Data Enabled Predictive Control

Dr. Keith Moffat

We can use data to find the valid I/O sequences

All and **only** the valid I/O sequences are of the form

$$\begin{bmatrix} \mathbf{u}_L \\ \mathbf{y}_L \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{u}_L^{(1)} & \mathbf{u}_L^{(2)} & \mathbf{u}_L^{(3)} & \cdots & \mathbf{u}_L^{(L+n)} \\ \mathbf{y}_L^{(1)} & \mathbf{y}_L^{(2)} & \mathbf{y}_L^{(3)} & \cdots & \mathbf{y}_L^{(L+n)} \end{bmatrix}}_{H} g$$

with $g \in \mathbb{R}^{L+n}$.

Hankel matrices

A very efficient use of your data points:

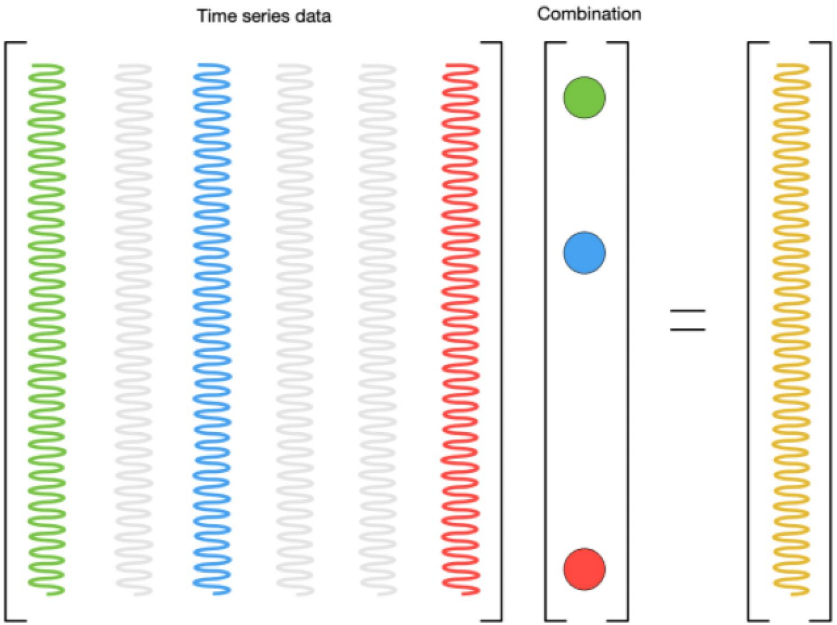
$$\dots \quad u_0 \quad u_1 \quad u_2 \quad u_3 \quad u_4 \quad u_5 \quad u_6 \quad u_7 \quad \dots$$

rearranged as

$$\mathcal{H}_L(\mathbf{u}) = \begin{bmatrix} u_0 & u_1 & u_2 & \dots & u_{L+n-1} \\ u_1 & u_2 & u_3 & & \\ u_2 & u_3 & u_4 & & \\ \dots & & & & \\ u_L & & & & u_{2L+n-1} \end{bmatrix}$$

All system trajectories of length L are generated as

$$\begin{bmatrix} \mathcal{H}_L(\mathbf{u}) \\ \mathcal{H}_L(\mathbf{y}) \end{bmatrix} g$$



For DeePC, we divide each Hankel matrix (input and output) into

- one Hankel matrix for the past measurements
- one Hankel matrix for the future measurements

$$\begin{bmatrix} \mathcal{H}_L(\mathbf{u}_{\text{data}}) \\ \mathcal{H}_L(\mathbf{y}_{\text{data}}) \end{bmatrix} g = \begin{bmatrix} \mathbf{u}_{\text{past}} \\ \mathbf{u} \\ \mathbf{y}_{\text{past}} \\ \mathbf{y} \end{bmatrix}$$

MPC

$$\min_{\mathbf{u},\mathbf{x},\mathbf{y}} \sum_{k=0}^{K-1} y_k^2 + r u_k^2$$

$$\begin{aligned} \text{subject to}\quad & x_{k+1} = A x_k + B u_k \\ & y_k = C x_k \\ & x_0 = x \\ & u_k \in \mathcal{U}_k \ \forall k \\ & y_k \in \mathcal{Y}_k \ \forall k \end{aligned}$$

DeePC

$$\min_{\mathbf{u},\mathbf{x},\mathbf{y},g} \sum_{k=0}^{K-1} y_k^2 + r u_k^2$$

$$\begin{aligned} \text{subject to}\quad & \begin{bmatrix} \mathcal{H}_L(\mathbf{u}_{\text{data}}) \\ \mathcal{H}_L(\mathbf{y}_{\text{data}}) \end{bmatrix} g = \begin{bmatrix} \mathbf{u}_{\text{past}} \\ \mathbf{u} \\ \mathbf{y}_{\text{past}} \\ \mathbf{y} \end{bmatrix} \\ & u_k \in \mathcal{U}_k \ \forall k \\ & y_k \in \mathcal{Y}_k \ \forall k \end{aligned}$$

SISO DeePC

$$\min_{\mathbf{u},\mathbf{x},\mathbf{y},g} \sum_{k=0}^{K-1} y_k^2 + r u_k^2$$

$$\text{subject to} \quad \begin{bmatrix} \mathcal{H}_L(\mathbf{u}_{\text{data}}) \\ \mathcal{H}_L(\mathbf{y}_{\text{data}}) \end{bmatrix} g = \begin{bmatrix} \mathbf{u}_{\text{past}} \\ \mathbf{u} \\ \mathbf{y}_{\text{past}} \\ \mathbf{y} \end{bmatrix}$$

$$u_k \in \mathcal{U}_k \ \forall k$$

$$y_k \in \mathcal{Y}_k \ \forall k$$

MIMO DeePC

$$\min_{\mathbf{u},\mathbf{x},\mathbf{y},g} \sum_{k=0}^{K-1} y_k^2 + r u_k^2$$

$$\text{subject to} \quad \begin{bmatrix} \mathcal{H}_L(\mathbf{u}_{\text{data}}) \\ \mathcal{H}_L(\mathbf{y}_{\text{data}}) \end{bmatrix} g = \begin{bmatrix} \mathbf{u}_{\text{past}} \\ \mathbf{u} \\ \mathbf{y}_{\text{past}} \\ \mathbf{y} \end{bmatrix}$$

$$u_k \in \mathcal{U}_k \ \forall k$$

$$y_k \in \mathcal{Y}_k \ \forall k$$

What is the difference between the two formulations?

MIMO DeePC Hankel Matrix

Block Hankel matrix:

$w_d^2(1)$	$w_d^2(2)$	$w_d^2(3)$	$w_d^2(4)$	$w_d^2(5)$	$w_d^2(6)$
$w_d^2(2)$	$w_d^2(3)$	$w_d^2(4)$	$w_d^2(5)$	$w_d^2(6)$	$w_d^2(7)$
$w_d^2(3)$	$w_d^2(4)$	$w_d^2(5)$	$w_d^2(6)$	$w_d^2(7)$	$w_d^2(8)$
$w_d^2(4)$	$w_d^2(5)$	$w_d^2(6)$	$w_d^2(7)$	$w_d^2(8)$	$w_d^2(9)$
$w_d^2(5)$	$w_d^2(6)$	$w_d^2(7)$	$w_d^2(8)$	$w_d^2(9)$	$w_d^2(10)$

Markovsky, Ivan, Linbin Huang, and Florian Dörfler. "Data-driven control based on the behavioral approach: From theory to applications in power systems." *IEEE Control Systems* (2023).

MIMO DeePC

$$\min_{\mathbf{u},\mathbf{x},\mathbf{y},g} \sum_{k=0}^{K-1} y_k^2 + r u_k^2$$

$$\text{subject to} \quad \begin{bmatrix} \mathcal{H}_L(\mathbf{u}_{\text{data}}) \\ \mathcal{H}_L(\mathbf{y}_{\text{data}}) \end{bmatrix} g = \begin{bmatrix} \mathbf{u}_{\text{past}} \\ \mathbf{u} \\ \mathbf{y}_{\text{past}} \\ \mathbf{y} \end{bmatrix}$$

$$u_k \in \mathcal{U}_k \ \forall k$$

$$y_k \in \mathcal{Y}_k \ \forall k$$

What is the maximum rank for a Hankel matrix for a MIMO system with noiseless data?

MIMO DeePC Hankel Matrix

- the rank of H stops increasing after $mL + n$

$$\underbrace{\begin{bmatrix} \mathbf{u}_L^{(1)} & \mathbf{u}_L^{(2)} & \mathbf{u}_L^{(3)} & \cdots \\ \mathbf{y}_L^{(1)} & \mathbf{y}_L^{(2)} & \mathbf{y}_L^{(3)} & \cdots \end{bmatrix}}_{H} g$$

This is true only in the **noiseless case**.

The effect of noise on the Hankel matrix

- the rank of H stops increasing after $mL + n$

$$\underbrace{\begin{bmatrix} \mathbf{u}_L^{(1)} & \mathbf{u}_L^{(2)} & \mathbf{u}_L^{(3)} & \cdots \\ \mathbf{y}_L^{(1)} & \mathbf{y}_L^{(2)} & \mathbf{y}_L^{(3)} & \cdots \end{bmatrix}}_{H} g$$

This is true only in the **noiseless case**.

In the presence of **noise in the data**, the rank of H will increase past $mL + n$.

MIMO Systems: it is helpful to know the “lag” of the system

Lag of a system

The **lag** of a system is the smallest ℓ such that

$$\mathcal{O}_\ell = \begin{bmatrix} C \\ CA \\ \cdots \\ CA^{\ell-1} \end{bmatrix}$$

has rank n.

- A past trajectory of length $T_{\text{ini}} \geq \ell$ is needed as “initial conditions”.

For observable SISO systems, $\ell = n$.

Lecture slide correction

- A past trajectory of length $T_{\text{ini}} \geq \ell$ is needed as “initial conditions”.
- For MIMO systems
  - input $u \in \mathbb{R}^m$
  - output $y \in \mathbb{R}^p$
  - state $x \in \mathbb{R}^n$

  an input/output sequences of at length $T \geq (m+1)(T_{\text{ini}} + K + n) - 1$ is needed
- For SISO systems, $\ell = n$.

What should the lower bound be on T?

The (misled) optimism of DeePC with noisy data

If there is noise in the data, the DeePC optimization can (and most likely) will choose trajectories that are not feasible for the true system.

Often, these trajectories use large g vector values. Why?

The (misled) optimism of DeePC with noisy data

Consider the (pathological) case in which one of the outputs is not controllable.

$$y_k = Cx_k = \begin{bmatrix} \star & \star & \star \\ 0 & 0 & 0 \end{bmatrix} x_k$$

All output sequences should be of the form

$$y_k = \begin{bmatrix} \star \\ 0 \end{bmatrix}$$

However, if the data contains noise, some columns in the data matrix may present a non-zero second coordinate in the output

$$\underbrace{\begin{bmatrix} \mathbf{u}_L^{(1)} & \mathbf{u}_L^{(2)} & \mathbf{u}_L^{(3)} & \cdots \\ \mathbf{y}_L^{(1)} & \mathbf{y}_L^{(2)} & \mathbf{y}_L^{(3)} + \epsilon & \cdots \end{bmatrix}}_{H} g$$

The (misled) optimism of DeePC with noisy data

If there is noise in the data, the DeePC optimization can (and most likely) will choose trajectories that are not feasible for the true system.

Often, these “misled optimistic” trajectories use large g vector values. Why?

Hand-wavy intuition:
The over-optimism comes from the g vector selecting trajectories that are only possible after adding noise. The noise contributes smaller singular values than the true data. To activate the trajectories that are created by noise and correspond to small singular values, the g vector must be large.

How can you avoid selecting the “misled optimistic” trajectories?

Addressing the (misled) optimism of DeePC with noisy data

$$\min_{\mathbf{u},\mathbf{y},g,\hat{\mathbf{u}}_{\text{data}},\hat{\mathbf{y}}_{\text{data}}} \quad \sum_{k=0}^{K-1} \|y_k\|_Q^2 + \|u_k\|_R^2$$

$$\text{subject to} \quad (\hat{\mathbf{u}}_{\text{data}}, \hat{\mathbf{y}}_{\text{data}}) = \text{argmin} \, \|\hat{\mathbf{u}}_{\text{data}} - \mathbf{u}_{\text{data}}\|^2 + \|\hat{\mathbf{y}}_{\text{data}} - \mathbf{y}_{\text{data}}\|^2$$

$$\text{subject to} \quad \text{rank}\left(\begin{bmatrix} \mathcal{H}_L(\hat{\mathbf{u}}_{\text{data}}) \\ \mathcal{H}_L(\hat{\mathbf{y}}_{\text{data}}) \end{bmatrix}\right) = L + n$$

$$\begin{bmatrix} \mathcal{H}_L(\hat{\mathbf{u}}_{\text{data}}) \\ \mathcal{H}_L(\hat{\mathbf{y}}_{\text{data}}) \end{bmatrix} g = \begin{bmatrix} \mathbf{u}_{\text{past}} \\ \mathbf{u} \\ \mathbf{y}_{\text{past}} \\ \mathbf{y} \end{bmatrix}$$

$$u_k \in \mathcal{U}_k \ \forall k$$

$$y_k \in \mathcal{Y}_k \ \forall k$$

Addressing the (misled) optimism of DeePC with noisy data

Step 2: Convex relaxation → drop the rank constraint

$$
\begin{aligned}
\min_{\mathbf{u},\mathbf{y},g,\hat{\mathbf{u}}_{\text{data}},\hat{\mathbf{y}}_{\text{data}}} \quad & \sum_{k=0}^{K-1} \|y_k\|_Q^2 + \|u_k\|_R^2 \\
\text{subject to} \quad & (\hat{\mathbf{u}}_{\text{data}}, \hat{\mathbf{y}}_{\text{data}}) = \operatorname{argmin} \|\hat{\mathbf{u}}_{\text{data}} - \mathbf{u}_{\text{data}}\|^2 + \|\hat{\mathbf{y}}_{\text{data}} - \mathbf{y}_{\text{data}}\|^2 \\
& \begin{bmatrix} \mathcal{H}_L(\hat{\mathbf{u}}_{\text{data}}) \\ \mathcal{H}_L(\hat{\mathbf{y}}_{\text{data}}) \end{bmatrix} g = \begin{bmatrix} \mathbf{u}_{\text{past}} \\ \mathbf{u} \\ \mathbf{y}_{\text{past}} \\ \mathbf{y} \end{bmatrix} \\
& \|g\|_0 \leq L + n \\
& u_k \in \mathcal{U}_k \ \forall k \\
& y_k \in \mathcal{Y}_k \ \forall k
\end{aligned}
$$

Is the 0-norm constraint convex?

Addressing the (misled) optimism of DeePC with noisy data

Step 3: from sparsity of g to a bound on $\|g\|_1$

$$\min_{\mathbf{u},\mathbf{y},g,\hat{\mathbf{u}}_{\text{data}},\hat{\mathbf{y}}_{\text{data}}} \quad \sum_{k=0}^{K-1} \|y_k\|_Q^2 + \|u_k\|_R^2$$

$$\text{subject to} \quad \begin{bmatrix} \mathcal{H}_L(\mathbf{u}_{\text{data}}) \\ \mathcal{H}_L(\mathbf{y}_{\text{data}}) \end{bmatrix} g = \begin{bmatrix} \mathbf{u}_{\text{past}} \\ \mathbf{u} \\ \mathbf{y}_{\text{past}} \\ \mathbf{y} \end{bmatrix}$$

$$\|g\|_1 \leq \alpha$$

$$u_k \in \mathcal{U}_k \ \forall k$$

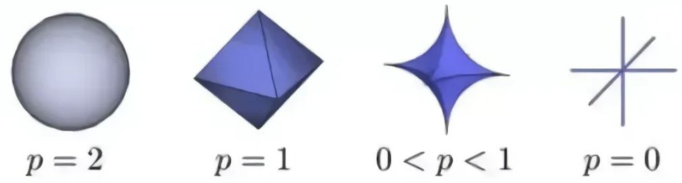
$$y_k \in \mathcal{Y}_k \ \forall k$$

Is the 1-norm constraint convex?

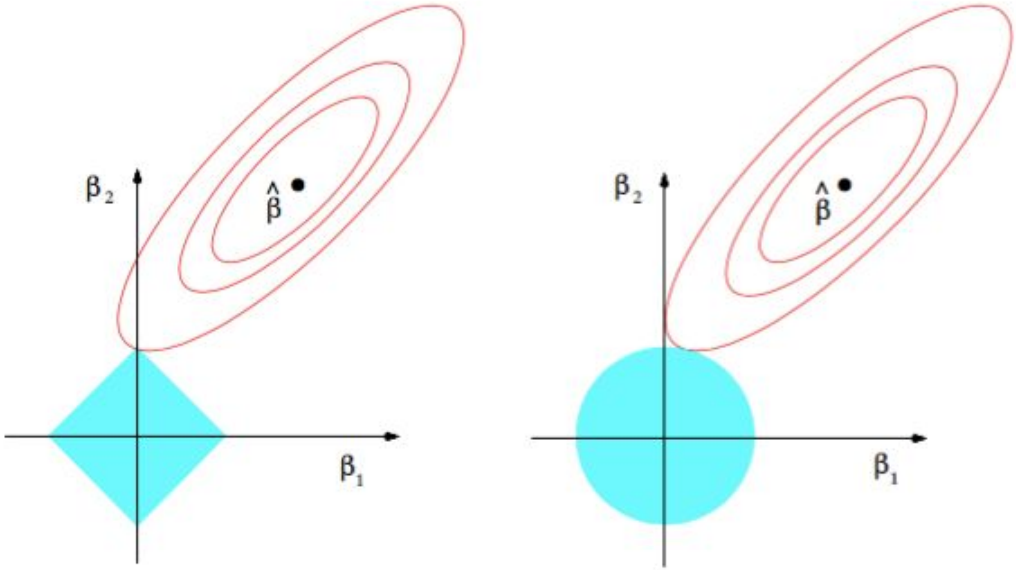
Will the 1-norm still induce sparsity on g?

The unit ball for some different norms (and pseudo-norms)

$$\|\mathbf{x}\|_p := \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}$$



The 1-norm and sparsity



Fif. 3.11 from *Elements of Statistical Learning* by Hastie, Tibshirani, and Friedman

Addressing the (misled) optimism of DeePC with noisy data

Step 3: from sparsity of g to a bound on $\|g\|_1$

$$\min_{\mathbf{u},\mathbf{y},g,\hat{\mathbf{u}}_{\text{data}},\hat{\mathbf{y}}_{\text{data}}} \quad \sum_{k=0}^{K-1} \|y_k\|_Q^2 + \|u_k\|_R^2$$

$$\text{subject to} \quad \begin{bmatrix} \mathcal{H}_L(\mathbf{u}_{\text{data}}) \\ \mathcal{H}_L(\mathbf{y}_{\text{data}}) \end{bmatrix} g = \begin{bmatrix} \mathbf{u}_{\text{past}} \\ \mathbf{u} \\ \mathbf{y}_{\text{past}} \\ \mathbf{y} \end{bmatrix}$$

$$\|g\|_1 \leq \alpha$$

$$u_k \in \mathcal{U}_k \ \forall k$$

$$y_k \in \mathcal{Y}_k \ \forall k$$

Is the 1-norm constraint convex? Yes.

Will the 1-norm still induce sparsity on g? Yes.

Do we know α?

Addressing the (misled) optimism of DeePC with noisy data

Step 4: Replace the bound on $\|g\|_1$ with a regularization term

$$\min_{\mathbf{u},\mathbf{x},g} \quad \sum_{k=0}^{K-1} y_k^2 + r u_k^2 + \lambda_g \|g\|_1$$

$$\text{subject to} \quad \begin{bmatrix} \mathcal{H}_L(\mathbf{u}_{\text{data}}) \\ \mathcal{H}_L(\mathbf{y}_{\text{data}}) \end{bmatrix} g = \begin{bmatrix} \mathbf{u}_{\text{past}} \\ \mathbf{u} \\ \mathbf{y}_{\text{past}} \\ \mathbf{y} \end{bmatrix}$$

$$u_k \in \mathcal{U}_k \ \forall k$$

$$y_k \in \mathcal{Y}_k \ \forall k$$

This is regularized DeePC

What is the effect of large λ_g? Small λ_g ?

Choosing λ_g

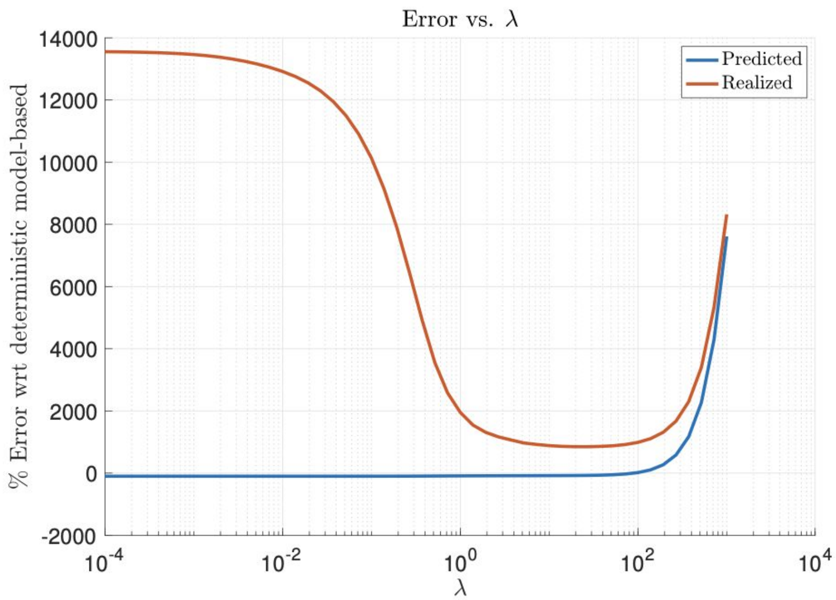


Figure 4.1: Predicted and realized errors (relative to the ground-truth optimal performance and averaged over 100 data sets) with 1-norm regularizer $\lambda\|g\|_1$.

Dörfler, Florian, Jeremy Coulson, and Ivan Markovsky. "Bridging direct and indirect data-driven control formulations via regularizations and relaxations." *IEEE Transactions on Automatic Control* 68.2 (2022): 883-897.

Jeremy Coulson's Ph.D. Thesis, "Data-enabled Predictive Control, Theory and Practice"
https://www.research-collection.ethz.ch/bitstream/handle/20.500.11850/596662/7/Thesis.pdf

An (often) necessary slack variable for DeePC with noisy data

$$\min_{\mathbf{u},\mathbf{x},g} \quad \sum_{k=0}^{K-1} y_k^2 + r u_k^2 + \lambda_g \|g\|_1 + \lambda_\sigma \|\sigma\|_1$$

$$\text{subject to} \quad \begin{bmatrix} \mathcal{H}_L(\mathbf{u}_{\text{data}}) \\ \mathcal{H}_L(\mathbf{y}_{\text{data}}) \end{bmatrix} g = \begin{bmatrix} \mathbf{u}_{\text{past}} \\ \mathbf{u} \\ \mathbf{y}_{\text{past}} + \sigma \\ \mathbf{y} \end{bmatrix}$$

$$u_k \in \mathcal{U}_k \ \forall k$$

$$y_k \in \mathcal{Y}_k \ \forall k$$

Why is σ often necessary when there is noise in the data?