightening



New Data Structure – Page matrix

Hankel Matrix

$$\mathscr{H}_L(u,y) = \begin{pmatrix} \begin{pmatrix} u_1 \\ y_1 \end{pmatrix} & \begin{pmatrix} u_2 \\ y_2 \end{pmatrix} & \cdots \\ \begin{pmatrix} u_2 \\ y_2 \end{pmatrix} & \begin{pmatrix} u_3 \\ y_3 \end{pmatrix} & \cdots \\ \vdots & \vdots & \ddots \\ \begin{pmatrix} u_L \\ y_L \end{pmatrix} & \begin{pmatrix} u_{L+1} \\ y_{L+1} \end{pmatrix} & \cdots \end{pmatrix}$$

Page Matrix

$$\mathscr{P}_L(u,y) = \begin{pmatrix} \begin{pmatrix} u_1 \\ y_1 \end{pmatrix} & \begin{pmatrix} u_{L+1} \\ y_{L+1} \end{pmatrix} & \cdots \\ \begin{pmatrix} u_2 \\ y_2 \end{pmatrix} & \begin{pmatrix} u_{L+2} \\ y_{L+2} \end{pmatrix} & \cdots \\ \vdots & \vdots & \ddots \\ \begin{pmatrix} u_L \\ y_L \end{pmatrix} & \begin{pmatrix} u_{2L} \\ y_{2L} \end{pmatrix} & \cdots \end{pmatrix}$$

New Data Structure – Page matrix

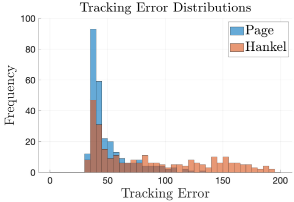
Hankel Matrix

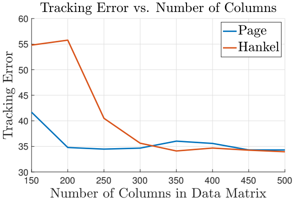
$$\mathscr{H}_L(u,y) = \begin{pmatrix} \begin{pmatrix} u_1 \\ y_1 \end{pmatrix} & \begin{pmatrix} u_2 \\ y_2 \end{pmatrix} & \cdots \\ \begin{pmatrix} u_2 \\ y_2 \end{pmatrix} & \begin{pmatrix} u_3 \\ y_3 \end{pmatrix} & \cdots \\ \vdots & \vdots & \ddots \\ \begin{pmatrix} u_L \\ y_L \end{pmatrix} & \begin{pmatrix} u_{L+1} \\ y_{L+1} \end{pmatrix} & \cdots \end{pmatrix}$$

Page Matrix

$$\mathscr{P}_L(u,y) = \begin{pmatrix} \begin{pmatrix} u_1 \\ y_1 \end{pmatrix} & \begin{pmatrix} u_{L+1} \\ y_{L+1} \end{pmatrix} & \cdots \\ \begin{pmatrix} u_2 \\ y_2 \end{pmatrix} & \begin{pmatrix} u_{L+2} \\ y_{L+2} \end{pmatrix} & \cdots \\ \vdots & \vdots & \ddots \\ \begin{pmatrix} u_L \\ y_L \end{pmatrix} & \begin{pmatrix} u_{2L} \\ y_{2L} \end{pmatrix} & \cdots \end{pmatrix}$$

Distrbutionally robust analysis is **tight** for Page matrix!



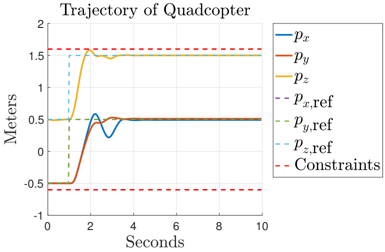


Putting it all together

Setup: Nonlinear noisy quadcopter model

Solution

DeePC
+ distributionally robust objective
+ CVaR constraints
+ leverage more data
+ Page matrix



Summary

Recap:

- Matrix of time-series data is a predictive model
- DeePC equivalent to MPC for deterministic LTI systems
- **regularizations** to DeePC provide **distributional robustness** to extend beyond deterministic LTI setting

Future work:

- Fundamental Lemma for **nonlinear/stochastic** systems
- **Online and adaptive** extensions to DeePC
- Connections of DeePC to **ID for control**

ETH zürich

Thanks!

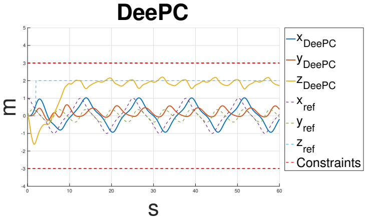
AUTOMATIC CONTROL LABORATORY

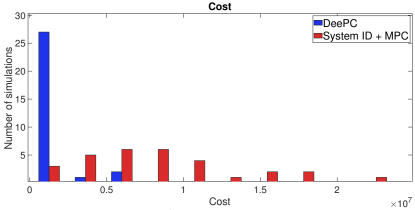
Jeremy Coulson
jcoulson@ethz.ch

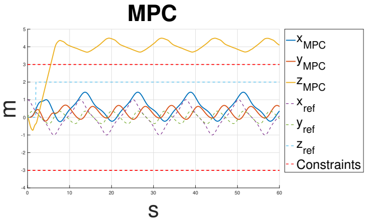
Appendix

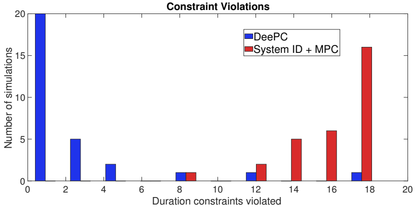
DeePC vs MPC

DeePC: ℓ^1-regularization for g and σ_y slack
MPC: system ID (prediction error method) + MPC









Direct better than indirect?

$\rightarrow$ still exploring