A practitioner's guide to noise handling strategies in data-driven predictive control

Citation for published version (APA):
Sassella, A., Breschi, V., & Formentin, S. (2023). A practitioner's guide to noise handling strategies in data-driven predictive control. *IFAC-PapersOnLine*, *56*(2), 1382-1387. https://doi.org/10.1016/j.ifacol.2023.10.1795

DOI:
10.1016/j.ifacol.2023.10.1795

Document status and date:
Published: 01/07/2023

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 56-2 (2023) 1382–1387

A practitioner's guide to noise handling strategies in data-driven predictive control ⋆

Andrea Sassella *, **Valentina Breschi** **, **Simone Formentin** *

* *Politecnico di Milano, P.zza L. da Vinci 32, 20133 Milano, Italy (e-mail: name.surname@polimi.it).*

** *Eindhoven University of Technology, 5600 MB, Eindhoven, Netherlands (e-mail: v.breschi@tue.nl).*

Abstract: Today's increasing availability of data is having a remarkable impact on control design. However, for data-driven control approaches to become widespread in practical applications, it is necessary to devise strategies that can effectively handle the presence of noise in the data used to design the controller. In this work, we analyse the existing approaches to deal with noisy measurements in data-driven predictive control (DDPC) and we highlight the advantages and downsides of each technique from a practitioner's perspective. Our qualitative conclusions are supported by the results obtained from two benchmark examples.

Keywords: Data-driven control, Predictive Control, Noise handling.

1. INTRODUCTION

The shift from model-based to data-driven control is now an affirmed trend within the control community. Indeed, in the last years quite an effort has been devoted to adapt the principles of Model Predictive Control (MPC) to a completely data-based framework (see *e.g.*, Coulson et al. (2019); Berberich et al. (2020); Breschi et al. (2022); Sassella et al. (2022a)), leading to a novel *Data-Driven Predictive Control* (DDPC) theory. At the core of this transition lays the result in Willems et al. (2005), which allows one to obtain a data-based characterization of the dynamics of a system, without explicitly requiring any preliminary identification step. Even if MPC is now a standard in industrial applications Forbes et al. (2015), only some attempts have instead been performed to apply its data-driven counterpart in real-world applications (see, *e.g.*, Huang et al. (2021)), and a lot has still to be done for these techniques to be extensively used in practice. In particular, since DDPC strategies rely on real-world measurements to construct a data-based predictor, *noise handling techniques* have a major role in making them viable in practice. Indeed, if not properly managed, noise can have dramatic consequences on closed-loop performance.

Existing techniques to deal with noisy data can be divided in two main classes. First, one may *pre-process* the employed data as in Sassella et al. (2022a) and Sassella et al. (2022b), respectively relying on repeated data collection campaigns and manipulation of data matrices via Dynamic Mode Decomposition (DMD) (see Kutz et al. (2016)) prior to the controller deployment. As an alternative approach, *regularization* has been proposed, see, *e.g.*, Berberich et al. (2020); Dörfler et al. (2022). Specifically, Dörfler et al. (2022) outlines different regularization strategies to handle noise, ranging from standard and weighted Tikhonov regularization to Lasso and elastic nets (Hastie et al. (2009)). By leveraging on the assumption of additive noise, L_2 regularization is instead proposed in Berberich et al. (2020); Huang et al. (2021), along with the inclusion of a slack variable acting on the output predictions. Under the assumption of bounded noise, in Berberich et al. (2020) the control problem is further augmented with a non-convex constraint on the slack, which can nonetheless be neglected by properly selecting the regularization penalties. Regularization is also used in Coulson et al. (2021) within a chance-constrained framework to cope with noise in the data, while robustly handling output constraints. Instead, Yin et al. (2023) proposes a DDPC that uses an approximated *maximum likelihood estimator* as predictor, overcoming the need to introduce regularization terms in the cost.

In this work, we review noise handling strategies for data-driven predictive control proposed in the literature from a perspective aligned with that of a practitioner. For each family of techniques, we thus outline the main features, advantages and drawbacks and practically show them through two numerical benchmark examples. The paper is organized as follows. The considered framework is introduced in 2. The existing strategies to handle noisy data are then analyzed in Section 3, with advantages and drawbacks illustrated numerically in Section 4.

2. DATA-DRIVEN SYSTEMS AND CONTROL

Consider a *discrete-time, linear, time invariant* (LTI) system, described by the following set of equations

$$\mathcal{S}: \begin{cases} x(t+1) = Ax(t) + Bu(t), \\ y^{o}(t) = Cx(t), \end{cases} \tag{1}$$

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$ and $y^{o}(t) \in \mathbb{R}^p$ are the state, exogenous input and *noiseless* output of the system at time $t \in \mathbb{N}$, respectively. Let the system be *controllable* and

⋆ This project was partially supported by the Italian Ministry of University and Research under the PRIN'17 project "Data-driven learning of constrained control systems", contract no. 2017J89ARP.

2405-8963 Copyright © 2023 The Authors. This is an open access article under the CC BY-NC-ND license.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2023.10.1795

observable, but its matrices $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}, C \in \mathbb{R}^{p\times n}$ be *unknown*. Meanwhile, let us assume that we have access to a batch sequence of *input/output* data pairs $\mathcal{D}_T = \{u(t), y(t)\}_{t=0}^{T-1}$, with the available output measurements being corrupted by noise, *i.e.*,

$$y(t) = y^{o}(t) + v(t), \tag{2}$$

where $v(t) \in \mathbb{R}^p$ is the realization at time t of a white, zero mean stochastic process. Assume that we have no additional insights on the noise corrupting the data. Moreover, suppose the input fed to the system during the data collection phase, *i.e.*, the sequence $\mathcal{U}_T = \{u(t)\}_{t=0}^{T-1}$, is *persistently exciting* of a sufficiently high order, according to the following definition (Willems et al. (2005)).

***Definition** 1.* (Persistence of excitation).
The sequence $\mathcal{U}_T$ is persistently exciting of order M if the associated Hankel matrix $U_{0,M,T} \in \mathbb{R}^{mN\times(T-M-1)}$, *i.e.*,

$$U_{0,M,T} = \begin{bmatrix} u(0) & u(1) & \cdots & u(T-M-1) \\ u(1) & u(2) & \cdots & u(T-M) \\ \vdots & \vdots & \ddots & \vdots \\ u(M-1) & u(M) & \cdots & u(T) \end{bmatrix}, \tag{3}$$

is full row rank, namely $\text{rank}(U_{0,M,T}) = mM$.

This notion of persistence of excitation is the key to obtain an informative, yet purely data-based, representation of the system under control. We now review the main results leading to two alternative data-driven descriptions of the unknown system. In both cases, the final predictor depends on a upper-bound $\eta > n$ of the true order of $\mathcal{S}$, which is here assumed to be *unknown*.

2.1 One-step ahead predictor

At time t, let us consider the following collection of past inputs and outputs of $\mathcal{S}$, *i.e.*,

$$\chi(t) = \left[u(t-\eta)' \cdots\ u(t-1)'\ y(t-\eta)' \cdots\ y(t-1)'\right]', \tag{4}$$

which can be seen as the state of a *non-minimal* realization of the unknown system. Furthermore, let us introduce the following concatenation of data matrices:

$$\Omega = \begin{bmatrix} U_{0,1,T-1} \\ \hat{X}_{0,T-1} \end{bmatrix} \in \mathbb{R}^{((\eta+1)m+\eta p)\times T} \tag{5}$$

where $\hat{X}_{0,T-1} = [\chi(0)\ \chi(1)\ \cdots\ \chi(T-1)] \in \mathbb{R}^{(p+m)\eta\times T}$. When the batch input sequence $\mathcal{U}_T$ is persistently exciting of order $(m+p)\eta+1$ and the data sequence is long enough (*i.e.*, $T \geq (m+1)(m+p)\eta + m$), it can be proven (see De Persis and Tesi (2020)) that

$$\text{rank}\,[\Omega] = (\eta+1)m + p\eta. \tag{6}$$

Accordingly, the one-step-ahead predictor at $k \in \mathbb{N}$ can be defined as follows De Persis and Tesi (2020):

$$\bar{\chi}(k+1|\mathcal{D}_T) = \hat{X}_{1,T}\Omega^{\dagger}\begin{bmatrix} u(k) \\ \chi(k) \end{bmatrix}, \tag{7a}$$

where $\hat{X}_{1,T} = [\chi(1)\ \chi(2)\ \cdots\ \chi(T)] \in \mathbb{R}^{(p+m)\eta\times T}$, and the predicted output is given by

$$\bar{y}(k|\mathcal{D}_T) = e_\eta'\hat{X}_{1,T}\Omega^{\dagger}\begin{bmatrix} 0_{1\times 2\eta} \\ I_{2\eta} \end{bmatrix}\chi(k), \tag{7b}$$

with e_η being the η-th versor of $\mathbb{R}^{2\eta}$. Note that, while in the *noiseless case* it can be proven that $\hat{\chi}(k+1|\mathcal{D}_T)$ exactly corresponds to the true extended state $\chi(k+1)$ (see De Persis and Tesi (2020)), the same conclusion does not directly follows in the presence of noise. In this challenging, yet more realistic, case, it thus becomes crucial to pair this identification-like result with noise handling strategies, to obtain a predictor actually describing the evolution of the underlying (unknown) system.

Table 1. Requirements on data collection & control problem features.

	One-step ahead (7)	Trajectory-based (9)
Pers. of excitation	$(m+p)\eta+1$	$L+2\eta$
T_{min}	$(m+1)(m+p)\eta+m$	$(m+1)(L+2\eta)-1$
n. of unknowns	mL	$T-L-\eta-1$

2.2 Trajectory-based predictor

Instead of a one-step-ahead predictor, an L-step predictor can be obtained as follows.

Theorem 1. (Trajectory-based representation). Let $\mathcal{D}_T = \{\mathcal{U}_T, \mathcal{Y}_T\}$ be a *noiseless* measured trajectory of $\mathcal{S}$ in (1), with $\mathcal{Y}_T = \{y(t)\}_{t=0}^{T-1}$. Assume $\mathcal{U}_T$ to be persistently exciting of order $L+\eta$. Then, $\{\bar{u}(k), \bar{y}(k)\}_{k=0}^{L-1}$ is a trajectory of $\mathcal{S}$ *if and only if* there exist a vector $\alpha \in \mathbb{R}^{T-L-1}$ such that:

$$\begin{bmatrix} \bar{u}_{[0,L-1]}(\mathcal{D}_T) \\ \bar{y}_{[0,L-1]}(\mathcal{D}_T) \end{bmatrix} = \begin{bmatrix} U_{0,L,T-1} \\ Y_{0,L,T-1} \end{bmatrix}\alpha, \tag{8}$$

with $Y_{0,L,T-1} \in \mathbb{R}^{pL\times T-L-1}$ being the Hankel matrix associated to $\mathcal{Y}_T$, and $\bar{u}_{[0,L-1]}(\mathcal{D}_T) \in \mathbb{R}^{mL}$ and $\bar{y}_{[0,L-1]}(\mathcal{D}_T) \in \mathbb{R}^{pL}$ being the column vectors stacking the predicted input and output trajectories of length L, respectively. □

Note that, in this case, the dynamics of the true system $\mathcal{S}$ is encapsulated in the parameter α. Since considering initial conditions is crucial in the solution of a predictive control problem, he behavioral model in (8) can be further augmented as:

$$\begin{bmatrix} \bar{u}_{[-\eta,L-1]}(\mathcal{D}_T) \\ \bar{y}_{[-\eta,L-1]}(\mathcal{D}_T) \end{bmatrix} = \begin{bmatrix} U_{0,L+\eta,T-1} \\ Y_{0,L+\eta,T-1} \end{bmatrix}\alpha. \tag{9}$$

In turn, differently from Theorem 1, this requires the input sequence $\mathcal{U}_T$ to be persistently exciting of order $L+2\eta$[1].

As for the one-step-ahead predictor, the accuracy of (9) might be undermined by the output measurements noise. As such, also in this scenario, it is crucial to counteract its effects by using suitable noise handling techniques.

2.3 Comparing the two data-driven representations

When selecting the DDPC approach to use, it is important to unveil the similarities between the predictors presented beforehand, and to understand how their differences might impact on the solution of the DDPC problem.

The two predictors require *different experiment design* for the data collection phase. By looking at Table 1, it is clear that the different experimental requirements on the batch input sequence $\mathcal{U}_T$ stem from the nature of the two predictors. On the one hand, the level of excitation needed to construct (7) is directly linked to the input/output

[1] This condition can be relaxed to $L+\eta+\zeta$, where ℓ is the smallest integer such that the observability matrix $O_\ell(A,C) := col(C, CA, \ldots, CA^{\zeta-1})$ has rank n

dimensions and the chosen parameter η, as the definition of the non-minimal state (4) depends on the latter. On the other hand, the level of excitation needed to build (9) is driven by the length of the prediction horizon L and, once again, by η. This difference is reflected in the minimum experiment length T_{min}, which is, in turn, shaped by the features of the input Hankel matrices characterizing the two predictors. Concerning the solution of the predictive control problem, using (7) entails optimizing mL variables, as in a model-based setting. Instead, as reported in Table 1, the actual unknown of (9) is α. As such, the number of optimization variables is shaped by the length of the dataset and η, while not being directly connected to the dimension of the input fed to the system. This can give rise to *computational issues* for large datasets, which are not fully shared by (7), as Ω^{-1} can be computed offline.

2.4 Deterministic predictive control

Without loss of generality, let us formulate the nominal DDPC problems focusing on the zero regulation problem [2]. Depending on the employed predictor, we end up with two different, although conceptually equivalent, formulations of the DDPC problem.

Let $L > 0$ be the prediction horizon, and $\mathbb{U} \subseteq \mathbb{R}^m$ and $\mathbb{Y} \subseteq \mathbb{R}^p$ be the sets specifying the input and output constraints, respectively. When using the predictor in (7), the control problem can be cast as follows:

$$\min_{\bar{u}_{[0,L-1]}} \sum_{k=0}^{L-1} \|\bar{y}(t+k)\|_Q^2 + \|\bar{u}(t+k)\|_R^2 \tag{10a}$$

$$\text{s.t.}\ \ \bar{\chi}(t+k+1)=\hat{X}_{1,T}\Omega^{\dagger}\begin{bmatrix}\bar{u}(t+k)\\ \bar{\chi}(t+k)\end{bmatrix},\ k=0,\dots,L-1, \tag{10b}$$

$$\bar{y}(t+k)=e'_{\eta}\hat{X}_{1,T}\Omega^{\dagger}\begin{bmatrix}0_{1\times 2\eta}\\ I_{2\eta}\end{bmatrix}\bar{\chi}(t+k),\ k=0,\dots,L-1, \tag{10c}$$

$$\bar{\chi}(t) = \chi(t), \tag{10d}$$

$$\bar{u}(t+k) \in \mathbb{U},\quad k=0,\dots,L-1, \tag{10e}$$

$$\bar{y}(t+k) \in \mathbb{Y},\quad k=0,\dots,L-1, \tag{10f}$$

$$\bar{\chi}(t+L) = 0_{(m+p)\eta\times 1}, \tag{10g}$$

where $\bar{u}_{[0,L-1]} \in \mathbb{R}^{mL}$ is the input sequence to be optimized, while $\bar{\chi}(t+k) \in \mathbb{R}^{(m+p)\eta}$ and $\bar{y}(t+k) \in \mathbb{R}^p$ are the extended state and output predicted at time $t+k$, $k=0,\dots,L$, respectively.

Instead, with the predictor (9) the DDPC problem is:

$$\min_{\alpha} \sum_{k=0}^{L-1} \|\bar{y}(t+k)\|_Q^2 + \|\bar{u}(t+k)\|_R^2 \tag{11a}$$

$$\text{s.t.}\ \ \begin{bmatrix}\bar{u}_{[t-\eta,t+L-1]}\\ \bar{y}_{[t-\eta,t+L-1]}\end{bmatrix} = \begin{bmatrix}U_{0,L+\eta,T}\\ Y_{0,L+\eta,T}\end{bmatrix}\alpha, \tag{11b}$$

$$\begin{bmatrix}\bar{u}_{[t-\eta,t-1]}\\ \bar{y}_{[t-\eta,t-1]}\end{bmatrix} = \begin{bmatrix}u_{[t-\eta,t-1]}\\ y_{[t-\eta,t-1]}\end{bmatrix}, \tag{11c}$$

$$\bar{u}(t+k) \in \mathbb{U},\quad k=0,\dots,L-1, \tag{11d}$$

$$\bar{y}(t+k) \in \mathbb{Y},\quad k=0,\dots,L-1, \tag{11e}$$

$$\begin{bmatrix}\bar{u}_{[t+L-\eta,t+L-1]}\\ \bar{y}_{[t+L-\eta,t+L-1]}\end{bmatrix} = \begin{bmatrix}0_{m\eta\times 1}\\ 0_{p\eta\times 1}\end{bmatrix}, \tag{11f}$$

where $\alpha \in \mathbb{R}^{T-L-\eta-1}$ is the *only* parameter to be optimized, in turn, shaping the optimal sequence $\bar{u}^{\star}_{[0,L-1]}$.

The main difference between (10) and (11) clearly lies in the optimization variables, their dimensions, and the predictor equations (see (10b)-(10c) and (11b), respectively). Meanwhile, in both cases, the cost penalizes the distance of the predicted output sequence from zero and the control effort, respectively weighted via the matrices $Q \succeq 0$ and $R \succ 0$, while the predicted inputs and outputs are equally constrained according to (10e)-(10f) and (11d)-(11f). Because of the chosen extended state in (4), both problems share the same initial conditions and terminal constraints, with $\chi(t)$ in (10d) being constructed exactly as the right-hand-side of (11c) and $\bar{\chi}(t+L)$ in (10g) being equal to the left-hand-side of (11f).

3. NOISE HANDLING STRATEGIES

When working with noisy data, the differences between the predictors result into the necessity of developing distinct noise handling strategies. In this section, we review existing techniques to tackle noisy data, by highlighting their main advantages and pitfalls with a synoptic view.

3.1 Managing noise in (10)

If the problem in (10) is solved, one can cope with noise by directly operating on the data matrices $\hat{X}_{1,T}$ and Ω used to build the predictor, before the actual deployment of the controller. Specifically, noise handling can be done according to two main strategies.

Averaging. Since working with (7a) allows one to directly operate on data, the approach in Sassella et al. (2022a) grounds on a suitable *design of experiment*. Since the noise is assumed to be zero mean and additive, the approach relies on performing *repeated experiments* with the same input sequence, to average out its effect. The data matrices characterizing (7a), *i.e.,* $\hat{X}_{1,T}$ and Ω, are thus constructed via an *averaged* dataset $\tilde{D}_T = \{u(t), \tilde{y}(t)\}_{t=1}^T$, with

$$\tilde{y}(t) = \frac{1}{N}\sum_{i=1}^{N} y_i(t), \tag{12}$$

and $y_i(t) \in \mathbb{R}^p$ being the output at time t of the i-th experiment, with $i = 1,\dots,N$.

Dynamic Mode Decomposition (DMD). Rather than repeated experiments, Sassella et al. (2022b) proposes the use of *dynamic mode decomposition*, thus directly manipulating $\hat{X}_{1,T}$ and Ω constructed by relying on a single experiment. Initially, *singular value decomposition* (SVD) is performed to rewrite the initial data matrices as:

$$\hat{X}_{1,T} = \hat{U}\hat{\Sigma}\hat{V}^* \qquad \Omega = U_{\Omega}\Sigma_{\Omega}V_{\Omega}^*, \tag{13}$$

where $\hat{\Sigma}$ and Σ_{Ω} are diagonal matrices, whose non-zero elements are the singular values of $\hat{X}_{1,T}$ and Ω, respectively, while $\hat{U} \in \mathbb{C}^{\eta\times\eta}$, $U_{\Omega} \in \mathbb{C}^{(m+\eta)\times(m+\eta)}$ and $\hat{V}, V_{\Omega} \in \mathbb{C}^{T\times T}$ are unitary matrices, with $\hat{V}^*$ and V_{Ω}^* being the conjugate transposes of $\hat{V}$ and V_{Ω}. Small singular values can then be set to zero by means of a thresholding rule specified beforehand (*e.g.,* Sassella et al. (2022b)), to retain only those singular values that are relevant in the description of the dynamics of the system.

[2] Simple manipulations of costs and constraints lead to more general problems.

3.2 Managing noise in (11)

The use of (9) allows one to embed noise handling techniques directly into the formulated control problem. For instance Berberich et al. (2020); Dörfler et al. (2022) shift from the nominal DDPC to its *regularized* counterpart:

$$\min_{\alpha} \sum_{k=0}^{L-1} \|\bar{y}(t+k)\|_Q^2 + \|\bar{u}(t+k)\|_R^2 + \ell(\alpha) \tag{14a}$$

$$\text{s.t.} \begin{bmatrix}\bar{u}_{[t-\eta,t+L-1]}\\ \bar{y}_{[t-\eta,t+L-1]}\end{bmatrix} = \begin{bmatrix}U_{0,L+\eta,T}\\ Y_{0,L+\eta,T}\end{bmatrix}\alpha, \tag{14b}$$

$$\begin{bmatrix}\bar{u}_{[t-\eta,t-1]}\\ \bar{y}_{[t-\eta,t-1]}\end{bmatrix} = \begin{bmatrix}u_{[t-\eta,t-1]}\\ y_{[t-\eta,t-1]}\end{bmatrix}, \tag{14c}$$

$$\bar{u}(t+k) \in \mathbb{U}, \quad k=0, \tag{14d}$$

$$\bar{y}(t+k) \in \mathbb{Y}, \quad k=0, \tag{14e}$$

$$\begin{bmatrix}\bar{u}_{[t+L-\eta,t+L-1]}\\ \bar{y}_{[t+L-\eta,t+L-1]}\end{bmatrix} = \begin{bmatrix}0_{m\eta\times 1}\\ 0_{p\eta\times 1}\end{bmatrix}, \tag{14f}$$

where $\ell : \mathbb{R}^{T-L-\eta-1} \to \mathbb{R}$ varies depending on the chosen regularization technique, as outlined in the following.

L$_2$ regularization. L$_2$ (or Tikhonov) regularization , namely

$$\ell(\alpha) = \lambda_2 \|\alpha\|_2^2, \tag{15}$$

is used in several works, *e.g.,* Breschi et al. (2022). This choice uniformly shrinks the elements of α, in an attempt to make the predictions obtained thorough (11b) less sensitive to noise on the batch outputs. By leveraging on the fact that noise is additive, a slightly different handling scheme (still relying on Tikhonov regularization) is proposed in Berberich et al. (2020), with the evolution in (14b) modified as:

$$\begin{bmatrix}\bar{u}_{[t-\eta,t+L-1]}\\ \bar{y}_{[t-\eta,t+L-1]}+\varepsilon\end{bmatrix} = \begin{bmatrix}U_{0,L+\eta,T}\\ Y_{0,L+\eta,T}\end{bmatrix}\alpha, \tag{16}$$

and the cost of (14) featuring the extended regularization term

$$\ell(\alpha,\varepsilon) = \lambda_2\|\alpha\|_2^2 + \lambda_\varepsilon\|\varepsilon\|_2^2, \tag{17}$$

where $\varepsilon \geq 0$ is a slack variable to be shrunk to achieve a trade-off between retaining a meaningful predictor and limiting the impact of noise.

Weighted L$_2$ regularization. Inspired by subspace methods, (15) can be modified as in Dörfler et al. (2022) as follows:

$$\ell(\alpha) = \lambda_2\|(I-\Pi)\alpha\|_2^2, \quad \Pi = \begin{bmatrix}U_P' & Y_P' & U_F\end{bmatrix}' \tag{18}$$

where $U_P = [U_{0,L+\eta,T}]_{1:\eta}$, $Y_P = [Y_{0,L+\eta,T}]_{1:\eta}$ and $U_F = [U_{0,L+\eta,T}]_{\eta+1:L+\eta}$ and the weighting matrix $(I-\Pi)$ is an orthogonal projector onto the kernel of the initial conditions and future inputs. The objective of the regularizer is again to shrink the values of α to reduce the impact of noise based on *identification-related* insights.

L$_1$ regularization. A Lasso-like scheme is also proposed in Dörfler et al. (2022), with:

$$\ell(\alpha) = \lambda_1\|\alpha\|_1, \tag{19}$$

whose effect is that of shrinking the elements of α towards zero. As such, this approach induces structure selection, which has been proven to be equivalent to a low-rank approximation of the Hankel matrix characterizing (14b).

Elastic net. To combine the benefits deriving from both *identification-based* insights and *structure selection* effects, the weighted L$_2$ and L$_1$ regularizations are summed in Dörfler et al. (2022) into

$$\ell(\alpha) = \lambda_1\|\alpha\|_1 + \lambda_2\|(I-\Pi)\alpha\|_2^2, \tag{20}$$

where the hyper-parameters $\lambda_1, \lambda_2 > 0$ trade-off between the different shrinking effects induced by the two terms.

Regularization and state constraints. In Coulson et al. (2021), regularization is exploited within a chance-constrained algorithm, to robustly handle noise in the data and state constraints. To this end, the problem in (11) is augmented by the introduction of a slack, similar to (17). Apart from the additional slack, the cost remains equal to that in (14a), with α regularized as:

$$\ell(\alpha) = \lambda_\epsilon\|\alpha\|_r, \tag{21}$$

where $\lambda_\epsilon = L_{obj}\epsilon$, L_{obj} is the Lipschitz constant associated with the part of the cost depending on the system output, and ϵ is a tunable parameter quantifying the desired robustness level. Note that, $r = 2$ in (21), the cost of the design problem tackled in Coulson et al. (2021) is a variation to the L$_2$ regularized with slack previously discussed. Moreover, thanks to the use of a chance-constrained optimization scheme, the approach in Coulson et al. (2021) has probabilistic guarantees on constraints satisfaction.

3.3 Maximum likelihood estimation for DDPC

As an alternative to the previous noise-handling techniques and predictors, Yin et al. (2023) introduces a DDPC technique that does not require the tuning of any regularization penalty. The approach relies on the so-called *Signal Matrix Model* (SMM), *i.e.,* a maximum likelihood prediction model build from noisy offline data under the assumption of Gaussian distributed noise. Specifically, (14b)-(14c) are replaced with

$$\alpha^t = \mathcal{P}(\alpha^{t-1})\bar{y}_{[t-\eta,t-1]} + \mathcal{Q}(\alpha^{t-1})\bar{u}_{[t-\eta,t+L-1]}, \tag{22a}$$

$$\bar{y}_{[t,t+L-1]} = Y_{\eta,L,T}\alpha^t, \tag{22b}$$

where α^t is the updated approximator (at time t) of the maximum likelihood model, while the matrices $\mathcal{P}(\alpha^{t-1})$ and $\mathcal{Q}(\alpha^{t-1})$ depend on the available batch of data, the approximator α^{t-1} at the previous time step $t-1$ and an estimate of the noise variance computed offline.

3.4 A comparison between noise handling strategies

The advantages and downsides of each noise handling technique previously introduced should play a lion's share in the choice of the data-driven predictive control strategy, given the pervasive presence of noise in real-world applications.The main features of these techniques are summarized in Table 2. As it can be seen, the first major difference stemming from solving (14) rather than (10) lies in the elements the noise handling strategy operates on.

When using the predictor in (7), both the averaging strategy and the DMD-based approach manipulate the predictor itself, *not* modifying the cost function of the control problem. Similarly, the SMM-based approach do not require any hyper-parameter tuning, since the noise variance can be estimated offline from data as discussed in Yin et al. (2023). Analogously, the hyperparameter of the noise handling techniques considered in combination with (7) (*i.e.,* the number of experiments N and the thresholds $\hat{r}, r_\Omega$ indicating the singular values to retain) can be

Table 2. Main features of different noise handling strategies in data-driven predictive control, with Σ_v denoting the covariance of the measurement noise.

	Averaging	DMD	L_2/Weighted L_2	L_2+slack	L_1	Elastic net	SMM
Noise handling (what)	$\hat{X}_{1,T}, \Omega$	$\hat{X}_{1,T}, \Omega$	α	α	α	α	α^t
Noise handling (when)	Pre-proc.	Pre-proc.	Real time	Real time	Real time	Real time	Mixed
Hyper-parameters	N	$\hat{r}, r_\Omega$	λ_2	$\lambda_2, \lambda_\sigma$	λ_1	λ_1, λ_2	Σ_v
Limits on experiment design	Yes	No	No	No	No	No	No
Hyper-parameter calibration	Pre-proc.	Pre-proc.	Closed-loop	Closed-loop	Closed-loop	Closed-loop	Pre-proc.

calibrated *offline*. At the same time, by the law of large numbers, the number of experiments N should be selected as high as possible for the averaging strategy to properly filter out noise. Moreover, the need to perform several data collection campaigns by always using the same forcing input can severely limit the applicability of this approach in practice. Instead, the additional term characterizing the regularization strategies for (14) modifies the cost of the control problem, thus indirectly *changing the trade-off* between control objectives sought via the chosen weights Q and R. As such, calibrating the hyper-parameters characterizing this last family of noise handling strategies is quite delicate, as wrong choices of such parameters can drastically modify the control problem one ultimately solves. Meanwhile, up to now no approach has been proposed to tune the regularization penalties beforehand, with these parameters that have to be ultimately selected via *possibly unsafe closed-loop calibration tests*.

4. NUMERICAL EXAMPLES

To compare the performance of the summarized noise handling strategies presented in Section 2.4, we consider two simulation examples. To asses the robustness of the approaches to different input and noise realizations, the performance is evaluated by exploiting several datasets collected through Monte Carlo simulations. As for the selection of the hyper-parameters, the threshold $\hat{r}, r_\Omega$ needed by DMD are automatically chosen through the strategy outlined in Sassella et al. (2022b), while the regularization parameters are always selected via closed-loop grid search by constructing the predictor with a new set of data for different values of these hyper-parameters and picking the one that minimizes the following performance index:

$$\mathcal{J} = \sum_{t=1}^{T_v} \|y(t)\|_Q^2 + \|u(t)\|_R^2, \qquad (23)$$

over a simulation horizon of length T_v, always starting from the same initial condition. The same index is also used to quantitatively assess the performance of the final closed-loop tests. We stress that, to fairly assess the control performance, while the predictor is built with noisy data, the simulated closed-loop tests are noise-free.

4.1 A benchmark case study

Consider the following benchmark system taken from Sassella et al. (2022a):

$$x(k+1) = \begin{bmatrix} 0.7326 & -0.0861 \\ 0.1722 & 0.9909 \end{bmatrix} x(k) + \begin{bmatrix} 0.0609 \\ 0.0064 \end{bmatrix} u(k), \quad (24)$$

where the input is constrained so as $-2 \leq u(k) \leq 2$ and the state to be fully measured. For the sake of space, in our numerical comparison we do not include the averaging method, since its performance has already been compared on the same benchmark with that of the DMD-based approach in Sassella et al. (2022b).

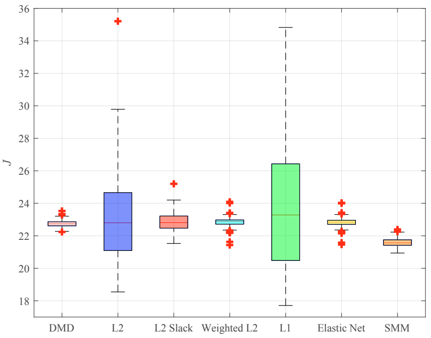

Fig. 1. Closed-loop tests with $T_v = 100$: $\mathcal{J}$ in (23) over 100 Monte-Carlo runs for the benchmark system.

Since the true order of the system in (24) is supposed to be unknown, the dimension of the extended state is set to $\eta = 10$. We solve a zero regulation problem over a prediction horizon $L = 2$, by imposing $R = 0.01$ and $Q = I$, while terminal constraints are not imposed given the length of the prediction horizon. The DDPC schemes and the relative noise handling strategies are all deployed by considering the same input/output set $\mathcal{D}_T$ of length $T = 1000$, with the input being a white noise sequence with uniform distribution within $[-5, 5]$ and the outputs corrupted by a zero mean, Gaussian distributed, white noise with standard deviation $0.024 \cdot I$.

As shown in Fig. 1, the use of DMD, the L_2 scheme with the additional slack variable [3], the weighted L_2 regularization elastic net lead to similar performance over the different realizations of the batch dataset, leading to costs that are closed to the model-based solution $\mathcal{J}^o = 22.21$. These approaches are slightly outperformed by the SMM-based approach. This result underlines the importance of exploiting identification-based insights when choosing the regularization strategy, while pointing out the effectiveness of DMD as a pre-processing tool. Conversely, plain L_2 and L_1 regularization tend to be rather sensitive to noise.

By considering 100 randomly chosen initial states, we further compare CPU time required with each noise handling approach, that are reported in Table 3. Clearly, regularization based approaches are generally computationally more demanding than DMD. Note that, achieving slightly better performance with the SMM-based approach implies an increase in the CPU time required to solve the optimization problem. Moreover, remarkably the use of the weighted L_2 regularization (and, thus, of elastic net) leads to a significant increase in CPU time, highlighting that

[3] $\lambda_\varepsilon = 10^4$ so as to shrink the slack variable as much as possible.

Table 3. CPU time [s] to compute $u(t)$ (benchmark system): mean ± standard deviation.

DMD	L_2	L_2+slack	L_1	Weighted L_2	Elastic net	SMM
0.16±0.02	0.23±0.03	0.27±0.04	0.47±0.09	1.88±0.13	2.62±0.48	0.31±0.0.04

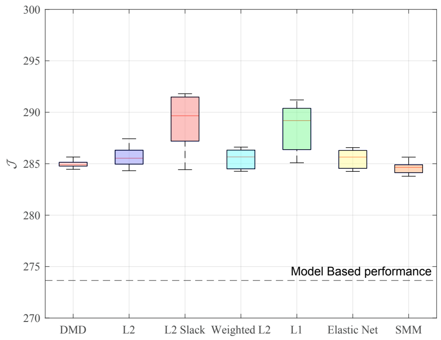

Fig. 2. Closed-loop tests with $T_v = 30$: $\mathcal{J}$ in (23) over 10 Monte-Carlo runs for the random system.

the price one has to pay to include identification-oriented insights in the regularization term is an increase in the complexity of the optimal problem to be solved.

4.2 Monte-Carlo simulation on a random system

We then consider an asymptotically stable random system with *unknown* order $n = 10$, with $m = 5$ inputs. As in the first example, we consider input constraint $|u| \leq 2$, while we impose $\eta = 22$. We solve the DDPC problems with prediction horizon $L = 3$, by imposing $R = 0.01$ and $Q = I$. In particular, we evaluate the effect of the noise handling strategies over 10 different datasets of length 500, collected applying input uniformly distributed in the interval $[-2, 2]$, where the output is corrupted by an additive, Gaussian distributed white noise, yielding SNR= 35 dB. Apart from the predictive control schemes with L_2 and slacks and L_1 regularization, which show higher performance variability, Fig. 2 clearly displays that most approaches lead to comparable performance. Indeed, the difference between the oracle cost and the median loss of all approaches is almost the same. Nonetheless, with the increase in the system dimension, the variance of the performance index $\mathcal{J}$ also increases when using L_2-based regularization strategies.

5. CONCLUSIONS

In this work, we review and compare existing techniques to handle noisy measurements in data-driven predictive control. Our analysis shows that *DMD can perform comparably to regularization-based strategies*, when the hyper-parameters are properly tuned and enough data are available. At the same time, it reveals the *high sensitivity of regularization to the choice of the penalty coefficients*. Meanwhile, our tests highlight the robustness of the strategy relying on a maximum likelihood predictive model since our setting complies with its main assumptions. Our test further highlights the additional computational effort required by identification-based regularization approaches, which can be problematic when coping with fast sampling systems. Future works will investigate strategies for the reduction of the sensitivity of trajectory-based approaches to the regularization weights, and methodologies for a safe tuning of these hyper-parameters.

REFERENCES

Berberich, J., Köhler, J., Müller, M.A., and Allgöwer, F. (2020). Data-driven model predictive control with stability and robustness guarantees. *IEEE Transactions on Automatic Control*, 66(4), 1702–1717.

Breschi, V., Sassella, A., and Formentin, S. (2022). On the design of regularized explicit predictive controllers from input-output data. *IEEE Transactions on Automatic Control*.

Coulson, J., Lygeros, J., and Dörfler, F. (2019). Data-enabled predictive control: In the shallows of the deepc. In *2019 18th European Control Conference (ECC)*, 307–312. IEEE.

Coulson, J., Lygeros, J., and Dorfler, F. (2021). Distributionally robust chance constrained data-enabled predictive control. *IEEE Transactions on Automatic Control*.

De Persis, C. and Tesi, P. (2020). Formulas for data-driven control: Stabilization, optimality, and robustness. *IEEE Transactions on Automatic Control*, 65(3), 909–924.

Dörfler, F., Coulson, J., and Markovsky, I. (2022). Bridging direct & indirect data-driven control formulations via regularizations and relaxations. *IEEE Transactions on Automatic Control*.

Forbes, M.G., Patwardhan, R.S., Hamadah, H., and Gopaluni, R.B. (2015). Model predictive control in industry: Challenges and opportunities. *9th IFAC Symposium on Advanced Control of Chemical Processes (ADCHEM)*, 48(8), 531–538.

Hastie, T., Tibshirani, R., Friedman, J.H., and Friedman, J.H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

Huang, L., Zhen, J., Lygeros, J., and Dörfler, F. (2021). Quadratic regularization of data-enabled predictive control: Theory and application to power converter experiments. *IFAC-PapersOnLine*, 54(7), 192–197.

Kutz, J.N., Brunton, S.L., Brunton, B.W., and Proctor, J.L. (2016). *Dynamic mode decomposition: data-driven modeling of complex systems*. SIAM.

Sassella, A., Breschi, V., and Formentin, S. (2022a). Data-driven design of explicit predictive controllers. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, 2821–2826.

Sassella, A., Breschi, V., and Formentin, S. (2022b). Noise handling in data-driven predictive control: A strategy based on dynamic mode decomposition. In R. Firoozi, N. Mehr, E. Yel, R. Antonova, J. Bohg, M. Schwager, and M. Kochenderfer (eds.), *Proceedings of The 4th Annual Learning for Dynamics and Control Conference*, volume 168 of *Proceedings of Machine Learning Research*, 74–85. PMLR.

Willems, J., Rapisarda, P., Markovsky, I., and De Moor, B. (2005). A note on persistency of excitation. *Systems & Control Letters*, 54(4), 325–329.

Yin, M., Iannelli, A., and Smith, R.S. (2023). Maximum likelihood estimation in data-driven modeling and control. *IEEE Transactions on Automatic Control*, 68(1), 317–328.