The role of regularization in data-driven predictive control

Valentina Breschi\textsuperscript{a}, Alessandro Chiuso\textsuperscript{b}, Simone Formentin\textsuperscript{a}

\textsuperscript{a} Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, P.za L. Da Vinci, 32, 20133 Milano, Italy.

\textsuperscript{b} Department of Information Engineering, University of Padova, Via Gradenigo 6/b, 35131 Padova, Italy.

Abstract

Data-driven predictive control (DDPC) has been recently proposed as an effective alternative to traditional model-predictive control (MPC) for its unique features of being time-efficient and unbiased with respect to the oracle solution. Nonetheless, it has also been observed in many examples that the noise on the output data may strongly jeopardize the final closed-loop performance, since it affects both the data-based system representation and the control update computed from the online measurements. Recent studies have empirically shown that regularization is potentially a successful tool to counteract the effect of noise. In this paper, the link between the stochastic data matrices and the predictive control problem is studied in detail. Then, leveraging on these results, we delve deep into the role of regularization for data-driven control, discussing when and how it should be applied. A benchmark numerical case study illustrates how control design is simplified when a method inspired to the proposed perspective, called $\gamma$-DDPC, is adopted. Specifically, when regularization can be avoided, no closed-loop experiments are required to tune the regularization weights via cross-validation.

Key words: data-based control, control of constrained systems, regularization, identification for control

1 Introduction

Direct data-driven control refers to the science of learning feedback controllers from data, without first undertaking a full modeling study of the plant to control [16]. Such a direct mapping of data onto the control action is indeed advisable in real-world problems, as modeling usually takes about 75\% of the time devoted to a control project [17], and accurate modeling for control requires significant time and several (costly) technical expertises, e.g., in the process domain and in the statistical tools for system identification. Additionally, accurate modeling may go well beyond what is strictly necessary for control purposes only, since often times rather limited knowledge of the system dynamics may be required to achieve the desired control objectives [21].

Early attempts in this direction date back to 1942, with the first studies by Ziegler and Nichols about PID autotuning [30]. More sophisticated, optimization based, approaches have been derived since then for fixed-order controller tuning, leading to a portfolio of techniques suitable for different problem formulations, see, e.g., [14,6,15,22]. However, it is only recently that, with the availability of large datasets and unparalleled computing power, such a paradigm shift in control design could be extended to more complex control architectures. For instance in the deterministic setting, by relying on the so-called “fundamental lemma” [28], model equations can be replaced by suitable data-based constraints\footnote{Such constraints can also be seen as an implicit, nonparametric, mapping of the input/output relationships. According to this interpretation, some researchers legitimately prefer to denote the strategies described herein as “indirect”. For this reason, we will simply talk about \textit{data-driven predictive control} from now on.} in the formulation of a Model Predictive Control (MPC) schemes, as discussed e.g. in [11] or [5]. Such a data-based framework may lead to different performance than traditional model-based MPC as it shows unique features. For instance, the sub-optimality gap measuring the control performance with respect to the optimal model-based control (namely, model-based control using the real model of the system) vanishes with the size of the dataset. Moreover, model-free predictive control may indirectly address the bias/variance trade-off in a more efficient manner; for instance it will not incur in the asymptotic bias induced by inaccurate modeling.
when complexity constraints are imposed on the model structure, as discussed in [23]. Nonetheless, as currently addressed, the replacement of the description of the system dynamics with data matrix equalities (or inequalities) is strictly speaking valid only in case of purely deterministic systems. It follows that the ubiquitous presence of “noise” inevitably leads to some approximations, which in turn may deteriorate the closed-loop performance.

The problem of noise and unmeasurable disturbance inputs in direct data-driven predictive control has been clear from the start, so different solutions have been proposed so far. In [5], the authors prove practical exponential stability of the closed loop in the presence of bounded additive output noise. The key ingredients are two: (i) some bounded slack variables to account for both the noisy online measurements and for the noisy data used for prediction, and (ii) some suitable regularization terms. In [2], a slightly different scheme is used, which computes the data-driven reachable set based on a matrix zonotope recursion starting from the measured output. For this scheme, the authors show they can guarantee robust constraint satisfaction, again in case of bounded process and measurement noises. The case of stochastic (white) measurement noise is addressed in [29], where a maximum likelihood framework is proposed to estimate the data-based constraints aimed to replace the model equations in the MPC formulation. In [29], an iterative two-stage approach is considered, where at each iteration first a model encoded by a data matrix constraint must be identified and then the online predictive control is computed. An approach to handle stochastic noise in the direct framework proposed in [11] and [5] is instead given in the recent paper [12]. In the above contribution, the authors exploit regularization as the key tool to handle the presence of noise in the output measurements, and empirically discuss the performance of different regularization schemes. We should remark that the work in [12] concerns only the open-loop predictive control problem, and not the receding horizon closed-loop implementation.

In this paper, we delve deep into the structure of data matrices in a stochastic framework, where both measurement as well as process noise are allowed, thus accounting for unmeasurable stochastic inputs affecting the dynamical behaviour of the controlled system. The stochastic nature of measured data is fully exploited to derive the data-driven formulation of the optimal constrained control problem. Finally, we provide a full receding-horizon implementation. Specifically, the novel contributions of our paper are:

- we derive the link between the optimal receding horizon predictive control problem and the stochastic Hankel data matrices;
- we show that the tunable parameter vector describing the system dynamics in its data-based form can be suitably decomposed into three terms with different roles. This decomposition allows to show that the free optimization variables live in a low dimensional subspace, thus also reducing the complexity of the MPC optimization problem. We will call the new procedure $\gamma$-DDPC from now on;
- we provide the link between $\gamma$-DDPC and other approaches from the literature which exploit regularization, showing how regularization parameters in those scheme can be optimally computed a-priori, without the need of specific knowledge of the problem at hand or on the dataset;
- finally, also by means of benchmark numerical examples, we discuss the role of regularization in light of the above results and conclude that additional regularization is not needed. This yields the significant advantage that no closed-loop experiment is needed to tune the regularization weights through cross-validation.

The remainder of the paper is as follows. In Section 2, we formally define the control problem of interest and its data-driven counterpart. Section 3 gives a novel insight into the data-driven system description employed in the computation of the optimal control action using subspace techniques. In light of the outcomes of the preceding analysis, Section 4 introduces the $\gamma$-DDPC variation and discusses the role of regularization in the data-driven control framework. The benchmark numerical example of Section 5 illustrates the effectiveness of the $\gamma$-DDPC perspective in designing a satisfactory control action with little to no effort. The paper is ended by some concluding remarks.

**Notation.** Matrices will be denoted with capitals (e.g. $A$), column vectors will be denoted with lowercase letters (e.g. $a$). The transpose of $A$ will be denoted with $A^\top$; the notation $A^\dagger$ will denote the Moore-Penrose pseudoinverse of $A$. Given deterministic (vector) sequences $a(t)$, $b(t)$ the notation $a(t) = O(b(t))$ means that there exist $M$ and $c < \infty$ such that, for all $t > M$,

$$\|a(t)\| \leq c\|b(t)\|. \quad (1)$$

Similarly we say that $a(t) = o(b(t))$ if

$$\lim_{t \to \infty} \frac{\|a(t)\|}{\|b(t)\|} = 0,$$

or, equivalently, that for all $\epsilon > 0$, there exists $M < \infty$ such that, for all $t > M$,

$$\|a(t)\| \leq \epsilon\|b(t)\|. \quad (2)$$

Probabilistic versions of $O(\cdot)$ and $o(\cdot)$ (i.e., with conditions (1) and (2) holding in probability) will be denoted by $O_P(\cdot)$ and $o_P(\cdot)$, see e.g., [25]. We use the symbol $\tilde{=} \equiv$ to denote equality up to $o_P(1/\sqrt{N})$ and $\overset{\hat{\scriptscriptstyle{\sim}}}{=} \overset{\hat{\scriptscriptstyle{\sim}}}{\equiv}$ to indicate
equality up to $O_p(1/\sqrt{N})$. $\Pi_A[B]$ denotes the orthogonal projection of the (rows of the) matrix $B$ on the row span of the matrix $A$, i.e.,

$$\Pi_A[B] = BA^\top (AA^\top)^{-1}A.$$ 

Similarly $\Pi_{A,C}[B]$ indicates the projection of $B$ onto the row span of $A$ and $C$. Finally, given a signal $w(k) \in \mathbb{R}^s$, we define the associated Hankel matrix $W_{[t_0,t_1],N} \in \mathbb{R}^{s(t_1-t_0+1) \times N}$ as:

$$W_{[t_0,t_1],N} := \frac{1}{\sqrt{N}} \begin{bmatrix} w(t_0) & w(t_0+1) & \cdots & w(t_0+N-1) \\
 w(t_0+1) & w(t_0+2) & \cdots & w(t_0+N) \\
 \vdots & \vdots & \ddots & \vdots \\
 w(t_1) & w(t_1+1) & \cdots & w(t_1+N-1) \end{bmatrix},$$

(3)

while the shorthand $W_{t_0} := W_{[t_0,t_0],N}$ is used to denote the Hankel containing a single row, namely:

$$W_{t_0} := \frac{1}{\sqrt{N}} \begin{bmatrix} w(t_0) & w(t_0+1) & \cdots & w(t_0+N-1) \end{bmatrix}. \tag{4}$$

2 Setting and goal

Consider an unknown discrete-time, linear time-invariant (LTI) stochastic plant, whose behaviour can always be described by the so-called innovation-form equations

$$\begin{align*}
& x(t+1) = Ax(t) + Bu(t) + Ke(t) \\
& y(t) = Cx(t) + Du(t) + e(t),
\end{align*} \tag{5}$$

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$ and $e(t) \in \mathbb{R}^p$ are the state, input and innovation process respectively, while $y(t) \in \mathbb{R}^p$ is the corresponding output signal. Without loss of generality we shall assume that (5) is minimal (i.e., reachable and observable).

Given a reference signal $y_r(t), t \in \mathbb{Z}$, and a control horizon $T$, the receding horizon predictive control problem can be framed as follows:

$$\begin{align*}
& \text{minimize} \quad \frac{1}{2} \sum_{k=t}^{t+T-1} \mathbb{E} \left[ \|y(k) - y_r(k)\|_Q^2 \right] + \|u(k)\|_R^2 \\
& \text{subject to} \quad x(k+1) = Ax(k) + Bu(k) + Ke(k), \quad k \in [t,t+T), \\
& \quad y(k) = Cx(k) + Du(k) + e(k), \quad k \in [t,t+T), \\
& \quad x(t) = x_{\text{init}}, \\
& \quad u(k) \in \mathcal{U}, \quad \mathbb{E}[y(k)] \in \mathcal{Y}, \quad k \in [t,t+T), \\
& \quad y_r(k) = y_r(t), \quad k \in [t,t+T). \tag{6a} \tag{6b} \tag{6c} \tag{6d} \tag{6e}
\end{align*}$$

where $x_{\text{init}}$ is the state at time $t$, $e(k)$ is a zero mean noise with variance $\text{Var}\{e(k)\}$, the sets $\mathcal{U}$, $\mathcal{Y}$ denote inputs and output constraints, and the expectation $\mathbb{E}[]$ is taken w.r.t. the future noise sequence $e(k), k \in [t,t+T)$, and conditionally on the initial state $x_{\text{init}}$ and the future input trajectory $u_f := \{ u(k), k \in [t,t+T) \}$. The tunable symmetric weights $Q \in \mathbb{R}^{p \times p}$ and $R \in \mathbb{R}^{m \times m}$, with $Q \succeq 0$ and $R \succ 0$, have to be selected to trade-off between tracking performance and the required control effort. For simplicity, without loss of generality, from now on we will consider a constant reference along the prediction horizon, i.e.,

$$y_r(k) = y_r(t), \quad k \in [t,t+T).$$

Our goal is to solve problem (6) when the systems matrices $A, B, C, D, K$ are not known and only a sequence of input output data $\mathcal{D}_{\text{Data}} = \{ u(j), y(j) \}_{j=1}^{N_{\text{Data}}}$ collected in open loop from system (5) is available.

2.1 Features of the predictive control problem

We now elaborate on the optimization problem (6) and make two important observations:

1. Problem (6) can be equivalently formulated only in terms of the so called “deterministic” part of the stochastic system (5), i.e., the one depending only on the control input and the initial state, but not on the noise $e(k)$.

2. The initial state $x_{\text{init}}$ at time $t$ does not have to be available. Indeed, it can be accounted for with arbitrary accuracy based on a sufficiently long window of past input-output observations.

To show that the first point holds, it is useful to rewrite the control problem (6) exploiting the decomposition of second order moments as the sum of squared means plus variance, i.e.,

$$\mathbb{E} \left[ \|y(k) - y_r(k)\|_Q^2 \right] = \mathbb{E} \left[ \|y(k)\|_Q^2 \right] + \mathbb{E} \left[ \|y(k) - \mathbb{E}[y(k)]\|_Q^2 \right]$$

independent of $u(k)$

Since the variance term $\mathbb{E}[[\|y(k) - \mathbb{E}[y(k)]\|_Q^2]$ is independent of the input signal $u(k), k \in [t,t+T)$, only the conditional (given $x_{\text{init}}$ and $u_f$) mean value of the output, namely $y^d(k) := \mathbb{E}[y(k)]$ affects the optimization problem. Denoting with $x^d(k)$ the conditional mean of $x(k), i.e. x^d(k) := \mathbb{E}[x(k)]$, it is straightforward to see that the optimal control problem (6) can be equivalently recast\footnote{Extension to data collected in closed-loop is possible. Yet, for the sake of exposition, its treatment is deferred to future publications.}
and, for any $\rho > 0$, it holds that
\begin{equation}
\begin{aligned}
&\text{minimize } u(k), k \in [t, t+T] \\
&\frac{1}{2} \sum_{k=t}^{t+T-1} \| y^d(k) - y_r(k) \|^2_Q + \| u(k) \|^2_R
\end{aligned}
\end{equation}
\tag{7a}

\(\text{s.t.} \quad x^d(k+1) = Ax^d(k) + Bu(k), \quad k \in [t, t+T), \) 
\tag{7b}

\(y^d(k) = Cx^d(k) + Du(k), \quad k \in [t, t+T), \) 
\tag{7c}

\(x^d(t) = x_{\text{init}}, \) 
\tag{7d}

\(u(k) \in \mathcal{U}, \quad y^d(k) \in \mathcal{Y}, \quad k \in [t, t+T). \) 
\tag{7e}

Even though only the “deterministic” part of the system influences the optimal control problem, it is important to stress that measured data are indeed affected by noise. This should be accounted for when exploiting measured data $\mathcal{D}_{\text{data}}$ to solve (7).

As it concerns the second observation, to prove its validity we exploit the fact that (5) can be written in innovation (or “whitening” [9]) form. Accordingly, it holds that
\begin{equation}
\begin{aligned}
&\{ x(k + 1) = (A - KC)x(k) + Bu(k) + Ky(k), \\
&e(k) = y(k) - Cz(k) + Du(k), \}
\end{aligned}
\tag{8}

and, for any $\rho > 0, \rho \in \mathbb{Z},$
\begin{equation}
x(t) = (A - KC)^p x(t - p) + \sum_{p=1}^{\rho} [\Phi_p u(t-p) + \Psi_p y(t-p)],
\end{equation}

where $\Phi_p = (A - KC)^{p-1} B$ and $\Psi_p = (A - KC)^{p-1} K$.

By denoting with $\lambda_{\text{max}}$ the eigenvalues of $A - KC$ of largest absolute value, under the (mild) assumption that the matrix $A - KC$ is strictly stable, i.e., $|\lambda_{\text{max}}| < 1$, we have that:
\begin{equation}
x(t) = C \begin{bmatrix} u^r_t \\ y^r_t \end{bmatrix} + O(|\lambda_{\text{max}}|^p),
\end{equation}

where the last term vanishes to zero exponentially, $C$ stacks the (reversed) controllability matrices $C_u$ and $C_y$, i.e.,
\begin{equation}
C = \begin{bmatrix} C_u & C_y \end{bmatrix} = \begin{bmatrix} \Phi_p & \cdots & \Phi_1 & \Psi_p & \cdots & \Psi_2 \end{bmatrix},
\end{equation}

and
\begin{equation}
u^r_t := \begin{bmatrix} u(t-p) \\ u(t-2) \\ u(t-1) \end{bmatrix}, \quad y^r_t := \begin{bmatrix} y(t-p) \\ y(t-2) \\ y(t-1) \end{bmatrix}.
\end{equation}

The relation in (10) thus guarantees that, up to $O(|\lambda_{\text{max}}|^p)$ terms, the initial state can be uniquely reconstructed with a finite window of past data.

**Remark 1 (State/data relation)** In the so-called “deterministic case”, i.e., when there is no process/measurement noise in (5), the state at time $t$ is a (deterministic) function of a finite past window of input-output data. As such, $\mathcal{C}_{\text{det}}$ such that
\begin{equation}
x(t) = \mathcal{C}_{\text{det}} \begin{bmatrix} u^r_t \\ y^r_t \end{bmatrix},
\end{equation}

provided $\rho \geq n$. This is a trivial consequence of (9) and of the observability of the system.

**Remark 2 (Choice of $\rho$ - part 1)** In (subspace) system identification, see e.g., [3,8,7,10] the quantity $\rho$, known also as the “past horizon”, has to be determined from measured data trading off bias and variance. Indeed, $\rho$ should be large, so that the quantity $O(|\lambda_{\text{max}}|^p)$ can be neglected, but a large $\rho$ ultimately requires estimating larger sample covariances. A simple and effective way of determining $\rho$ in a data-driven fashion is by using Akaike’s criterion (e.g., FPE) [1], with the latter choice also guaranteeing that $O(|\lambda_{\text{max}}|^p) = o(1/\sqrt{N_{\text{data}}})$. This is in contrast with common practice in the literature of DDPC where the length $\rho$ of the past horizon (i.e., the number of block rows in $Y_P$ and $U_P$) is not linked to the eigenvalues of $(A - KC)$, but rather it is generally chosen based on (e.g., an upper bound of) the “order” $n$ of the deterministic model.

3 DDPC formulation via subspace methods

In this Section, we exploit ideas from subspace identification to recast Problem (7) in terms of observed input output data $\mathcal{D}_{\text{data}}$.

3.1 Preliminaries

Let us firstly define
\begin{equation}
z(k) := \begin{bmatrix} u(k) \\ y(k) \end{bmatrix},
\end{equation}

to introduce the shorthands for the “past” Hankel matrices, namely
\begin{equation}
U_P := U_{[0, \rho-1], N}, \quad Y_P := Y_{[0, \rho-1], N}, \quad Z_P := Z_{[0, \rho-1], N}
\end{equation}
and the “future” ones, i.e.,
\begin{equation}
U_F := U_{[\rho, \rho+T-1], N}, \quad Y_F := Y_{[\rho, \rho+T-1], N}, \quad E_F := E_{[\rho, \rho+T-1], N}
\end{equation}

Note that, once the lengths of both the “past” $\rho$ and “future” $T$ are fixed, the number of columns $N$ of the Hankel data matrices is chosen in such a way that all the
available data are exploited, namely \( N := N_{\text{data}} - T - \rho \).

Let us further introduce the extended observability matrix \( \Gamma \in \mathbb{R}^{pT \times n} \) associated with the system in (5), namely

\[
\Gamma = \begin{bmatrix}
    C \\
    CA \\
    CA^2 \\
    \vdots \\
    CA^{T-1}
\end{bmatrix},
\]

and the Toeplitz matrices \( \mathcal{H}_d \in \mathbb{R}^{pT \times mT} \) and \( \mathcal{H}_s \in \mathbb{R}^{pT \times pT} \) formed with its Markov parameters, i.e.,

\[
\mathcal{H}_d = \begin{bmatrix}
    D & 0 & 0 & \ldots & 0 \\
    CB & D & 0 & \ldots & 0 \\
    \vdots & \vdots & \vdots & \ddots & \vdots \\
    C A^{T-2} B & C A^{T-3} B & C A^{T-4} B & \ldots & D
\end{bmatrix}, \quad (15a)
\]

\[
\mathcal{H}_s = \begin{bmatrix}
    I & 0 & 0 & \ldots & 0 \\
    C K & I & 0 & \ldots & 0 \\
    \vdots & \vdots & \vdots & \ddots & \vdots \\
    C A^{T-2} K & C A^{T-3} K & C A^{T-4} K & \ldots & I
\end{bmatrix}. \quad (15b)
\]

The Hankel matrix of future outputs \( Y_F \) thus satisfies the following:

\[
Y_F = \Gamma X_\rho + \mathcal{H}_d U_F + \mathcal{H}_s E_F,
\]

which is the equation often considered as a starting point in subspace identification [26,9]. Note that, based on (10) and provided \( \rho \) is chosen in a data-driven fashion as discussed in Remark 2, \( X_\rho \) can be written as

\[
X_\rho = C_u U_F + C_y Y_F + (A - K C)^\rho X_0 \overset{\cdot}{=} C Z_P, \quad (17)
\]

where \( Z_P = \begin{bmatrix} U_F^T & Y_F^T \end{bmatrix}^T \) and \( \cdot \) denotes equality up to \( O_p(1/\sqrt{N}) \) terms. We can now characterize the future noise \( E_F \) according to the following.

**Lemma 1 (Projection of noise)** For any fixed \( \rho \) in (11), it holds that

\[
\Pi_{Z_P, U_F}(E_F) = Y \begin{bmatrix}
    Z_P \\
    U_F
\end{bmatrix}, \quad (18)
\]

where \( \| Y \| = O_p\left( \frac{1}{\sqrt{N}} \right) \) and \( Z_P, U_F \) are defined as in (12) and (13).

**Proof** By definition,

\[
\Pi_{Z_P, U_F}(E_F) = EF \begin{bmatrix} Z_P^T & U_F^T \end{bmatrix} \begin{bmatrix}
    Z_P & Z_P^T U_F \\
    U_F & U_F^T
\end{bmatrix}^{-1} \begin{bmatrix}
    Z_P^T \\
    U_F^T
\end{bmatrix}.
\]

so that

\[
Y = EF \begin{bmatrix} Z_P^T & U_F^T \end{bmatrix} \begin{bmatrix}
    Z_P & Z_P^T U_F \\
    U_F & U_F^T
\end{bmatrix}^{-1}. \]

It is sufficient to observe that the term on the left-hand side \( \hat{\Sigma}_{eFz} \) and \( \hat{\Sigma}_{eFup} \) are sample cross-covariances between future innovations and past data \( (z_P) \) or future inputs \( u_F \), and thus converge to zero in probability with rate \( 1/\sqrt{N} \), whereas the rightmost term converged to the input-output covariance matrix, which is bounded away from zero.

This Lemma further allows us to characterize the future outputs \( Y_F \) as follows.

**Lemma 2 (Projection of the output)** The projection \( \hat{Y}_F := \Pi_{Z_P, U_F}(Y_F) \) satisfies

\[
\hat{Y}_F = \Gamma \hat{X}_\rho + \mathcal{H}_d U_F + \mathcal{H}_s \Pi_{Z_P, U_F}(E_F)
\]

\[
\hat{Y}_F = \Gamma \hat{X}_\rho + \mathcal{H}_d U_F + \mathcal{H}_s C Z_P, \quad (19)
\]

where \( \hat{X}_\rho := \Pi_{Z_P, U_F}(X_\rho) = C Z_P \).

**Proof** The proof straightforwardly follows from the observation that the projection is a linear operator, by exploiting Lemma 1 on the projection of the noise term and Equation (17) on the approximation of the state using a finite set of past data.

Given the projected initial condition \( \hat{X}_\rho \) and the input \( U_F \), Lemma 2 establishes that the projected output \( \hat{Y}_F \) equals the evolution of the deterministic part of the system (5), up to \( O_p(1/\sqrt{N}) \) terms. This result is formalized in the following Theorem.

**Theorem 1 (Output/data relation)** Given any \( \alpha \in \mathbb{R}^N \), the vector \( \hat{y}_f^d := \hat{Y}_F \alpha \) satisfies the relation

\[
\hat{y}_f^d = \hat{\Gamma} \hat{x}_f^d + \mathcal{H}_d u_f + O_p\left( \frac{1}{\sqrt{N}} \right)
\]

\[
\hat{y}_f^d = \hat{\Gamma} \hat{x}_f^d + \mathcal{H}_d u_f, \quad (20)
\]
The following lemma formalizes this result.

\[ Z = \sum_{k} \alpha_k z_{init} \]

possible initial condition and sequence of control inputs that the matrices

\[ \hat{y}_f := \hat{Y}_F\alpha \]

and \( z_{init} := Z\alpha \).

**Proof** The proof is an immediate consequence of Lemma 2. In fact, defining \( \hat{y}_f := \hat{Y}_F\alpha \) and using Equation (19), we have that

\[ \hat{y}_f := \hat{Y}_F\alpha = \Gamma C Z\alpha + \mathcal{H}_d U_F\alpha. \]

The result in Theorem 1 should be read as follows. If the sequence of past input-output \( u(k) \) and \( y(k) \) for \( k \in [t - \rho, t - 1] \) equals \( z_{init} \) and the future inputs \( u(k) \) in the time window \( k \in [t, t + T - 1] \) (see (21b)) are given by \( u_f \), the corresponding "deterministic" output, i.e.,

\[ y_f := \begin{bmatrix} y^d(t) \\ y^d(t+1) \\ \vdots \\ y^d(t+T-1) \end{bmatrix}, \]

is a linear transformation through \( \alpha \) of the projected future outputs \( \hat{Y}_F \), up to \( O_P(1/\sqrt{N}) \) terms.

### 3.2 Towards DDPC

For every pair of initial conditions and future inputs that can be written as linear combinations of \( Z \) and \( U_F \) (see (12) and (13)), Theorem 1 shows that one can compute the deterministic output of (5) (up to \( O_P(1/\sqrt{N}) \) terms) from a finite set input-output data only, without knowing the true system (5). Under the additional assumption that the training input \( u(t) \) has a full rank spectral density matrix and the innovation process has positive definite variance \( \text{Var}\{e(t)\} > 0 \) \footnote{Since the our purpose is not to discuss the weakest conditions under which the results of Theorem 1 can be generalized, here we make a sufficient assumption that is general enough for being widely applicable in practice.}, we can guarantee that the matrices \( Z \) and \( U_F \) have full rank, so that any possible initial condition and sequence of control inputs can be generated by linear combination of their columns. The following lemma formalizes this result.

**Lemma 3 (Persistency of excitation)** If the input process has full rank spectral density that is bounded away from zero and \( \text{Var}\{e(t)\} > 0 \), then for any choice of \( \rho \) and \( T \) and provided \( N > (m + p)(\rho + T) \), the block Hankel matrix

\[ Z_{data} := \begin{bmatrix} Z_F \\ U_F \\ Y_F \end{bmatrix} \in \mathbb{R}^{(m+p)(\rho+T)\times N} \]

has full rank almost surely.

**Proof** The proof is a direct consequence of the fact that, under the stated assumptions, the joint spectral density matrix of the input-output process \( z(t) := [u^T(t) \ y^T(t)]^T \) does not vanish on the unit circle and, therefore, the intersection between the (joint) past and input spaces contains only the zero random variable (see e.g. [20]). Thus, the Hankel matrix formed with input output trajectories has full rank almost surely.

Under the latter, the result in Theorem 1 can be generalized to all initial conditions and future inputs, as stated in the main result of this Section.

**Theorem 2 (Output/data relation - generalized)** Under the assumptions in Lemma 3, given any (past) joint input and output trajectory

\[ z_{init} := \begin{bmatrix} z(t-\rho) \\ \vdots \\ z(t-2) \\ z(t-1) \end{bmatrix}, \]

and any choice of the future control input

\[ u_f := \begin{bmatrix} u(t) \\ u(t+1) \\ \vdots \\ u(t+T-1) \end{bmatrix}, \]

the corresponding "deterministic" output

\[ y_f := \begin{bmatrix} y^d(t) \\ y^d(t+1) \\ \vdots \\ y^d(t+T-1) \end{bmatrix} \]

satisfies:

\[ y_f = \hat{Y}_F\alpha^* + O_P(1/\sqrt{N}) = \hat{Y}_F\alpha^* \]
where \( \alpha^* \) is the minimum-norm solution of the system of linear equations:

\[
\begin{bmatrix}
  z_{\text{init}} \\
  u_f
\end{bmatrix} = \begin{bmatrix}
  Z_P \\
  U_F
\end{bmatrix} \alpha.
\]  

(26)

where \( \hat{Y}_F := \Pi_{Z_P, U_F}(Y_F) \).

**Proof** Under the assumption of Lemma 3, the matrix \( Z_{\text{data}} \) has full rank and, therefore, \( \forall z_{\text{init}} \) and \( u_f \), there exists \( \alpha \) such that

\[
\begin{bmatrix}
  z_{\text{init}} \\
  u_f
\end{bmatrix} = \begin{bmatrix}
  Z_P \\
  U_F
\end{bmatrix} \alpha.
\]  

(27)

Thus, exploiting Theorem 1, the corresponding deterministic output satisfies

\[
y_f^d = \hat{Y}_F \alpha + \mathcal{O}_P(1/\sqrt{N}) \triangleq \hat{Y}_F \alpha.
\]

This is true for all possible solutions of (27), and in particular it holds for its minimum-norm solution \( \alpha^* \).

**Remark 3 (The case of deterministic systems)**

The reader may observe that, when \( e(t) = 0 \) (that is the system is actually deterministic), Lemma 3 does not hold. Indeed, for for \( \rho > n \), it is well known (see, e.g., [24]) that the Hankel matrix \( Z_{\text{data}} \) in (22) and \( Z_P \in \mathbb{R}^{(m+p)\times N} \) in (26) have rank equal to

\[
\text{rank}(Z_{\text{data}}) = n + m(\rho + T) < (m + p)(\rho + T),
\]

\[
\text{rank}(Z_P) = n + m\rho < (m + p)\rho.
\]

These relations are indeed the basis for the so-called “intersection algorithms” in subspace identification and also can be seen as algebraic formulations of the well known “Willems’ fundamental lemma” [28]. Nonetheless, in this case, any finite (deterministic) trajectory \( z_{\text{init}} \) of the system (5) belongs to the column span of \( Z_P \). As such, provided that \( z_{\text{init}} \) is an “admissible” sequence of input/output pairs of the given deterministic system, then (26) has a solution.

We are now ready to recast the control problem (7) in a data driven fashion as follows:

\[
\begin{align*}
\text{minimize} & \quad J \left( \begin{bmatrix} y_f^d \\ u_f \end{bmatrix} \right) \\
\text{s.t.} & \quad \alpha^* = \begin{bmatrix} Z_P \\ U_F \end{bmatrix} \begin{bmatrix} z_{\text{init}} \\ u_f \end{bmatrix}, \\
& \quad y_f^d = \hat{Y}_F \alpha^*, \\
& \quad u(k) \in \mathcal{U}, y_f^d(k) \in \mathcal{Y}, k \in [t, t + T), \\
& \quad \|y_f^d(k) - y_r(k)\|^2_F + \|u(k)\|^2_R.
\end{align*}
\]  

(28)

where

\[
J \left( \begin{bmatrix} y_f^d \\ u_f \end{bmatrix} \right) = \frac{1}{2} \sum_{k=t}^{t+T-1} \|y_f^d(k) - y_r(k)\|^2_F + \|u(k)\|^2_R,
\]

(28e)

and \( \hat{Y}_F := \Pi_{Z_P, U_F}(Y_F) \), while \( z_{\text{init}} \) and \( u_f \) are defined as in (23) and (24).

3.3 Related works

Recent papers have discussed very similar problems starting from a deterministic viewpoint, i.e., assuming that \( e(t) = 0 \), \( \forall t \) in (5). Among them, we consider the problem with elastic net regularization in [13, Section IV.D], that we rewrite for the control problem considered in this work by using our notation as follows:

\[
\text{minimize} \quad J \left( \begin{bmatrix} y_f^d \\ u_f \end{bmatrix} \right) + \lambda_1 \|\alpha\|_1 + \lambda_2 \|(I - \Pi)\alpha\|_p
\]  

(29a)

\[
\text{s.t.} \quad \begin{bmatrix} z_{\text{init}} \\ u_f \\ y_f^d \end{bmatrix} = \begin{bmatrix} Z_P \\ U_F \\ Y_F \end{bmatrix} \alpha,
\]

(29b)

\[
u(k) \in \mathcal{U}, \quad y_f^d(k) \in \mathcal{Y}, \quad k \in [t, t + T),
\]

(29c)

where \( \Pi \) is the orthogonal projector onto the column span of \( \begin{bmatrix} Z_P^\top \\ U_F^\top \end{bmatrix} \), i.e.,

\[
\Pi := \begin{bmatrix} Z_P^\top \\ U_F^\top \end{bmatrix} \begin{bmatrix} Z_P \\ U_F \end{bmatrix}^\top.
\]

(29d)

The following proposition provides the connection between the problem in (29) and the one in (28).

**Theorem 3 (Optimal regularization)** Assuming the cost \( J(\cdot) \) in (29) is equal to (28e), then the solution to problem (28) coincides with the one of (29) for \( \lambda_1 = 0 \) and \( \lambda_2 = \infty \).

**Proof** For \( \lambda_1 = 0 \) and \( \lambda_2 = +\infty \), Problem 29 reduces to

\[
\begin{align*}
\text{minimize} & \quad J \left( \begin{bmatrix} y_f^d \\ u_f \end{bmatrix} \right) \\
\text{s.t.} & \quad \begin{bmatrix} z_{\text{init}} \\ u_f \\ y_f^d \end{bmatrix} = \begin{bmatrix} Z_P \\ U_F \\ Y_F \end{bmatrix} \alpha, \\
& \quad \|(I - \Pi)\alpha\| = 0 \\
& \quad u(k) \in \mathcal{U}, \quad y_f^d(k) \in \mathcal{Y}, \quad k \in [t, t + T).
\end{align*}
\]  

(30)

In addition, by decomposing \( Y_F := \hat{Y}_F + \tilde{Y}_F \), where \( \tilde{Y}_F = \)
\[ \Pi_{Z_p, U_p}(Y_F), \text{ and } \hat{Y}_F = Y_F - \Pi_{Z_p, U_p}(Y_F), \text{ we have that} \]
\[ \hat{Y}_F = Y_F \Pi \quad \hat{Y}_F = Y_F(I - \Pi). \]
Then, when \((I - \Pi)\alpha = 0\), we have
\[ Y_F\alpha = Y_F\Pi\alpha + Y_F(I - \Pi)\alpha = Y_F\Pi\alpha = \hat{Y}_F\alpha. \]

This result not only shows the connection between the control problem considered in this work and the regularized one proposed in [13], but also puts the results shown in [13], where the role of \(\lambda_1\) and \(\lambda_2\) is evaluated experimentally, into a rigorous frame.

4 The \(\gamma\)-DDPC scheme

4.1 LQ decomposition in DDPC

In this Section, we reformulate problem (28) by exploiting the LQ decomposition of the Hankel data matrices. On the one hand, this procedure leads to an even closer connection with subspace identification. On the other, it allows us to parametrize the solution to (28) in terms of a lower dimensional parameter vector. We thus consider the LQ decomposition of the joint input-output block Hankel matrix \(Z_{\text{data}}\) in (22), namely:

\[
\begin{bmatrix}
Z_p \\
U_F \\
Y_F
\end{bmatrix} = 
\begin{bmatrix}
L_{11} & 0 & 0 \\
L_{21} & L_{22} & 0 \\
L_{31} & L_{32} & L_{33}
\end{bmatrix} 
\begin{bmatrix}
Q_1 \\
Q_2 \\
Q_3
\end{bmatrix}, \tag{31}
\]

where the matrices \(\{L_{ij}\}_{i=1}^3\) are all non-singular (under the assumptions of Lemma 3) and \(Q_i\) have orthonormal rows, i.e. i.e., \(Q_iQ_i^\top = I\), for \(i = 1, \ldots, 3\), \(Q_iQ_j^\top = 0\), \(i \neq j\).

First of all, let us observe that \(\hat{Y}_F := \Pi_{Z_p, U_p}(Y_F)\) in Lemma 2 can be expressed in terms of the LQ decomposition (31) as:

\[ \hat{Y}_F = \begin{bmatrix}
L_{31} & L_{32}
\end{bmatrix} 
\begin{bmatrix}
Q_1 \\
Q_2
\end{bmatrix}. \tag{32} \]

By exploiting (31) and (32), we can thus express the constraint in (26) as follows:

\[
\begin{aligned}
z_{\text{init}} &= Z_p\alpha = L_{11}Q_1\alpha, \tag{33a} \\
u_f &= U_F\alpha = \begin{bmatrix}
L_{21} & L_{22}
\end{bmatrix} 
\begin{bmatrix}
Q_1 \\
Q_2
\end{bmatrix} \alpha, \tag{33b}
\end{aligned}
\]

where (33a) accounts for the initial condition of the predictive control problem, whereas (33b) links the optimal \(\alpha\) with the control input. The predicted output in (25) can then be rewritten as

\[ \hat{y}_f = \hat{Y}_F\alpha = \begin{bmatrix}
L_{31} & L_{32}
\end{bmatrix} 
\begin{bmatrix}
Q_1 \\
Q_2
\end{bmatrix} \alpha^*, \tag{34} \]

where \(\alpha^*\) is the minimum-norm solution to (33).

We can now leverage on triangular structure of (33) to characterize the minimum-norm solution \(\alpha^*\). In particular, (33a) always admits a solution (see Lemma 3 and Remark 3), that satisfies the following property.

Lemma 4 (Definition of \(\gamma_1\)) Let \(\alpha_{\text{init}} \in \mathbb{R}^N\) be the minimum-norm \(\alpha\) solving (33a). Then, by defining \(\gamma_1^\star \in \mathbb{R}^{(m+p)p}\) as the unique solution of

\[ z_{\text{init}} = L_{11}\gamma_1^*, \tag{35} \]

\(\alpha_{\text{init}}\) can be written as \(\alpha_{\text{init}}^* = Q_1^\top \gamma_1^*, \) so that

\[ \alpha_{\text{init}}^* \in \text{colspan}(Q_1^\top). \]

\textbf{Proof} Since \(Z_p\) has full column rank, so does \(L_{11}\) and any solution \(\alpha\) to (33a) must satisfy

\[ Q_1\alpha = L_{11}^{-1}z_{\text{init}} = \gamma_1^*. \]

The minimum-norm solution \(\alpha_{\text{init}}^*\) can be found by as

\[ \alpha_{\text{init}}^* = Q_1^\top \gamma_1^* = Q_1^\top \gamma_1^*, \]

can then be rewritten as

\[ u_f = L_{21}\gamma_1^* + L_{22}\gamma_2 \]

\[ \gamma_2 = Q_2\alpha \]

Based on this representation, we can provide additional insights on \(\alpha^*\), through the following result.

Lemma 5 (Definition of \(\gamma_2\)) Let \(\alpha_{f}^* \in \mathbb{R}^N\) indicate the minimum-norm \(\alpha\) solving (36). Accordingly, define \(\gamma_2^* \in \mathbb{R}^{mT}\) as the unique solution of the least squares problem

\[ L_{22}\gamma_2 = u_f - L_{21}\gamma_1^* \]

where \(\gamma_1^*\) is defined in Lemma 4. Then \(\alpha_{f}^*\) can be written as:

\[ \alpha_{f}^* = Q_2^\top \gamma_2^*, \]

so that

\[ \alpha_{f}^* \in \text{colspan}(Q_2^\top). \]

\textbf{Proof} Since the matrix \([Z_p^\top \ U_F^\top]^\top\) has full rank, also \(L_{22}\) has full rank and is thus invertible. Any solution \(\alpha\) to

\[ u_f = L_{21}Q_1\gamma_1^* + L_{22}Q_2\alpha \]

\[ \alpha_{f}^* = Q_2^\top \gamma_2^*, \]

where \(\gamma_2^*\) is the minimum-norm solution to (36).

\[ u_f = L_{21}Q_1\gamma_1^* + L_{22}Q_2\alpha \]

\[ \alpha_{f}^* \in \text{colspan}(Q_2^\top). \]
must therefore satisfy
\[ Q_2 \alpha = L_{22}^{-1} [u_f - L_{21} \gamma_1^*]. \]
and the minimum-norm solution is given by
\[ \alpha_j^* = Q_j^1 L_{22}^{-1} [u_f - L_{21} \gamma_1^*] = Q_j^2 L_{22}^{-1} [u_f - L_{21} \gamma_1^*] = Q_j^2 \gamma_j^*. \]

From Lemma 4 and 5, we can then characterize the minimum-norm parameter \( \alpha \) of the whole behavioral model in (28b)-(28c) as follows.

**Theorem 4 (Decomposition of \( \alpha^* \))** Let \( \alpha_{\text{init}} \in \mathbb{R}^N \) and \( \alpha_j^* \in \mathbb{R}^N \) be defined as in Lemma 4 and 5, respectively. Then, they satisfy the following properties:

1. \( \alpha_{\text{init}} = Q_1^1 \gamma_1^*; \)
2. \( \alpha_j^* = Q_j^1 \gamma_j^*; \)
3. \( \alpha_{\text{init}} \) is orthogonal to \( \alpha_j^*; \)
4. \( Q_1 \alpha_j^* = 0 \) and \( Q_2 \alpha_{\text{init}} = 0 \)
5. \( Q_3 \alpha_j^* = Q_3 \alpha_{\text{init}} = 0. \)

Therefore, \( \alpha^* = \alpha_{\text{init}} + \alpha_j^* \) is the minimum-norm vector satisfying the conditions:

\[
\begin{bmatrix}
z_{\text{init}} \\
u_f
\end{bmatrix} = \begin{bmatrix}
Z_P \\
U_P
\end{bmatrix} \alpha^* = \begin{bmatrix}
L_{11} & 0 \\
L_{21} & L_{22}
\end{bmatrix} \begin{bmatrix}
Q_1 \\
Q_2
\end{bmatrix} \alpha^*
= \begin{bmatrix}
L_{11} & 0 \\
L_{21} & L_{22}
\end{bmatrix} \begin{bmatrix}
\gamma_1^* \\
\gamma_2^*
\end{bmatrix},
\]

**Proof** Conditions 1 and 2 have been proved in Lemmas 4 and 5 respectively. Condition 3 and 4 are direct consequences of the fact that \( Q_1^1 Q_2 = 0 \). Finally, Condition 5 derives from the fact that \( Q_3^i Q_i = 0, \) for \( i = 1, 2 \). It is also straightforward to verify that, indeed, \( \alpha^* = \alpha_{\text{init}} + \alpha_j^* \) is a solution to (39). The fact that it is the minimum-norm solution derives from the fact that \( \alpha^* \) belongs to the column space of \([Q_1^1 Q_2^1]. \)

The properties highlighted in Section 4.1 allows us to reformulate the DDPC problem as follows:

\[
\begin{align*}
\minimize_{\gamma_1,\gamma_2} & \quad J \left( \begin{bmatrix}
y_f^q \\
u_f
\end{bmatrix} \right) \\
\text{s.t.} & \quad \begin{bmatrix}
z_{\text{init}} \\
u_f \\
y_f^q
\end{bmatrix} = \begin{bmatrix}
L_{11} & 0 \\
L_{21} & L_{22}
\end{bmatrix} \begin{bmatrix}
\gamma_1 \\
\gamma_2
\end{bmatrix}, \\
u(k) \in \mathcal{U}, \ y_f^q(k) \in \mathcal{Y}, \ k \in [t, t + T), \end{align*}
\]

**Algorithm 1 \( \gamma \)-DDPC at time \( t \)**

**Input:** Matrices \( \{L_{ij}\}_{i,j=1}^2 \); penalties \( Q \geq 0 \), \( R \succ 0 \); target \( y_f; \) constraint sets \( \mathcal{U} \) and \( \mathcal{Y} \); initial conditions \( z_{\text{init}}. \)

1. **Find** \( \gamma_1^* \) via (41);
2. **Optimize** \( \gamma_2 \) by solving (42);
3. **Construct** \( u_f \) according to (37);
4. **Extract** the first optimal input from \( u_f \).

**Output:** Optimal input \( u^*(t) \).

where the cost is defined in (28c) and we reshape the predictor based on the properties of the minimum-norm \( \alpha \) highlighted in Theorem 4.

By looking at (40), it can be easily noticed that the cost and the value constraints in (40c) are independent of \( \gamma_1 \). In turn, \( \gamma_1 \) is solely determined by the initial conditions \( z_{\text{init}}. \) As such, \( \gamma_1 \) is not a proper optimization variable, but acts as a constraint that can be explicitly solved by setting:
\[
\gamma_1^* = L_{11}^{-1} z_{\text{init}}. \tag{41}
\]

It is worth stressing once more that, according to Lemma 4, \( \gamma_1^* \) and \( \gamma_2 \) found through (41) coincides with the ones leading to the minimum-norm \( \alpha \) satisfying the initial conditions.

The constrained optimization problem to be solved at each time instant thus result into a reduced problem on \( \gamma_2 \) only, i.e.,

\[
\begin{align*}
\minimize_{\gamma_2} & \quad J \left( \begin{bmatrix}
y_f^q \\
u_f
\end{bmatrix} \right) \\
\text{s.t.} & \quad \begin{bmatrix}
y_f \\
u_f
\end{bmatrix} = \begin{bmatrix}
L_{21} & L_{22} \\
L_{31} & L_{32}
\end{bmatrix} \begin{bmatrix}
\gamma_1^* \\
\gamma_2^*
\end{bmatrix},
\end{align*}
\]

with \( \gamma_1(t) \) fixed at the solution of (41).

According to this decomposition, we propose the \( \gamma \)-DDPC scheme, summarized in Algorithm 1. Apart from inheriting the properties of the predictor highlighted in Section 4.1 with respect to noise handling, the \( \gamma \)-DDPC scheme is likely to be computationally advantageous. Indeed, the dimension of the optimization variable \( \gamma_2 \in \mathbb{R}^{mT} \) in (42) is likely to be considerably smaller than the one of \( \alpha \in \mathbb{R}^N \). At the same time, retrieving \( \gamma_1 \) requires the inversion of a matrix with dimensions dictated by the chosen \( \rho \).

**Remark 4 (Choice of \( \rho \) (part II))** The length of the “past” window plays a pivotal role in shaping the performance of the predictive controller. On the one hand, \( \rho \) should be chosen by following an identification-oriented reasoning (see Remark 2). On the other, a smaller \( \rho \) reduces the number of data needed to solve the DDPC prob-
lem ($N_{data} := N + T + \rho$), and it would result in a computationally lighter DDPC problem. Its value has thus to be selected by trading-off between these opposite requirements.

4.2 Explaining regularization in DDPC

By looking at the DDPC problem from a different angle, the results presented so far allow us to have a clearer vision on the actual effect that additional regularization terms have on the optimal control action generated when solving (28). We stress that currently the use of regularization is the only strategy proposed in the literature to cope with stochastic noise in DDPC.

The properties highlighted in Theorem 4 indicate that $Q_3 \alpha$ should be set to zero, if one seeks to reduce the effect of noise on the predictions exploited to determine the optimal control action. At the same time, one should not excessively shrink the values of $Q_3 \alpha$, for $i = 1, 2$. While these two conflicting requirements on $\alpha$ can be easily accommodated when decomposing the predictor as in Section 4.1, this operation is not as easy when the predictor in (28b)-(28c) is used as it is. Indeed, in this last case, one can only try shrink the whole vector $\alpha$ by introducing a regularizer in the cost, as already proposed in [13, 15]. Although such procedure has proven to be effective, the regularization strength has to be well calibrated to trade-off between reducing the norm of $\alpha$ and retaining the information needed to produce a meaningful control action. In turn, achieving this balance requires the fine tuning of the regularization penalty, representing a well-known drawback of regularization-based DDPC approaches. Indeed, existing procedures generally require closed-loop experiments to calibrate the regularization parameters, which can endanger the safety of the plant, ultimately limiting the applicability of existing DDPC strategies.

5 A benchmark case study

To assess the effectiveness of the proposed $\gamma$-DDPC scheme, while validating the conclusions drawn in Section 4.2, we consider the same benchmark example proposed in [4]. Therefore, the unknown plant to be controlled is assumed to be described by the following model:

\[
\begin{align*}
    x_{k+1} &= \begin{bmatrix} 0.7326 & -0.0861 \\ 0.1722 & 0.9909 \end{bmatrix} x_k + \begin{bmatrix} 0.0609 \\ 0.0064 \end{bmatrix} u_k + K e_k, \\
    y_k &= \begin{bmatrix} 0.0064 & 0.0064 \end{bmatrix} + e_k,
\end{align*}
\]

where the innovation is assumed to be zero-mean and Gaussian distributed, i.e., $e \sim \mathcal{N}(0, \sigma^2)$. By considering a prediction horizon of length $T = 30$ and $\rho = 23$ (selected according to Remark 2), we design the DDPC controller solving a zero regulation problem by running Algorithm 1 with $Q = I$, $R = 10^{-3}$ and $y_i(t) = 0 \forall t$, as in [4]. To have a quantitative assessment of performance, for all closed-loop tests we consider the following indexes:

\[
J = \sum_{t=0}^{T-1} ||y_t||_Q^2 + ||u_t||_R^2, \quad J_u = \sum_{t=0}^{T-1} u_t^2,
\]

that allow us to have a compact information on the tracking performance and the input effort in testing.

Let us initially look at the case in which $K = 0$, so that the innovation simply acts as a measurement noise. When collecting the data used to construct the predictor in (42b), we consider for increasing values of noise variance $\sigma^2$. For each noise level, we perform 30 Monte Carlo simulations of length $T = 1000$ with a random input sequence, uniformly distributed in the interval $[-5, 5]$. Closed-loop performance is then evaluated for each predictive model and level of noise by running noiseless tests of length $T_v = 50$, always starting from the same initial condition. Fig. 1 shows the box plots of the indexes resulting from using the 30 predictors available for each noise level. Both the performance and control effort are quite consistent when the average SNR is high. Instead, a slight degradation in performance is experienced when the average noise corrupting the data used to construct the predictor is around 3 dB, along with an increase in the control effort required during testing. Similar conclusions can be drawn in terms of performance for the case in which $K \in \mathbb{R}^{1 \times 2}$ is randomly chosen according to a normal distribution, with all the eigenvalues of $A - KC$ being inside the unit circle (see Fig. 1). These results generally show that the proposed $\gamma$-DDPC strategy is robust to noise, while not requiring tuning any hyperparameter except for $\rho$, which is common to all DDPC approaches. This consideration is further confirmed by the performance achieved over a noisy validation set, as shown in Fig. 2. We stress that the difference in variance between the inputs in Fig. 2(a) and the ones in Fig. 2(b) is due to the process noise, that affects the closed loop states in this last case. Nonetheless, such a dissimilarity is barely perceivable on the outputs.

We then compare the performance of the $\gamma$-DDPC scheme with the ones of the oracle MPC, i.e., the predictive controller designed by directly leveraging on (43) and exploiting a Kalman filter to estimate the state $\hat{x}$.

Note that, by exploiting the oracle MPC, the obtained

---

4. All tests have been carried out on an M1 chip, running MATLAB 2021a, while the optimization problems are solved with CVX [18,19].

5. The initial state is assumed to be slightly uncertain, i.e., $\hat{x}_0 = x_0 + \nu$ with $\nu \sim \mathcal{N}(0, 10^{-6} I)$. 

10
Fig. 1. Validation tests on a noiseless closed-loop: performance indexes vs noise levels over 30 Monte Carlo predictors. The noise level is assessed via the average signal-to-noise ratio (SNR) on the output.

Table 1
Performance indexes in (44) (mean ± standard deviation) vs predictive strategy for a randomly chosen $K$ and SNR = 6 dB.

|                      | Noiseless validation | Noisy validation |
|----------------------|----------------------|------------------|
|                      | $J_o$                | $J_u$            | $\mathbf{J}$    | $\mathbf{J}_u$  |
| Oracle               | 22.34                | 55.46            | 25.98±4.82      | 120.96±12.96    |
| $\gamma$-DDPC       | 22.38±0.03           | 58.12±2.82       | 25.53±5.14      | 118.09±11.84    |
| ARX                  | 22.43±0.03           | 88.64±3.68       | 26.09±5.06      | 127.26±13.04    |
| “Oracle” ARMAX      | 23.35±(7.4·10⁻⁴)    | 55.51±1.05       | 28.55±5.33      | 189.00±4.47     |
| ARMAX                | 23.31±2.40           | 65.26±29.63      | 28.46±8.55      | 162.80±27.59    |

performance indexes (44) are:

$$J^o = 22.34, \quad J^o_u = 55.46.$$  \hfill (45)

Clearly, by comparing the results in Fig. 1 with the “ideal” indexes in (45), both the closed-loop performance and the control effort characterizing the $\gamma$-DDPC solution are generally comparable to the oracle ones, with a slight drop in performance and increase in the required control effort when the noise on batch data becomes prominent. As shown in Fig. 3, this behavior is due to slight differences in the optimal input sequences over the transient. Except for these steps and despite the presence of noise, both the closed-loop response and the input sequence obtained with $\gamma$-DDPC are still rather consistent with the oracle ones, showing that the proposed strategy is robust to noise, while not requiring any hyper-parameter tuning, except for $\rho$. 
(a) $K = 0$ and $\text{SNR} = 14$    (b) Random $K$ and $\text{SNR} = 11$

Fig. 2. Validation tests on a noisy closed-loop: average (dashed lines) closed-loop response and inputs with their standard deviations (shaded area) over the 30 predictors. The signal-to-noise-ratio is measured in dB.

By considering the more challenging scenario when $K$ in (43) is randomly chosen, we further compare the performance of Algorithm 1 with the ones achieved by using an identified model of the plant for the lowest SNR tested. To this end, we keep the input/output structure of the predictor by identifying both an autoregressive model with exogenous inputs (ARX) of order 23, an “oracle” autoregressive moving average model with exogenous inputs (ARMAX) of order 2 and ARMAX models with orders selected according to Remark 2. As shown in Table 1, the values of the performance index $J$ are generally similar, with the use of identified models leading to slightly smaller standard deviations when the closed-loop tests are performed in an (ideal) noiseless setting. Nonetheless, the use of all identified models tends to require an additional control effort, especially when a noisy closed-loop test is considered (see $J_u$ in Table 1). When using the ARMAX identified without exploiting any prior on the system, performance deteriorates. This result might stem from the fact that only the deterministic part of the learned model is used to design the MPC, while the rest is discarded, but this information is not exploited during the identification phase. By still focusing on the case of a randomly chosen $K$ and an average Signal-to-Noise ratio of 6 dB, we finally evaluate how $\gamma$-DDPC performs with respect to the oracle MPC when increasing $N_{\text{data}}$ over noisy closed-loop tests. As shown in Fig. 4, the difference between the overall cost and the required control effort tends to decrease with the number of data, in line with established results in system identification. Note that, the same trend characterizes closed-loop per-

---

6 The model is identified with N4SID [27].

7 Accordingly, the average order of the ARMAX models is 8, while its standard deviation is 5.
5.1 On the effect of additional regularization

Let us now focus on the scenario where $K = 0$ and $\sigma^2$ is such that $\text{SNR} \approx 14$ dB. We now study the effect of an additional 2-norm regularization on $\gamma_2$, with $\beta > 0$ indicating the associated penalty. As shown in Fig. 5, the performance index $J$ tends to be rather insensitive to the additional regularization term up to a certain value of $\beta$. However, when $\beta$ increases sufficiently, performance tends to deteriorate, while the input tends to consistently decrease. Since such a behavior is certainly undesirable, this result validates in this experimental case the claims in Section 4.2. Indeed, the additional regularization leads to a deterioration of performance, likely to be induced by the change that the regularization enforces on the actual performance-oriented cost.

To prove the effectiveness of our structural choices, within the same framework we consider the DDPC problem with the predicted output defined as

$$y_f = \sum_{i=1}^{3} L_{3i} \gamma_i,$$

and $\gamma_3$ not set to zero beforehand, as in the proposed $\gamma$-DDPC approach. In this case, $\gamma_3$ is steered to zero via an addition of a 2-norm regularization term on it in the cost. Let $\eta > 0$ denote the weight associated to this additional term. As shown in Fig. 6, only by heavily regularizing $\gamma_3$ we obtain performance comparable with the ones obtained with the $\gamma$-DDPC algorithm (see Fig. 1(a) and Fig. 1(c)). Indeed, for low $\eta$, the controller is ineffective and the system actually operates in open-loop. These results once again show the expected detrimental effect of poor choices of the regularization parameter,
5.2 Comparison with regularized DDPC schemes

We now compare the performance attained with the $\gamma$-DDPC strategy with the one achieved by (i) designing the predictive control problem through the approach proposed in [5] and (ii) by solving (29) by setting $\lambda_1 = 0$ according to Theorem 3.

As shown in Fig. 7, when exploiting the approach proposed in [5], the choice of the regularization parameter is crucial to attain satisfactory performance, so as to balance the need to have a meaningful control action and the one of rejecting noise. As such, this behavior validate the conclusions drawn in Section 4.2. Once, these results show that regularizing the whole parameter vector $\alpha$ requires a careful selection of the regularization penalty, which is instead directly enforced in the proposed $\gamma$-DDPC scheme.

Instead, the results reported in Fig. 8 validate the parallelism between our formulation and the one proposed in [13], as highlighted in Theorem 3. Specifically, by comparing Fig. 8 and Fig. 6 for $\lambda_1 = 0$, it is clear that $\lambda_2$ has an effect similar to the one of $\eta$ and that (as expected) it is advisable to set $\lambda_2$ as high as possible.

5.3 Sensitivity to $\rho$

The only hyper-parameter of the proposed $\gamma$-DDPC strategy is the maximum delay $\rho$ characterizing (11), which is shared with all the other DDPC approaches prosed in the literature. As shown in Fig. 9(a) and Fig. 9(c), when the SNR is around 14 dB, excessively...
small values of $\rho$ results into poorer performance with respect to the ones attained when $\rho > 10$. In this scenario, performance are thus insensitive to the choice of $\rho$, as long as it is big enough. Instead, both the performance and the input effort tend to result into the same indexes for high $\rho$. On the other hand, when the SNR increases (see Fig. 9(b) and Fig. 9(d)), performance tend to be more affected by the choice of $\rho$, with both low and high values of this parameter resulting in a deterioration of performance. This result was somehow expected, given the pervasive presence of noise in the data used to construct the predictor of the DDPC problem.

6 Conclusions

In this paper, we have provided a rigorous formulation of data-driven predictive control in a stochastic setting. As a by-product, we have discussed the role of regularization, which has been often advocated in the literature as a tool to extend deterministic ideas to the noisy setting. Specifically, we have shown that a unique regularization term is not sufficient either to reduce the impact of noise on the predictions and to comply with the initial conditions. Instead, by relying on the predictor decomposition proposed here (namely, the $\gamma$-DDPC approach), not only we can find the best balance between a proper control action and a meaningful system description via the data matrices, but such a trade-off is automatically retrieved with no additional regularization terms, except for the existing penalty on the input in the cost function. This fact leads to the additional advantage that no closed-loop tuning of the penalty terms is needed.

Future works will be devoted to the analysis of regularization in case DDPC is applied to nonlinear systems.

References

[1] H. Akaike. Fitting autoregressive models for prediction. Annals of the Institute of Statistical Mathematics, 21:243–247, 1969.
