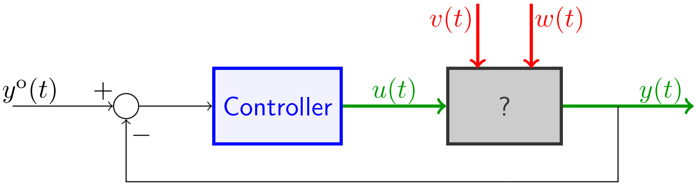
Data-driven predictive control of stochastic systems

S. Formentin (with V. Breschi and A. Chiuso)

Virtual study day of the French Identification group

January 19th, 2023

Direct data-driven (DD) control

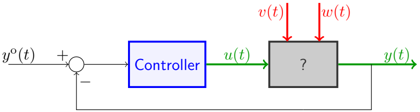


Key question

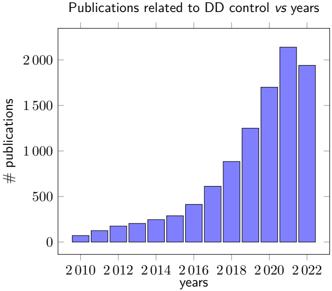
How to design the controller **directly** from a set of I/O data?

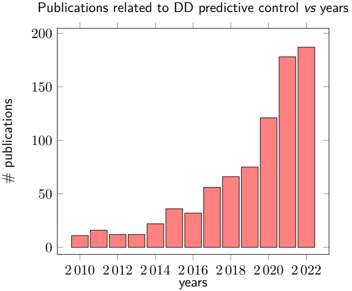
The relevance of DD control

Modeling is known to be the most time-consuming step of a control project ($\sim 75\%$ of total time)



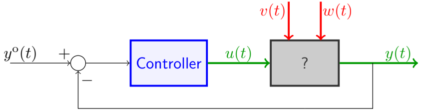
Number of publications on DD control over the years



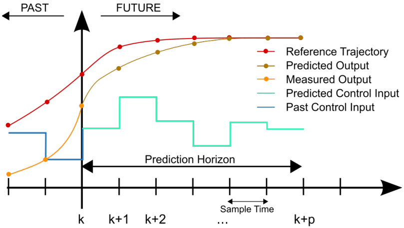


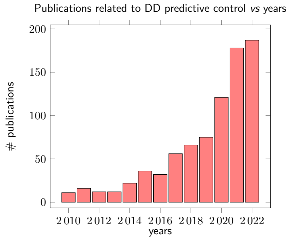
Outline

Our focus

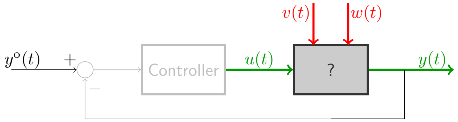


Shift from deterministic to stochastic DD predictive control





Our main assumptions



- **Unknown** system: Linear Time Invariant (LTI)

$$\begin{cases} x(t+1) = Ax(t) + Bu(t) + w(t) \\ y(t) = Cx(t) + Du(t) + v(t) \end{cases}$$

- **Process/measurement** noise: white, zero mean

$$w \sim wn(0, \Gamma_w^2), \quad v \sim wn(0, \Gamma_v^2), \quad cov(w, v) = \Gamma_{wv}$$

- **Dataset**: $\mathcal{D}_{N^d} = \{u^d(t), y^d(t)\}_{t=1}^{N^d}$
- **Input** $u^d(t)$: persistently exciting of sufficient order
- **Set point**: $y^{o}(t) = y^{o}, \quad \forall t \geq 0$

Back to basics: the model-based problem

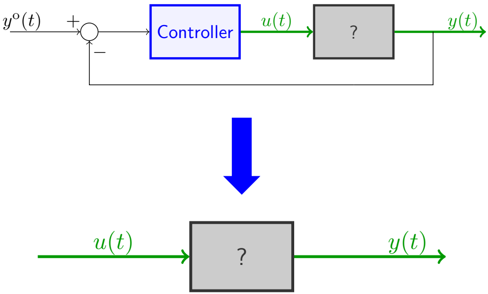
$$\begin{aligned}
\underset{\substack{\bar{u}_{[0,L-1]}(t)\\ \bar{y}_{[0,L-1]}(t)}}{\text{minimize}} \quad & \sum_{k=0}^{L-1} \underbrace{\mathbb{E}[\|\bar{y}_k(t) - y^{\mathrm{o}}\|_Q^2] + \|\bar{u}_k(t)\|_R^2}_{\ell(\bar{u}_k(t), \bar{y}_k(t))} \\
\text{s.t.} \quad & \bar{x}_{k+1}(t) = \mathbf{A}\bar{x}_k(t) + \mathbf{B}\bar{u}_k(t) + \mathbf{w_k}, \quad k \in [0, L-1) \\
& \bar{y}_k(t) = \mathbf{C}\bar{x}_k(t) + \mathbf{D}\bar{u}_k(t) + \mathbf{v_k}, \quad k \in [0, L-1) \\
& \mathbf{x_0(t) = x(t)} \\
& \bar{u}_k(t) \in \mathcal{U}, \;\; \mathbb{E}[\bar{y}_k(t)] \in \mathcal{Y}, \quad k \in [0, L-1)
\end{aligned}$$

How to use data instead of model parameters ?

How to cope with noise ?

How to cope with unknown initial conditions ?

Shifting from a model-based to a data-driven formulation



Goals

#1: Describe the **predictive model** of the system in a purely **input/output** framework

#2: Express the **initial conditions** with **inputs & outputs** only

Towards an input/output predictive model

Leveraging behavioral systems theory, the **system dynamics** can be expressed as a combination of input/output trajectories
(as in *e.g.*, Coulson et al., 2019; Berberich et al., 2021)

If a state-space model generates $\mathcal{D}_{N^d} = \{u^d(t), y^d(t)\}_{t=1}^{N^d}$

$$\begin{cases} \bar{x}_k(t+1) = A\bar{x}_k(t) + B\bar{u}_k(t), \\ \bar{y}_k(t) = C\bar{x}_k(t) + D\bar{u}_k(t) \end{cases}$$

$$\Downarrow$$

the data satisfy the following equation for a certain $\alpha(\mathbf{t})$

$$\begin{bmatrix} \bar{u}_{[0,L-1]}(t) \\ \bar{y}_{[0,L-1]}(t) \end{bmatrix} = \underbrace{\begin{bmatrix} H_L(u^d) \\ H_L(y^d) \end{bmatrix}}_{\text{Hankel matrices build from data}} \alpha(\mathbf{t})$$

Initial conditions as functions of inputs/outputs

$$x(t) = A^{\rho} x(t-\rho) + \mathcal{C} \begin{bmatrix} u_{[t-\rho,t-1]} \\ y_{[t-\rho,t-1]} \end{bmatrix}$$

For a stable system, we can go from an initial state...

$$\bar{x}_0(t) = x(t)$$

...to a past input/output trajectory

$$\begin{bmatrix} \bar{u}_{[-\rho,-1]}(t) \\ \bar{y}_{[-\rho,-1]}(t) \end{bmatrix} = \begin{bmatrix} u_{[t-\rho,t-1]} \\ y_{[t-\rho,t-1]} \end{bmatrix}$$

$$\rho \geq n$$

(see, *e.g.*, Willems et al., 2005, Moonen et al., 1989)

Initial conditions as functions of inputs/outputs

$$x(t) = A^{\rho}x(t-\rho) + \mathcal{C}\begin{bmatrix} u_{[t-\rho,t-1]} \\ y_{[t-\rho,t-1]} \end{bmatrix}$$

For a stable system, we can go from an initial state...

$$\bar{x}_0(t) = x(t)$$

$$\Downarrow$$

...to a past input/output trajectory

$$\begin{bmatrix} \bar{u}_{[-\rho,-1]}(t) \\ \bar{y}_{[-\rho,-1]}(t) \end{bmatrix} = \begin{bmatrix} u_{[t-\rho,t-1]} \\ y_{[t-\rho,t-1]} \end{bmatrix}$$

How to tune ρ if the **actual order** n of the system is unknown?

Deterministic data-driven PC (1)

Substituting the **previous relations** into the predictive control problem...

$$\begin{aligned}
\underset{\substack{\bar{u}_{[0,L-1]}(t),\alpha(t)\\ \bar{y}_{[0,L-1]}(t)}}{\text{minimize}} \quad & \sum_{k=0}^{L-1} \underbrace{\|\bar{y}_k(t) - y^{\mathrm{o}}\|_Q^2 + \|\bar{u}_k(t)\|_R^2}_{\ell(\bar{u}_k(t),\bar{y}_k(t))} \\
\text{s.t.} \quad & \begin{bmatrix} \bar{u}_{[-\rho,L-1]}(t) \\ \bar{y}_{[-\rho,L-1]}(t) \end{bmatrix} = \begin{bmatrix} H_{L+\rho}(u^d) \\ H_{L+\rho}(y^d) \end{bmatrix} \alpha(t) \\
& \begin{bmatrix} \bar{u}_{[-\rho,-1]}(t) \\ \bar{y}_{[-\rho,-1]}(t) \end{bmatrix} = \begin{bmatrix} u_{[t-\rho,t-1]} \\ y_{[t-\rho,t-1]} \end{bmatrix} \\
& \bar{u}_k(t) \in \mathcal{U}, \quad \bar{y}_k(t) \in \mathcal{Y}, \quad k \in [0, L-1)
\end{aligned}$$

Features of the problem

Equivalent to the model-based problem, provided ρ is **big enough**

(Coulson et al., 2019)

Deterministic data-driven PC (2)

Substituting the **previous relations** into the predictive control problem...

$$\underset{\substack{\bar{u}_{[0,L-1]}(t),\alpha(t)\\ \bar{y}_{[0,L-1]}(t)}}{\text{minimize}} \quad \sum_{k=0}^{L-1} \underbrace{\|\bar{y}_k(t)-y^{\mathrm{o}}\|_Q^2+\|\bar{u}_k(t)\|_R^2}_{\ell(\bar{u}_k(t),\bar{y}_k(t))}$$

$$\text{s.t.} \quad \begin{bmatrix}\bar{u}_{[-\rho,L-1]}(t)\\ \bar{y}_{[-\rho,L-1]}(t)\end{bmatrix} = \begin{bmatrix}H_{L+\rho}(u^d)\\ H_{L+\rho}(y^d)\end{bmatrix}\alpha(t)$$

$$\begin{bmatrix}\bar{u}_{[-\rho,-1]}(t)\\ \bar{y}_{[-\rho,-1]}(t)\end{bmatrix} = \begin{bmatrix}u_{[t-\rho,t-1]}\\ y_{[t-\rho,t-1]}\end{bmatrix}, \quad \begin{bmatrix}\bar{u}_{[L-\rho,L-1]}(t)\\ \bar{y}_{[L-\rho,L-1]}(t)\end{bmatrix} = \begin{bmatrix}0\\ y^{\mathrm{o}}\end{bmatrix}$$

$$\bar{u}_k(t)\in\mathcal{U}, \quad \bar{y}_k(t)\in\mathcal{Y}, \quad k\in[0,L-1)$$

Features of the problem

#1: **Equivalent** to model-based provided ρ is **big enough**
#2: **Stability** & **recursive feasibility** are guaranteed

(Berberich et al., 2021)

Back to basics: the model-based problem

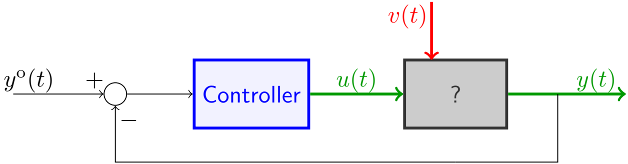
$$\begin{aligned}
\underset{\substack{\bar{u}_{[0,L-1]}(t)\\ \bar{y}_{[0,L-1]}(t)}}{\text{minimize}} \quad & \sum_{k=0}^{L-1} \underbrace{\mathbb{E}[\|\bar{y}_k(t)-y^{\mathrm{o}}\|_Q^2]+\|\bar{u}_k(t)\|_R^2}_{\ell(\bar{u}_k(t),\bar{y}_k(t))} \\
\text{s.t.} \quad & \bar{x}_{k+1}(t)=\mathbf{A}\bar{x}_k(t)+\mathbf{B}\bar{u}_k(t)+\mathbf{w_k}, \quad k\in[0,L-1) \\
& \bar{y}_k(t)=\mathbf{C}\bar{x}_k(t)+\mathbf{D}\bar{u}_k(t)+\mathbf{v_k}, \quad k\in[0,L-1) \\
& \mathbf{x_0(t)=x(t)} \\
& \bar{u}_k(t)\in\mathcal{U}, \ \ \mathbb{E}[\bar{y}_k(t)]\in\mathcal{Y}, \quad k\in[0,L-1)
\end{aligned}$$

How to use **data** instead of model parameters

How to cope with noise ?

How to cope with unknown initial conditions ?

Challenges with noisy outputs



How is the "true" system changed?

$$\begin{cases} x(t+1) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) + \mathbf{v(t)} \end{cases}$$

The **input** and the **model** do not uniquely define the **output** trajectory

DDPC under bounded measurement noise

Starting from the nominal formulation with terminal constraints

$$\underset{\substack{\bar{u}_{[0,L-1]}(t),\alpha(t)\\ \bar{y}_{[0,L-1]}(t),\sigma(t)}}{\text{minimize}} \quad \sum_{k=0}^{L-1} \ell(\bar{u}_k(t), \bar{y}_k(t)) + \lambda_\alpha \bar{\varepsilon} \|\alpha(t)\|^2 + \lambda_\sigma \|\sigma(t)\|^2$$

$$\text{s.t.} \quad \begin{bmatrix} \bar{u}_{[-\rho,L-1]}(t) \\ \bar{y}_{[-\rho,L-1]}(t) + \sigma(t) \end{bmatrix} = \begin{bmatrix} H_{L+\rho}(u^d) \\ H_{L+\rho}(y^d) \end{bmatrix} \alpha(t)$$

$$\begin{bmatrix} \bar{u}_{[-\rho,-1]}(t) \\ \bar{y}_{[-\rho,-1]}(t) \end{bmatrix} = \begin{bmatrix} u_{[t-\rho,t-1]} \\ y_{[t-\rho,t-1]} \end{bmatrix}, \quad \begin{bmatrix} \bar{u}_{[L-\rho,L-1]}(t) \\ \bar{y}_{[L-\rho,L-1]}(t) \end{bmatrix} = \begin{bmatrix} 0 \\ y^{\mathrm{o}} \end{bmatrix}$$

$$\bar{u}_k(t) \in \mathcal{U}, \quad \bar{y}_k(t) \in \mathcal{Y}, \quad k \in [0, L-1)$$

$$\|\sigma_k(t)\|_\infty \leq \bar{\varepsilon}(\|\alpha(t)\|_1 + 1), \; k \in [-\rho, L-1]$$

Practical stability and **recursive feasibility** are guaranteed by regularizing $\alpha(t)$ and introducing a slack

(Berberich et al., 2021)

DDPC under bounded measurement noise

Starting from the nominal formulation with terminal constraints

$$\underset{\substack{\bar{u}_{[0,L-1]}(t),\alpha(t)\\ \bar{y}_{[0,L-1]}(t),\sigma(t)}}{\text{minimize}} \quad \sum_{k=0}^{L-1} \ell(\bar{u}_k(t),\bar{y}_k(t)) + \lambda_\alpha \bar{\varepsilon}\|\alpha(t)\|^2 + \lambda_\sigma \|\sigma(t)\|^2$$

$$\text{s.t.} \quad \begin{bmatrix} \bar{u}_{[-\rho,L-1]}(t) \\ \bar{y}_{[-\rho,L-1]}(t) + \sigma(t) \end{bmatrix} = \begin{bmatrix} H_{L+\rho}(u^d) \\ H_{L+\rho}(y^d) \end{bmatrix} \alpha(t)$$

$$\begin{bmatrix} \bar{u}_{[-\rho,-1]}(t) \\ \bar{y}_{[-\rho,-1]}(t) \end{bmatrix} = \begin{bmatrix} u_{[t-\rho,t-1]} \\ y_{[t-\rho,t-1]} \end{bmatrix}, \quad \begin{bmatrix} \bar{u}_{[L-\rho,L-1]}(t) \\ \bar{y}_{[L-\rho,L-1]}(t) \end{bmatrix} = \begin{bmatrix} 0 \\ y^{\text{o}} \end{bmatrix}$$

$$\bar{u}_k(t) \in \mathcal{U}, \quad \bar{y}_k(t) \in \mathcal{Y}, \quad k \in [0, L-1)$$

$$\|\sigma_k(t)\|_\infty \leq \bar{\varepsilon}(\|\alpha(t)\|_1 + 1), \; k \in [-\rho, L-1]$$

#1: How to tune the regularization parameters?
#2: How to cope with the changes in the **penalties** induced by the regularization?

An alternative: α-regularization

Starting from the nominal formulation w/o terminal constraints

$$\underset{\substack{\bar{u}_{[0,L-1]}(t),\alpha(t)\\ \bar{y}_{[0,L-1]}(t)}}{\text{minimize}} \quad \sum_{k=0}^{L-1} \ell(\bar{u}_k(t),\bar{y}_k(t)) + \lambda_1\|\alpha(t)\|_1 + \lambda_2\|(I-\Pi)\alpha(t)\|^2$$

$$\text{s.t.} \quad \begin{bmatrix} \bar{u}_{[-\rho,L-1]}(t) \\ \bar{y}_{[-\rho,L-1]}(t) \end{bmatrix} = \begin{bmatrix} H_{L+\rho}(u^d) \\ H_{L+\rho}(y^d) \end{bmatrix} \alpha(t)$$

$$\begin{bmatrix} \bar{u}_{[-\rho,-1]}(t) \\ \bar{y}_{[-\rho,-1]}(t) \end{bmatrix} = \begin{bmatrix} u_{[t-\rho,t-1]} \\ y_{[t-\rho,t-1]} \end{bmatrix}$$

$$\bar{u}_k(t) \in \mathcal{U}, \quad \bar{y}_k(t) \in \mathcal{Y}, \quad k \in [0, L-1)$$

$(I-\Pi)$: orthogonal projector onto the kernel of the initial conditions and future outputs

Noise is coped with by **shrinking** α via 1-norm regularization and exploiting subspace identification inkling

(Dörfler et al., 2021)

An alternative: α-regularization

Starting from the nominal formulation w/o terminal constraints

$$\underset{\substack{\bar{u}_{[0,L-1]}(t),\alpha(t)\\ \bar{y}_{[0,L-1]}(t)}}{\text{minimize}} \quad \sum_{k=0}^{L-1} \ell(\bar{u}_k(t), \bar{y}_k(t)) + \lambda_1 \|\alpha(t)\|_1 + \lambda_2 \|(I-\Pi)\alpha(t)\|^2$$

$$\text{s.t.} \quad \begin{bmatrix} \bar{u}_{[-\rho,L-1]}(t) \\ \bar{y}_{[-\rho,L-1]}(t) \end{bmatrix} = \begin{bmatrix} H_{L+\rho}(u^d) \\ H_{L+\rho}(y^d) \end{bmatrix} \alpha(t)$$

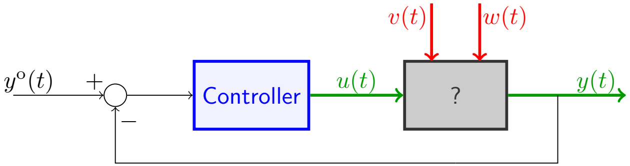
$$\begin{bmatrix} \bar{u}_{[-\rho,-1]}(t) \\ \bar{y}_{[-\rho,-1]}(t) \end{bmatrix} = \begin{bmatrix} u_{[t-\rho,t-1]} \\ y_{[t-\rho,t-1]} \end{bmatrix}$$

$$\bar{u}_k(t) \in \mathcal{U}, \quad \bar{y}_k(t) \in \mathcal{Y}, \quad k \in [0, L-1)$$

$(I-\Pi)$: orthogonal projector onto the kernel of the initial conditions and future outputs

#1: How to tune the regularization parameters?
#2: How to cope with the changes in the **penalties** induced by the regularization?

Shifting to a stochastic framework



How are data actually generated?

$$\begin{cases} x(t+1) = Ax(t) + Bu(t) + \mathbf{w(t)} \\ y(t) = Cx(t) + Du(t) + \mathbf{v(t)} \end{cases}$$

The **input** and the **model** do not define uniquely the **output** trajectory

From the model to its equivalent innovation form

For a better understanding on where the **noise enters** in the picture...

Initial model

$$\begin{cases} x(t+1) = Ax(t) + Bu(t) + w(t) \\ y(t) = Cx(t) + Du(t) + v(t) \end{cases}$$

$$\Downarrow$$

Innovation form

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) + Ke(t) \\ y(k) = Cx(k) + Du(k) + e(k) \end{cases}$$

Shifting to the prediction form it holds...

The maximum eigenvalue $\lambda_{\max}$ of $A - KC$ satisfy $|\lambda_{\max}| < 1$

Initial conditions as functions of inputs/outputs with noise

$$x(t) = (A - KC)^{\rho} x(t-\rho) + \mathcal{C} \begin{bmatrix} u_{[t-\rho,t-1]} \\ y_{[t-\rho,t-1]} \end{bmatrix}$$

The **past trajectories** $u_{[t-\rho,t-1]}$ and $y_{[t-\rho,t-1]}$ are noisy

$$\Downarrow$$

$$x(t) = \mathcal{C} \begin{bmatrix} u_{[t-\rho,t-1]} \\ y_{[t-\rho,t-1]} \end{bmatrix} + \underbrace{O(|\lambda_{max}|^{\rho})}_{\to 0 \text{ for } \rho \to \infty}$$

Initial conditions as functions of inputs/outputs with noise

$$x(t) = (A - KC)^\rho x(t-\rho) + \mathcal{C} \begin{bmatrix} u_{[t-\rho,t-1]} \\ y_{[t-\rho,t-1]} \end{bmatrix}$$

⬇

$$x(t) = \mathcal{C} \begin{bmatrix} u_{[t-\rho,t-1]} \\ y_{[t-\rho,t-1]} \end{bmatrix} + \underbrace{O(|\lambda_{max}|^\rho)}_{\to 0 \text{ for } \rho\to\infty}$$

- Link between the error performed in reconstructing $x(t)$ from input/output **data** and ρ
- Trade-off: ρ must be low due to computational/memory constraints and predictor variance

Initial conditions as functions of inputs/outputs with noise

$$x(t) = (A - KC)^{\rho} x(t-\rho) + \mathcal{C} \begin{bmatrix} u_{[t-\rho,t-1]} \\ y_{[t-\rho,t-1]} \end{bmatrix}$$

$$x(t) = \mathcal{C} \begin{bmatrix} u_{[t-\rho,t-1]} \\ y_{[t-\rho,t-1]} \end{bmatrix} + \underbrace{O(|\lambda_{max}|^{\rho})}_{\to 0 \text{ for } \rho \to \infty}$$

Tune ρ with AIC or other standard criteria in *system identification*

Recasting the model with subspace identification

We decompose the **Hankel matrices** $H_{L+\rho}(u^d)$ and $H_{L+\rho}(y^d)$

Past

$$Z_P = \begin{bmatrix} U_P \\ Y_P \end{bmatrix} \rightarrow \begin{array}{l} U_P = H_{[0,\rho-1],M}(u^d) \\ Y_P = H_{[0,\rho-1],M}(y^d) \end{array}$$

Future

$$U_F = H_{[\rho,L+\rho-1],M}(u^d)$$
$$Y_P = H_{[\rho,L+\rho-1],M}(y^d)$$
$$E_F = H_{[\rho,L+\rho-1],M}(e^d)$$

Future output sequence

$$Y_F = \Gamma \underbrace{X_\rho}_{\mathcal{C}Z_P + O(|\lambda_{max}|^\rho)} + \mathcal{H}_d U_F + \mathcal{H}_s E_F$$

Removing future noises by projection

Future outputs based on past inputs/outputs and future inputs

$$Y_F = \Gamma[\mathcal{C}Z_P + O(|\lambda_{max}|^\rho)] + \mathcal{H}_d U_F + \mathcal{H}_s E_F$$

$$\Downarrow$$

Projecting the future outputs onto the row span of Z_P and U_F

$$\Pi_{Z_P,U_F}(Y_F) = \hat{Y}_F = \Gamma\mathcal{C}Z_P + \mathcal{H}_d U_F + \mathcal{H}_s \underbrace{\Pi_{Z_P,U_F}(E_F)}_{O(\sqrt{\log(\log(N^d))/N^d})}$$

Data-driven predictive model

$$\underbrace{\hat{y}_{[t,t+L-1]}}_{\hat{Y}_F\alpha} = \mathcal{C}\underbrace{\begin{bmatrix} u_{[t-\rho,t-1]} \\ y_{[t-\rho,t-1]} \end{bmatrix}}_{\xi(t)=Z_P\alpha} + \mathcal{H}_d \underbrace{u_{[t,t+L-1]}}_{U_F\alpha} + \varepsilon(\rho, N^d), \quad \forall\alpha$$

Towards a constrained SPC formulation

Theorem (Breschi et al., 2022)

If the input sequence $\{u^d(t)\}_{t=1}^{N^d}$ is **persistently exciting**, for **any past** input/output trajectory $\xi(t)$, **future** input sequence $u_{[t,t+L-1]}$, it holds that

$$\hat{y}_{[t,t+L-1]} = \hat{Y}_F \alpha^\star + O_P\left(\frac{1}{\sqrt{N^d}}\right)$$

and

$$\alpha^\star \text{ satisfies } \begin{bmatrix} \xi(t) \\ u_{[t,t+L-1]} \end{bmatrix} = \begin{bmatrix} Z_P \\ U_F \end{bmatrix} \alpha$$

Recasting the control loss

$$\mathbb{E}\left[\|y_k(t) - y^o\|_Q^2\right] = \|\underbrace{\mathbb{E}\left[y_k(t)\right]}_{\hat{y}_k(t)} - y^o\|_Q^2 + \underbrace{\mathbb{E}\left[\|y_k(t) - \mathbb{E}\left[y_k(t)\right]\|_Q^2\right]}_{\text{independent from } u_k(t)}$$

Constrained SPC

Exploiting the previous relations...

$$\begin{aligned} \underset{\substack{\bar{u}_{[0,L-1]}(t) \\ \bar{y}_{[0,L-1]}(t)}}{\text{minimize}} \quad & \sum_{k=0}^{L-1} \underbrace{\|\bar{y}_k(t) - y^{\text{o}}\|_Q^2 + \|\bar{u}_k(t)\|_R^2}_{\ell(\bar{u}_k(t), \bar{y}_k(t))} \\ \text{s.t.} \quad & \alpha^\star = \begin{bmatrix} Z_P \\ U_F \end{bmatrix}^\dagger \begin{bmatrix} \xi(t) \\ \bar{u}_{[0,L-1]}(t) \end{bmatrix} \\ & \bar{y}_{[0,L-1]}(t) = \hat{Y}_F \alpha^\star \\ & \bar{u}_k(t) \in \mathcal{U}, \quad \bar{y}_k(t) \in \mathcal{Y}, \quad k \in [0, L-1) \end{aligned}$$

Equal to...

Existing subspace predictive control schemes

(Favoreel et al., 1999; Fiedler et al., 2021)

Constrained SPC

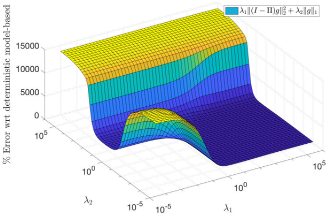
Exploiting the previous relations...

$$\underset{\substack{\bar{u}_{[0,L-1]}(t)\\ \bar{y}_{[0,L-1]}(t)}}{\text{minimize}} \quad \sum_{k=0}^{L-1} \ell(\bar{u}_k(t), \bar{y}_k(t))$$

$$\text{s.t.} \quad \alpha^\star = \begin{bmatrix} Z_P \\ U_F \end{bmatrix}^\dagger \begin{bmatrix} \xi(t) \\ \bar{u}_{[0,L-1]}(t) \end{bmatrix}$$

$$\bar{y}_{[0,L-1]}(t) = \hat{Y}_F \alpha^\star$$

$$\bar{u}_k(t) \in \mathcal{U}, \bar{y}_k(t) \in \mathcal{Y}, \;\; k \in [0, L-1)$$



(Dörfler et al., 2021)

Equivalent to...

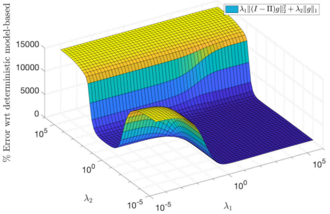
The regularized approach using **shrinkage** and **subspace identification** inkling, for $\lambda_1 = 0$ and $\lambda_2 \to \infty$

(Breschi et al., 2022)

Constrained SPC

Exploiting the previous relations...

$$\begin{aligned}
\underset{\substack{\bar{u}_{[0,L-1]}(t)\\ \bar{y}_{[0,L-1]}(t)}}{\text{minimize}} \quad & \sum_{k=0}^{L-1} \ell(\bar{u}_k(t), \bar{y}_k(t)) \\
\text{s.t.} \quad & \alpha^\star = \begin{bmatrix} Z_P \\ U_F \end{bmatrix}^\dagger \begin{bmatrix} \xi(t) \\ \bar{u}_{[0,L-1]}(t) \end{bmatrix} \\
& \bar{y}_{[0,L-1]}(t) = \hat{Y}_F \alpha^\star \\
& \bar{u}_k(t) \in \mathcal{U}, \bar{y}_k(t) \in \mathcal{Y}, \;\; k \in [0, L-1)
\end{aligned}$$



(Dörfler et al., 2021)

We get an **indication** on how to tune the regularization parameters for regularized DDPC schemes

γ-DDPC: towards a numerically efficient implementation

Starting again from the approaches where $\alpha(t)$ is **optimized**

How can we exploit the **previous** results to enhance the efficiency of these schemes?

The steps we perform are:

#1: LQ decomposition of the Hankel matrices

$$\begin{bmatrix} Z_P \\ U_F \\ Y_F \end{bmatrix} \alpha(t) = \begin{bmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix} \alpha(t) = \begin{bmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{bmatrix} \begin{bmatrix} \gamma_1(t) \\ \gamma_2(t) \\ \gamma_3(t) \end{bmatrix}$$

#2: set $\gamma_3(t) = 0$ (Results stemming from the projections)

(Breschi et al., 2022)

The steps of γ-DDPC

Since

$$\begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \\ L_{31} & L_{32} \end{bmatrix} \begin{bmatrix} \gamma_1(t) \\ \gamma_2(t) \end{bmatrix} = \begin{bmatrix} \xi(t) \\ \bar{u}_{[0,L-1]}(t) \\ \bar{y}_{[0,L-1]}(t) \end{bmatrix}$$

Account for the initial conditions

$$\gamma_1^\star(t) = L_{11}^{-1}\xi(t)$$

We can **decouple** the problem of *fitting* the initial conditions to that of **optimizing** the control action

The steps of γ-DDPC

Since

$$\begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \\ L_{31} & L_{32} \end{bmatrix} \begin{bmatrix} \gamma_1(t) \\ \gamma_2(t) \end{bmatrix} = \begin{bmatrix} \xi(t) \\ \bar{u}_{[0,L-1]}(t) \\ \bar{y}_{[0,L-1]}(t) \end{bmatrix}$$

Satisfying constraints

$$\begin{aligned} \underset{\substack{\bar{u}_{[0,L-1]}(t), \gamma_2(t) \\ \bar{y}_{[0,L-1]}(t)}}{\text{minimize}} \quad & \sum_{k=0}^{L-1} \ell(\bar{u}_k(t), \bar{y}_k(t)) \\ \text{s.t.} \quad & \begin{bmatrix} \bar{u}_{[0,L-1](t)} \\ \bar{y}_{[0,L-1](t)} \end{bmatrix} = \begin{bmatrix} L_{21} & L_{22} \\ L_{31} & L_{32} \end{bmatrix} \begin{bmatrix} \gamma_1^\star(t) \\ \gamma_2(t) \end{bmatrix} \\ & \bar{u}_k(t) \in \mathcal{U}, \bar{y}_k(t) \in \mathcal{Y}, \; k \in [0, L-1) \end{aligned}$$

The steps of γ-DDPC

Account for the initial conditions

$$\gamma_1^\star(t) = L_{11}^{-1}\xi(t)$$

Satisfying constraints

$$\begin{aligned}
\underset{\substack{\bar{u}_{[0,L-1]}(t),\gamma_2(t)\\ \bar{y}_{[0,L-1]}(t)}}{\text{minimize}} \quad & \sum_{k=0}^{L-1} \ell(\bar{u}_k(t), \bar{y}_k(t)) \\
\text{s.t.} \quad & \begin{bmatrix} \bar{u}_{[0,L-1](t)} \\ \bar{y}_{[0,L-1](t)} \end{bmatrix} = \begin{bmatrix} L_{21} & L_{22} \\ L_{31} & L_{32} \end{bmatrix} \begin{bmatrix} \gamma_1^\star(t) \\ \gamma_2(t) \end{bmatrix} \\
& \bar{u}_k(t) \in \mathcal{U}, \bar{y}_k(t) \in \mathcal{Y},\ k \in [0, L-1)
\end{aligned}$$

The **error** in using the data-driven predictive model is linked to the **number of data** and the choice of ρ

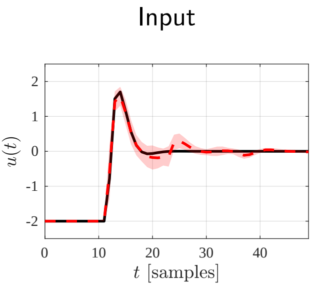
(Breschi et al., 2022)

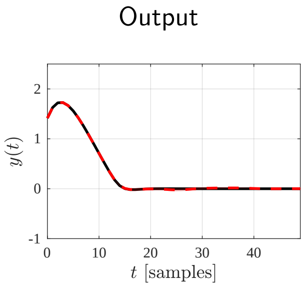
Benchmark example: γ-DDPC *vs* oracle

Benchmark example

$$\begin{cases} x(t+1) = \begin{bmatrix} 0.7326 & -0.0861 \\ 0.1722 & 0.9909 \end{bmatrix} x(t) + \begin{bmatrix} 0.0609 \\ 0.0064 \end{bmatrix} u(t) + \begin{bmatrix} 0.9089 \\ 0.4838 \end{bmatrix} e(t) \\ y(t) = \begin{bmatrix} 0 & 1.4142 \end{bmatrix} x(t) + e(t) \end{cases}$$

$e \sim \mathcal{N}(0, 0.01)$ (Bemporad et al., 2002)

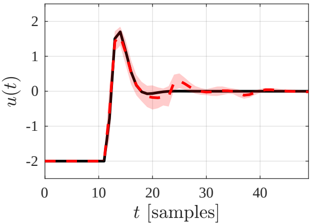




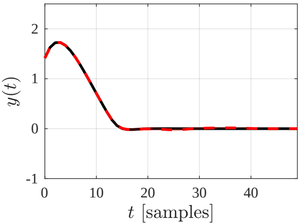
$\bar{\text{SNR}} = 11$ dB

Benchmark example: γ-DDPC *vs* oracle

Input



Output



The **mean** inputs and outputs over 30 Monte Carlo simulations (- -) are *close* to the ones resulting from using the oracle MPC (-)

$\bar{\text{SNR}} = 11$ dB

Benchmark example: γ-DDPC *vs* 2-norm on γ_2

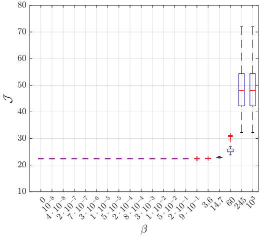
Benchmark example

$$\begin{cases} x(t+1) = \begin{bmatrix} 0.7326 & -0.0861 \\ 0.1722 & 0.9909 \end{bmatrix} x(t) + \begin{bmatrix} 0.0609 \\ 0.0064 \end{bmatrix} u(t) \\ y(t) = \begin{bmatrix} 0 & 1.4142 \end{bmatrix} x(t) + e(t) \end{cases}$$

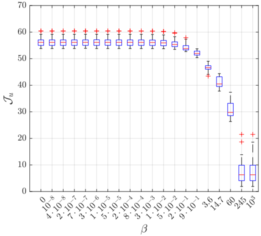
$e \sim \mathcal{N}(0, 0.01)$ (Bemporad et al., 2002)

Are there benefits in introducing $\beta\|\gamma_2(t)\|^2$ in our cost?

Attained *vs* oracle cost



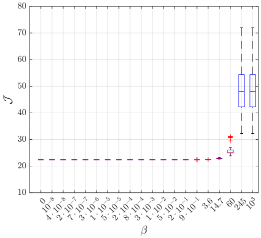
Attained *vs* oracle input cost



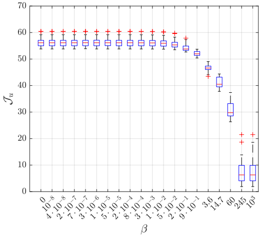
Benchmark example: γ-DDPC *vs* 2-norm on γ_2

Are there benefits in introducing $\beta\|\gamma_2(t)\|^2$ in our cost?

Attained *vs* oracle cost



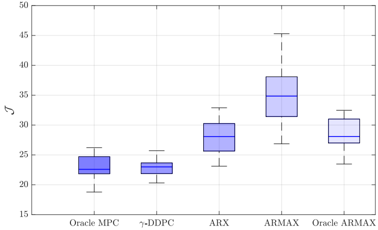
Attained *vs* oracle input cost



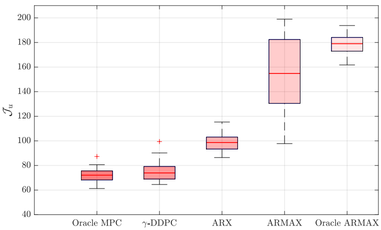
The **lower** β, the **more** the closed-loop behavior over 30 Monte Carlo simulation resembles the one induced by the *oracle MPC* ($\mathcal{J} = 22.34$ and $\mathcal{J}_u = 55.46$)

Benchmark example: DD *vs* MB

Output cost



Input cost

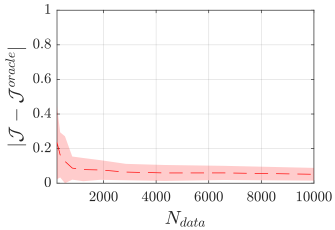


Closed-loop validation tests ($\overline{\text{SNR}} = 18$ dB): performance indexes *vs* predictive strategy over 30 Monte Carlo predictors

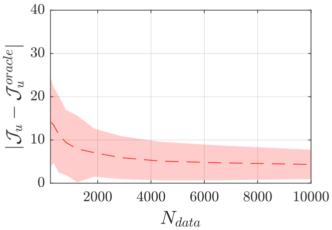
Benchmark example: γ-DDPC *vs* N^d

What is the effect induced by the number of available samples?

Attained *vs* oracle cost



Attained *vs* oracle input cost



The **larger** is the dataset, the **more** the closed-loop behavior over 30 Monte Carlo simulation resembles the one induced by the *oracle MPC*

Can we make γ-DDPC more "robust"?

- What if N^d is low?

⬇

$$\underset{\substack{\bar{u}_{[0,L-1]}(t),\gamma_2(t)\\ \bar{y}_{[0,L-1]}(t)}}{\text{minimize}} \quad \sum_{k=0}^{L-1} \ell(\bar{u}_k(t), \bar{y}_k(t)) + \lambda_\sigma \|\sigma(t)\|^2$$

$$\text{s.t.} \quad \begin{bmatrix} \bar{u}_{[0,L-1](t)} \\ \bar{y}_{[0,L-1](t)} + \sigma(t) \end{bmatrix} = \begin{bmatrix} L_{21} & L_{22} \\ L_{31} & L_{32} \end{bmatrix} \begin{bmatrix} \gamma_1^\star(t) \\ \gamma_2(t) \end{bmatrix}$$

$$\bar{u}_k(t) \in \mathcal{U}, \bar{y}_k(t) \in \mathcal{Y}, \;\; k \in [0, L-1)$$

We add a **slack** to account for the entity of our approximations

Tunable parameters and their interpretation

Also this scheme requires some hyper-parameters to be **tuned**

$$\begin{aligned}
\underset{\substack{\bar{u}_{[0,L-1]}(t),\gamma_2(t)\\ \bar{y}_{[0,L-1]}(t)}}{\text{minimize}} \quad & \sum_{k=0}^{L-1} \ell(\bar{u}_k(t),\bar{y}_k(t)) + \lambda_\sigma \|\sigma(t)\|^2 \\
\text{s.t.} \quad & \begin{bmatrix} \bar{u}_{[0,L-1](t)} \\ \bar{y}_{[0,L-1](t)} + \sigma(t) \end{bmatrix} = \begin{bmatrix} L_{21} & L_{22} \\ L_{31} & L_{32} \end{bmatrix} \begin{bmatrix} \gamma_1^\star(t) \\ \gamma_2(t) \end{bmatrix} \\
& \begin{bmatrix} \bar{u}_{[L-n,L-1](t)} \\ \bar{y}_{[L-n,L-1](t)} \end{bmatrix} = \begin{bmatrix} 0 \\ y^{\mathrm{o}} \end{bmatrix} \\
& \bar{u}_k(t) \in \mathcal{U}, \bar{y}_k(t) \in \mathcal{Y},\ k \in [0, L-1)
\end{aligned}$$

- λ_σ depends on our dataset and the choice of ρ ➜ $\lambda_\sigma \propto \frac{N^d}{\rho \log(\log(N^d))}$
- n should leave **enough freedom** for the optimization of the input
  ➜ $n < \rho$ (still greater or equal to the order of the system)

An alternative regularization scheme

Non-asymptotic behaviour

$$\underset{\substack{\bar{u}_{[0,L-1]}(t),\gamma_2(t)\\ \bar{y}_{[0,L-1]}(t)}}{\text{minimize}} \quad \sum_{k=0}^{L-1} \ell(\bar{u}_k(t), \bar{y}_k(t)) + \beta_2 \|\gamma_2\|^2$$

$$\text{s.t.} \quad \begin{bmatrix} \bar{u}_{[0,L-1](t)} \\ \bar{y}_{[0,L-1](t)} \end{bmatrix} = \begin{bmatrix} L_{21} & L_{22} \\ L_{31} & L_{32} \end{bmatrix} \begin{bmatrix} \gamma_1^\star(t) \\ \gamma_2(t) \end{bmatrix}$$

$$\begin{bmatrix} \bar{u}_{[L-n,L-1](t)} \\ \bar{y}_{[L-n,L-1](t)} \end{bmatrix} = \begin{bmatrix} 0 \\ y^{\mathrm{o}} \end{bmatrix}$$

$$\bar{u}_k(t) \in \mathcal{U}, \bar{y}_k(t) \in \mathcal{Y}, \; k \in [0, L-1)$$

- β_2 now has a different role: to keep the norm of γ_2 small (with $\gamma_3 = 0$)
- In fact, $var(\text{error}) \simeq T \frac{\|\gamma_1^*\|^2 + \|\gamma_2^*(\beta_2)\|^2}{N}$ (tunable via linear search)

An alternative regularization scheme

Non-asymptotic behaviour

$$\underset{\substack{\bar{u}_{[0,L-1]}(t),\gamma_2(t)\\ \bar{y}_{[0,L-1]}(t)}}{\text{minimize}} \quad \sum_{k=0}^{L-1} \ell(\bar{u}_k(t), \bar{y}_k(t)) + \beta_3\|\gamma_3\|^2$$

$$\text{s.t.} \quad \begin{bmatrix} \bar{u}_{[0,L-1](t)} \\ \bar{y}_{[0,L-1](t)} \end{bmatrix} = \begin{bmatrix} L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{bmatrix} \begin{bmatrix} \gamma_1^\star \\ \gamma_2 \\ \gamma_3 \end{bmatrix}$$

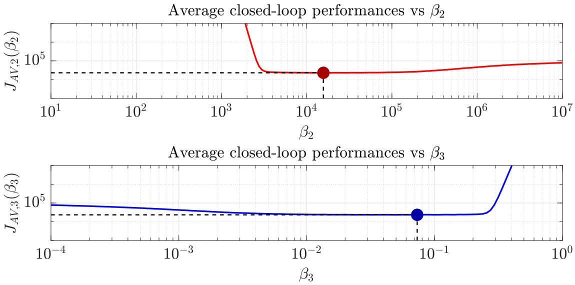
$$\begin{bmatrix} \bar{u}_{[L-n,L-1](t)} \\ \bar{y}_{[L-n,L-1](t)} \end{bmatrix} = \begin{bmatrix} 0 \\ y^{\mathrm{o}} \end{bmatrix}$$

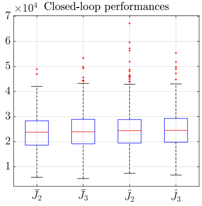
$$\bar{u}_k(t) \in \mathcal{U}, \bar{y}_k(t) \in \mathcal{Y}, \; k \in [0, L-1)$$

- β_3 now has a different role: to keep the norm of γ_3 small. In fact, $var(\text{error}) \simeq L_{33}\gamma_3$ (linear search: $\|\gamma_3^*(\beta_3)\|^2 \simeq T\frac{\|\gamma_1^*\|^2+\|\gamma_2^*(\beta_3)\|^2}{N}$)

Benchmark example: performance and optimal coefficients

Optimal coefficients in case of $N_d = 250$





Practically indistinguishable from oracle-type tuning based on off-line closed-loop experiments.

Conclusions

Existing approaches for the design of predictive controllers from data:

- Deterministic setting
- Measurement noise only

For the **stochastic setting**, we proposed a numerically efficient approach (γ-DDPC)

- **Decoupling** initial conditions' fitting and control design
- **Reducing** the number of optimization variables
- Regularization can be tuned **off-line**

Conclusions

Existing approaches for the design of predictive controllers from data:

For the **stochastic setting**, we proposed a numerically efficient approach (γ-DDPC)

Ongoing works:

- **Terminal ingredients** and stability guarantees
- Hyper-parameters tuning

Thank you!