

Sample- and computationally efficient data-driven predictive control

Mohammad Alsalti*, Manuel Barkey*, Victor G. Lopez and Matthias A. Müller

Abstract—Recently proposed data-driven predictive control schemes for LTI systems use non-parametric representations based on the image of a Hankel matrix of previously collected, persistently exciting, input-output data. Persistence of excitation necessitates that the data is sufficiently long and, hence, the computational complexity of the corresponding finite-horizon optimal control problem increases. In this paper, we propose an efficient data-driven predictive control (eDDPC) scheme which is both more sample efficient (requires less offline data) and computationally efficient (uses less decision variables) compared to existing schemes. This is done by leveraging an alternative data-based representation of the trajectories of LTI systems. We analytically and numerically compare the performance of this scheme to existing ones from the literature.

I. INTRODUCTION

Model predictive control (MPC) [1] is an optimization-based control technique that involves solving a finite-horizon optimal control problem repeatedly at each time instant, and then applying the first part of the optimal input to the system. In recent years, the development of data- and learning-based MPC schemes has received a significant amount of attention. Here, the unknown system model or the MPC controller itself are either approximated using machine learning techniques or directly computed using input-output data of the system, see, e.g. [2], [3] and the references therein.

One prominent approach to data-driven system analysis and control exploits the behavioral approach to systems theory [4]. For instance, Willems' fundamental lemma [5] provides a non-parametric representation of the finite-length behavior of a discrete-time LTI system as the image of a Hankel matrix of a single persistently exciting (PE) trajectory. In particular, it asserts that any trajectory of an LTI system can be expressed as a linear combination of time-shifts of a single measured trajectory. This result was successfully used for system analysis and control design for linear systems and extended to classes of nonlinear systems. The reader is referred to [3] and [6] for a comprehensive review.

Among several successful applications of the fundamental lemma is its use in data-driven predictive control schemes (DDPC). Specifically, instead of relying on the typical steps of deriving and identifying a predictive model for the MPC, one can use previously collected input-output data as a data-based non-parametric predictive model. This was first proposed in [7] and further developed in [8]. Theoretical

guarantees were established in [9], [10]. Extensions of this data-driven predictive control scheme to classes of nonlinear systems appeared in [11]–[14]. Several other extensions and applications of this scheme include: stochastic DDPC [15], [16], segmented DDPC [17], distributed DDPC [18], [19] and generalized DDPC [20], among many others.

Crucial to the successful application of this scheme is that the offline collected data is persistently exciting (PE). This, however, necessitates that the data is sufficiently long, and this required number of data points increases with increased system order, number of inputs and prediction horizon length. As a result, the computational complexity of solving the corresponding finite-horizon optimal control problem increases. Moreover, it may not be easy to obtain such PE data in practice. For instance, the length of the available data might not be long enough to allow for application of DDPC schemes with long prediction horizons. In such circumstances, it was shown in [21] how one can obtain alternative non-parametric representations of LTI systems using a small number of (potentially irregularly measured) data points. In this work, we exploit the proposed representation in [21] and illustrate its use in an efficient DDPC scheme.

The contributions of this paper are as follows: we propose an efficient data-driven predictive control (eDDPC) scheme which is both sample efficient (requires less offline data) and computationally efficient (includes less decision variables) compared to existing schemes. We show that the closed-loop behavior is equivalent to that of existing schemes and hence enjoys the same theoretical guarantees. We then analytically and numerically compare the performance of the proposed scheme with other existing ones from the literature and illustrate how it consistently outperforms them.

The paper is structured as follows: Section II introduces notation and necessary preliminaries. In Section III, we formulate the eDDPC scheme. Section IV includes analytical and numerical comparisons to existing schemes and Section V concludes the paper.

II. PRELIMINARIES

The sets of integers, natural and real numbers are denoted by $\mathbb{Z}, \mathbb{N}, \mathbb{R}$, respectively. The restriction of integers to an interval is denoted by $\mathbb{Z}_{[a,b]}$, for $b > a \in \mathbb{Z}$. We use I_m to denote an $m \times m$ identity matrix and $0_{n \times m}$ to denote an $n \times m$ matrix of zeros. When the dimensions are clear from the context, we omit the subscript for simplicity. For a matrix $M \in \mathbb{R}^{m \times n}$, we denote its image by $\text{im}(M)$ and its kernel by $\text{ker}(M)$. When a basis of $\text{ker}(M)$ is to be computed, we write $N = \text{null}(M)$ which returns a matrix N of appropriate dimensions such that $MN = 0$. For a

Leibniz University Hannover, Institute of Automatic Control, 30167 Hannover, Germany. E-mail: alsalti@irt.uni-hannover.de, barkey@irt.uni-hannover.de, lopez@irt.uni-hannover.de, mueller@irt.uni-hannover.de.

This work has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 948679).

*Mohammad Alsalti and Manuel Barkey contributed equally to this work.

symmetric positive definite matrix $P = P^\top \succ 0$, we define the weighted norm of a vector x as $\|x\|_P := \sqrt{x^\top P x}$. In contrast, $\|x\|_i$, $i \in \{1, 2, \infty\}$, denote the standard ℓ_1 , ℓ_2 (Euclidean), and ℓ_∞ norms, respectively.

The set of infinite length, q -variate, real-valued time series $w = (w_0, w_1, \dots)$ is denoted by $(\mathbb{R}^q)^\mathbb{N}$. For $T \in \mathbb{N}$, the set of finite-length, q -variate, real-valued time series $w = (w_0, w_1, \dots, w_{T-1})$ is denoted by $(\mathbb{R}^q)^T$. With slight abuse of notation, we also use w to denote the stacked vector $w = [w_0^\top \ w_1^\top \ \dots \ w_{T-1}^\top]^\top \in \mathbb{R}^{qT}$, and a window of it by $w_{[a,b]}$ where $0 \leq a < b \leq T-1$. The Hankel matrix of depth L of w is defined as

$$\mathcal{H}_L(w) = [w_{[0,L-1]} \ w_{[1,L]} \ \dots \ w_{[T-L,T-1]}].$$

The (finite-length) behavior $(\mathcal{B}|_T)$ \mathcal{B} of a dynamical system is defined as the set of all (finite-) infinite-length trajectories that can be generated by the system (cf. [4]). A trajectory of length T of the system is denoted by $w \in \mathcal{B}|_T \subset (\mathbb{R}^q)^T$. The system is linear if \mathcal{B} is a subspace and it is time-invariant if it is invariant to the action of the *shift operator*, defined as $\sigma^j w(k) := w(k+j)$ for $j \in \mathbb{N}$.

Let $u_t \in \mathbb{R}^m$ and $y_t \in \mathbb{R}^p$ denote the inputs and outputs of system \mathcal{B} at time t . We define a partitioning of the variable $w_t \in \mathbb{R}^q$ such that $w_t = \begin{bmatrix} u_t \\ y_t \end{bmatrix}$. The set of discrete-time LTI systems with q variables and bounded complexity (m, n, ℓ) is denoted by $\mathcal{L}_{m,n,\ell}^q$, where $q = m+p$ and (m, n, ℓ) denote (i) the number of inputs $\mathbf{m}(\mathcal{B}) \leq m$, (ii) the order of the system $\mathbf{n}(\mathcal{B}) \leq n$, and (iii) the lag of the system $\mathbf{l}(\mathcal{B}) \leq \ell$, which is the observability index in the state-space framework. These integers satisfy the following relation $\ell \leq n \leq p\ell$ [22].

A finite-dimensional LTI system $\mathcal{B} \in \mathcal{L}_{m,n,\ell}^q$ admits a kernel representation [4, Part I]

$$\mathcal{B} = \ker R(\sigma) := \{w : \mathbb{Z}_{\geq 0} \rightarrow \mathbb{R}^q \mid R(\sigma)w = 0\}, \quad (1)$$

where the operator $R(\sigma)$ is defined by the polynomial matrix

$$R(z) = \begin{bmatrix} r_1(z) \\ \vdots \\ r_g(z) \end{bmatrix} = \begin{bmatrix} r_{1,0} + r_{1,1}z + \dots + r_{1,\ell_1}z^{\ell_1} \\ \vdots \\ r_{g,0} + r_{g,1}z + \dots + r_{g,\ell_g}z^{\ell_g} \end{bmatrix}, \quad (2)$$

with $r_{i,j} \in \mathbb{R}^{1 \times q}$. Given a trajectory $w \in \mathcal{B}|_T$ of an LTI system $\mathcal{B} \in \mathcal{L}_{m,n,\ell}^q$, it holds by linearity and shift-invariance that $\text{im}(\mathcal{H}_L(w)) \subseteq \mathcal{B}|_L$. When equality holds, we obtain a data-based representation of the finite-length behavior of the system. For the case of controllable systems, Willems' fundamental lemma [5] provides conditions on the input, such that $\text{im}(\mathcal{H}_L(w)) = \mathcal{B}|_L$. This condition is known as persistence of excitation (PE) and is defined as follows.

Definition 1: A sequence $u \in (\mathbb{R}^m)^T$ is said to be persistently exciting of order L if $\text{rank}(\mathcal{H}_L(u)) = mL$.

Lemma 1: [5] Let $w \in \mathcal{B}|_T$ with $\mathcal{B} \in \mathcal{L}_{m,n,\ell}^q$ controllable. Let $u \in (\mathbb{R}^m)^T$ be PE of order $L+n$, then $\text{rank}(\mathcal{H}_L(w)) = mL+n$, and $\bar{w} \in \mathcal{B}|_L$ if and only if there exists $\alpha \in \mathbb{R}^{T-L+1}$ such that

$$\mathcal{H}_L(w)\alpha = \bar{w}. \quad (3)$$

This lemma states that each length- L trajectory of a controllable LTI system can be represented as a linear combination of time-shifts of previously collected data w (which correspond to the columns of the Hankel matrix $\mathcal{H}_L(w)$), given that the input is sufficiently exciting. We now recall the following result, which allows us to retrieve a kernel representation (2) of an LTI system directly from data.

Corollary 1: [21, Cor. 2] Let $w \in \mathcal{B}|_T$ where $\mathcal{B} \in \mathcal{L}_{m,n,\ell}^q$. Let $d \geq \ell+1$, and suppose $\text{rank}(\mathcal{H}_d(w)) = md+n$. Then, the coefficients of the polynomial matrix $R(\sigma)$ in (2) are given by the rows of $R_d \in \mathbb{R}^{pd-n \times qd}$ which satisfies $R_d \mathcal{H}_d(w) = 0$.

By exploiting the kernel structure of Hankel matrices, it was shown in [21] that one can obtain a full column rank matrix whose image is equal to the finite-length behavior of the system $\mathcal{B}|_L$. This result is summarized in the following theorem, and a sketch of the proof is provided for completeness.

Theorem 1: [21] Given $w \in \mathcal{B}|_T$ where $\mathcal{B} \in \mathcal{L}_{m,n,\ell}^q$ is controllable, let $u \in (\mathbb{R}^m)^T$ be persistently exciting of order $d+n$ for $d = \ell+1$. Then, for any $L \geq d$, $\bar{w} \in \mathcal{B}|_L$ if and only if there exists a vector $\beta \in \mathbb{R}^{mL+n}$ such that

$$P\beta = \bar{w}, \quad (4)$$

where $P = \text{null}(\Gamma)$ and Γ is given by

$$\Gamma = \begin{bmatrix} r_{1,0} & r_{1,1} & \dots & r_{1,d-1} \\ r_{2,0} & r_{2,1} & \dots & r_{2,d-1} \\ \vdots & \vdots & \ddots & \vdots \\ r_{pd-n,0} & r_{pd-n,1} & \dots & r_{pd-n,d-1} \\ & r_{1,0} & r_{1,1} & \dots & r_{1,d-1} \\ & r_{2,0} & r_{2,1} & \dots & r_{2,d-1} \\ & \vdots & \vdots & \ddots & \vdots \\ & r_{p,0} & r_{p,1} & \dots & r_{p,d-1} \\ & & \ddots & \ddots & \ddots \\ & & & r_{1,0} & r_{1,1} & \dots & r_{1,d-1} \\ & & & r_{2,0} & r_{2,1} & \dots & r_{2,d-1} \\ & & & \vdots & \vdots & \ddots & \vdots \\ & & & r_{p,0} & r_{p,1} & \dots & r_{p,d-1} \end{bmatrix}, \quad (5)$$

and $r_{i,j} \in \mathbb{R}^{1 \times q}$ are the elements of the matrix R_d in Cor. 1.

Proof: Since the input is persistently exciting of order $d+n$, it holds by Lemma 1 that $\text{rank}(\mathcal{H}_d(w)) = md+n$. Therefore, by Corollary 1, one can compute a basis for the left kernel of this matrix (i.e., $R_d \mathcal{H}_d(w) = 0$) which specifies a kernel representation of the system. The rest of the proof follows from [21, Lemma 2] and [21, Theorem 3], where it was shown that $\text{im}(P) = \mathcal{B}|_L$, for any $L \geq d$, with P defined as in the theorem statement. ■

Theorem 1 provides a data-based representation of the trajectories of an LTI system, but is different from the one provided by Lemma 1. This alternative representation has two advantages: (i) it can be obtained from a (very) small number of data points. In particular, we only need input data that is PE of order $d+n$ (independent of L). This is achieved by computing R_d as the left kernel of $\mathcal{H}_d(w)$ as in Corollary 1 (note that this can even be done if the data w potentially contains irregularly missing values, cf. [21]), which allows us to represent any trajectory \bar{w} of length L

(for arbitrary $L \geq d$) via (4). (ii) It does not result in an over-parameterization of the spanned input-output trajectories. Specifically, the regressor vector β in (4) is of dimension $mL + n$ only, i.e., independent of the number of data points.

In Section III-B we comment in more detail on the similarities and differences between Theorem 1 and Lemma 1. Then, we explain how we make use of those differences to propose an efficient data-driven predictive control scheme.

III. EFFICIENT DDPC

In this section, we propose an efficient data-driven predictive control scheme which is both sample efficient (uses less offline data points) and computationally efficient (includes less decision variables) compared to existing schemes. This is done by using the alternative data-based representation of LTI systems in Theorem 1 as a predictor in a receding horizon scheme. In Section III-A, we start by giving an overview of existing DDPC schemes and then in Section III-B we introduce our proposed eDDPC scheme in the nominal (noise-free) case.

A. Overview of existing schemes

As proposed in [7], [8], a data-driven predictive control scheme for LTI systems relies on the non-parametric representation in (3) of the finite-length trajectories of an LTI system to make predictions over a finite horizon. Later in [9], terminal equality constraints were introduced in order to exponentially stabilize a (known) equilibrium point of an unknown LTI system. Such an equilibrium point is defined in terms of the system's inputs and outputs as follows.

Definition 2: [9] We say that w^s is an equilibrium of $\mathcal{B} \in \mathcal{L}_{m,n,\ell}^q$ if the sequence $\bar{w} \in (\mathbb{R}^q)^{n+1}$ with $\bar{w}_k = w^s$, for all $k \in \mathbb{Z}_{[0,n]}$, is a trajectory of the system, i.e., $\bar{w} \in \mathcal{B}|_{n+1}$.

The DDPC scheme in [9] repeatedly solves, at each time step t , the following finite-horizon optimal control problem

$$\begin{aligned} \min_{\alpha(t), \bar{w}(t)} \quad & \sum_{k=0}^{L-1} \|\bar{w}_k(t) - w^s\|_W^2 \\ \text{s.t.} \quad & \bar{w}_{[-n, L-1]}(t) = \mathcal{H}_{L+n}(w^{\text{data}})\alpha(t) \\ & \bar{w}_{[-n, -1]}(t) = w_{[t-n, t-1]} \\ & \bar{w}_{[L-n, L-1]}(t) = w_n^s \\ & \bar{w}_k(t) \in \mathbb{W}, \quad \forall k \in \mathbb{Z}_{[0, L-1]}. \end{aligned} \quad (6)$$

The notation in (6) is summarized as follows: $w^{\text{data}} \in (\mathbb{R}^q)^T$ refers to the a priori collected input-output data, while $\bar{w}(t) \in (\mathbb{R}^q)^L$ refers to the predicted input-output trajectories (over the horizon length L) at time t . The online measurements are denoted by w_t . The stage cost is a quadratic function that penalizes the deviation from the given set point (for some weighting matrix $W \succ 0$). We use w_n^s to denote an n -dimensional vector containing n instances of w^s . Finally, \mathbb{W} denotes the constraints set and is defined as follows

$$\mathbb{W} := \{w = \begin{bmatrix} u \\ y \end{bmatrix} \mid u \in \mathbb{U}, y \in \mathbb{Y}\}, \quad (7)$$

where \mathbb{U}, \mathbb{Y} denote the input and output constraint sets, respectively, with $w^s \in \text{int}(\mathbb{W})$.

To implement the DDPC scheme, one requires an offline data sequence $w^{\text{data}} \in (\mathbb{R}^q)^T$, where the input is PE of order $L + 2n$ ($L \geq n$), as well as an initial trajectory $w_{[t-n, t-1]}$ to fix the internal state. Notice that, unlike Lemma 1, persistence of excitation here is required to be of order $L + 2n$ instead of $L + n$. This is because the length of the predicted trajectories is extended by n instances to account for the initialization step, i.e., $\bar{w}_{[-n, -1]}(t) = w_{[t-n, t-1]}$ in (6). Persistence of excitation necessitates that the Hankel matrix of the input $\mathcal{H}_{L+n}(w^{\text{data}})$ has at least as many columns as rows, and hence, $T \geq (m + 1)(L + 2n) - 1$ (cf. Definition 1). Therefore, implementing a DDPC scheme with a large prediction horizon requires a longer sequence of offline collected data. However, in many practical settings, it may be difficult to collect such a long sequence, or one may only have a short sequence of data which is not enough to span trajectories of length $L + n$.

In [17], the authors propose a partitioning of the (long) prediction horizon to multiple (n_s) segments of length $T_{\text{ini}} \geq \ell$. If the available data is long enough to span trajectories of length T_{ini} , then one can restructure the original DDPC (with longer horizon) to obtain the complete predicted trajectories. This partitioning increases the number of decision variables in the optimization problem at each time t , and the segmented scheme only shows better computational performance than standard DDPC when the prediction horizon is significantly large. In contrast, our proposed scheme (see Section III-B) reduces the sample complexity while simultaneously reducing the number of decision variables in the optimization problem, thus resulting in a more computationally efficient scheme.

For the DDPC scheme in (6), note that the dimension of the decision variable $\alpha(t) \in \mathbb{R}^{T-L-n+1}$ increases as the number of offline data points T increases (compare the lower bound for T above). This implies that for increasing system dimension, number of inputs and prediction horizon length, the problem in (6) becomes more expensive to solve. To address this issue, the authors of [23] implement a pre-processing step of computing the singular value decomposition of the Hankel matrix of data $\mathcal{H}_{L+n}(w^{\text{data}})$ in (6). Let such decomposition take the following form

$$\mathcal{H}_{L+n}(w^{\text{data}}) = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} S_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^\top \\ V_2^\top \end{bmatrix}, \quad (8)$$

where S_1 is a diagonal matrix that contains the non-zero singular values of $\mathcal{H}_{L+n}(w^{\text{data}})$, while U_i, V_i , for $i = \{1, 2\}$, are matrices of appropriate dimensions. As a consequence of the singular value decomposition, it holds that $\text{im}(U_1 S_1) = \text{im}(\mathcal{H}_{L+n}(w^{\text{data}}))$. Therefore, the authors propose using $U_1 S_1$ as a predictor in a predictive control scheme. Since $U_1 S_1$ has $m(L+n) + n$ columns, the corresponding regressor vector (denoted $g \in \mathbb{R}^{m(L+n)+n}$) is of reduced dimension. However, one still requires the same number of data points needed for the DDPC (in (6)) in order to build the matrix $\mathcal{H}_{L+n}(w^{\text{data}})$ and compute its SVD, making this scheme not sample-efficient.

In the following section, we propose a data-driven pre-

Algorithm 1 Offline data pre-processing for eDDPC

Input: Measured trajectory $w^{\text{data}} \in \mathcal{B}|_T$, satisfying $\text{rank}(\mathcal{H}_d(w^{\text{data}})) = md+n$ for $d = \ell+1$, where $\mathcal{B} \in \mathcal{L}_{m,n,\ell}^q$.

1. Compute a basis for the left kernel of $\mathcal{H}_d(w^{\text{data}})$, i.e., $R_d \mathcal{H}_d(w^{\text{data}}) = 0$.
2. Use R_d to build the matrix Γ as in (5), but with $L+n-d$ shifts.
3. Obtain $P = \text{null}(\Gamma)$.

Output: Matrix P such that $\text{im}(P) = \mathcal{B}|_{L+n}$.

dictive control scheme which is both more sample-efficient and more computationally efficient than existing schemes in the literature. This is done by implementing a few (algebraic) pre-processing steps on the collected data, which can be done prior to the online phase of the DDPC scheme.

B. Nominal eDDPC scheme

When comparing the results of Lemma 1 and Theorem 1, we see that both provide us with a non-parametric representation of the finite-length behavior of a controllable LTI system. This is true since $\text{im}(\mathcal{H}_L(w^{\text{data}})) = \text{im}(P) = \mathcal{B}|_L$. However, the following are two important distinctions between the two results that will enable us to arrive at an efficient data-driven predictive control scheme.

- D1** Theorem 1 requires at least $T \geq (m+1)(\ell+n+1) - 1$ data points to satisfy the PE condition, whereas Lemma 1 requires at least $T \geq (m+1)(L+n) - 1$. If the minimum T is chosen in both cases, then Theorem 1 will always require $(m+1)(L-\ell-1)$ less samples, for any $L > \ell+1$.
- D2** Given $\bar{w} \in \mathcal{B}|_L$, equation (3) has infinitely many solutions for $\alpha \in \mathbb{R}^{T-L+1}$, whereas the corresponding $\beta \in \mathbb{R}^{mL+n}$ vector in (4) is *unique for each trajectory*. This is because in order to fix a trajectory of an LTI system, one would only need $mL+n$ degrees of freedom that correspond to fixing the initial state (n) and specifying the input trajectory (mL). Notice that the dimension of β is *independent* of the number of collected data points, whereas the dimension of α increases with increasing T . In fact, even if the minimum T was chosen for the results of Lemma 1, then the dimension of β in Theorem 1 would still be smaller than the dimension of α by mn .

We now exploit **D1** and **D2** to propose an efficient data-driven predictive control scheme (eDDPC). Recall that in order to arrive at the data-based representation in (4), Theorem 1 implements a few (algebraic) pre-processing steps on the collected data. In Algorithm 1, we summarize these steps which can be efficiently done offline. Since the predicted trajectories are of length $L+n$ (to account for the initialization at each time), Step 2 of Algorithm 1 requires that we carry out $L+n-d$ shifts compared to $L-d$ shifts as in (5).

Once these (algebraic) pre-processing steps have been implemented, we obtain a matrix P such that $\text{im}(P) = \mathcal{B}|_{L+n}$. Now, we construct the following minimization problem,

Algorithm 2 eDDPC scheme

Input: Measured trajectory $w^{\text{data}} \in \mathcal{B}|_T$, satisfying $\text{rank}(\mathcal{H}_d(w^{\text{data}})) = md+n$ for $d = \ell+1$, where $\mathcal{B} \in \mathcal{L}_{m,n,\ell}^q$.

Offline phase: Run Algorithm 1 to obtain matrix P .

Online phase:

1. At time t , use past n input-output measurements $w_{[t-n,t-1]}$ to solve (9).
 2. Apply the input $u_t = \bar{u}_0^*(t)$ to the system.
 3. Set $t = t+1$ and return to 1.
-

which is solved at each time step t and defines our efficient data-driven predictive control scheme (eDDPC)

$$\begin{aligned} \min_{\beta(t), \bar{w}(t)} \quad & \sum_{k=0}^{L-1} \|\bar{w}_k(t) - w^s\|_W^2 \\ \text{s.t.} \quad & \bar{w}_{[-n,L-1]}(t) = P\beta(t) \\ & \bar{w}_{[-n,-1]}(t) = w_{[t-n,t-1]} \\ & \bar{w}_{[L-n,L-1]}(t) = w_n^s \\ & \bar{w}_k(t) \in \mathbb{W}, \quad \forall k \in \mathbb{Z}_{[0,L-1]}. \end{aligned} \quad (9)$$

Apart from using P to predict future trajectories in place of the Hankel matrices, the two problems in (6) and (9) have the same cost function and the same constraints. The proposed eDDPC scheme is summarized in Algorithm 2.

Since $\text{im}(P) = \text{im}(\mathcal{H}_{L+n}(w^{\text{data}}))$, the two optimization problems in (6) and (9) are in fact equivalent and the resulting closed-loop trajectories are identical. The following lemma formalizes the equivalence between our proposed eDDPC scheme and the one from [9].

Lemma 2: Given $w \in \mathcal{B}|_T$ where $\mathcal{B} \in \mathcal{L}_{m,n,\ell}^q$, let $\text{im}(\mathcal{H}_{L+n}(w^{\text{data}})) = \text{im}(P) = \mathcal{B}|_{L+n}$ where P is obtained by Algorithm 1. For a given initial trajectory $w_{[t-n,t-1]}$, if (6) and (9) are initially feasible, then the two DDPC schemes result in the same closed-loop trajectories u_t, y_t for $t \geq 0$.

Proof: The result directly follows from the fact that $\text{im}(P) = \text{im}(\mathcal{H}_{L+n}(w^{\text{data}}))$. ■

Remark 1: In [24] equivalence was established between the DeePC scheme of [8] (which solves at each time t a similar problem to (6), but without terminal constraints) and the subspace predictive control scheme (SPC) of [25]. A similar equivalence result can be established between SPC and eDDPC (9). This is omitted from this paper for space reasons, but also follows from the fact that $\text{im}(P) = \text{im}(\mathcal{H}_{L+n}(w^{\text{data}}))$.

Another implication of the equivalence of the two optimization problems (6) and (9) is that the proposed eDDPC scheme retains the same theoretical guarantees as the ones shown in [9] for the DDPC scheme specified by (6).

In the next section, we analytically compare the number of data points and the number of decision variables needed for both schemes. Later we illustrate how the proposed eDDPC scheme outperforms the existing schemes in [9], [17], [23].

IV. COMPARISON TO EXISTING SCHEMES

In this section, we analytically and numerically compare our proposed eDDPC scheme to the schemes reviewed in

TABLE I
ANALYTIC COMPARISON BETWEEN EDDPC AND EXISTING DDPC SCHEMES.

	DDPC	sDDPC	SVD-DDPC	eDDPC
$T \geq$	$(m+1)(L+2n)-1$	$(m+1)(2T_{\text{ini}}+n)-1$	$(m+1)(L+2n)-1$	$(m+1)(\ell+n+1)-1$
dim(regressor)	$T-L-n+1$	$n_s(T-2T_{\text{ini}}+1)$	$m(L+n)+n$	$m(L+n)+n$

Section III-A. These include the DDPC scheme of [9], the segmented DDPC scheme of [17] and the minimum-dimension DDPC scheme of [23]. Note that the authors of [17], [23] do not enforce a terminal equality constraint. In the following, we implement such terminal constraints to their schemes and refer to them as sDDPC and SVD-DDPC, respectively. This ensures that the closed-loop trajectories are identical and that our comparisons are fair. As mentioned in the introduction, several other DDPC schemes have been proposed in the past few years. In this paper, we restrict the comparison to the schemes discussed above since (i) the closed-loop behaviors of all of them are equivalent and, hence, sample efficiency and computational efficiency are studied and compared fairly, and (ii) the schemes in [17] and [23] explicitly address either sample efficiency [17] or computational efficiency [23] as well.

Table I analytically compares the minimum number of required offline data points as well as the complexity of the corresponding optimization problems solved at each time t (expressed in terms of the dimension of the regressor vector) for the following schemes: (i) the (DDPC) scheme of [9], (ii) the segmented DDPC (sDDPC) scheme of [17], (iii) the minimum-dimension DDPC (SVD-DDPC) scheme of [23] and finally (iv) our proposed (eDDPC) scheme (9). It can be seen that the proposed eDDPC scheme outperforms the existing ones in terms of sample complexity (uses less offline data) and computational complexity (includes less decision variables).

Compared to the SVD-DDPC scheme, eDDPC uses the same number of decision variables but uses less data points. For the sDDPC scheme, the length of each segment must satisfy $T_{\text{ini}} \geq \ell$. Here, we assumed that the prediction horizon is an integer multiple of T_{ini} , i.e., $L = n_s T_{\text{ini}}$ where n_s is the number of segments. In this case, sDDPC and eDDPC schemes use the same number of data points only when $T_{\text{ini}} = \ell = 1$. Otherwise, eDDPC always uses less data points. As for the number of decision variables, eDDPC always uses less decision variables, even when T is minimal.

Next, we carry out numerical simulations on random LTI systems to illustrate the efficiency of the eDDPC scheme. We consider systems of dimensions $n = \{4, 6, 8, 10, 12, 14\}$ (with $p = m = n - 2$). For each system order, we first generate 100 random systems (using Matlab's built-in `drss` command) and collect data by applying a PE input (sampled from a random a uniform distribution $\mathcal{U}(-1, 1)^m$) with the minimum possible length T as specified in Table I. Notice that for a fixed system order, number of inputs and number of outputs, the lag is not necessarily equal across all 100 systems. Therefore, for consistency we use a slightly longer data sequence for eDDPC than the minimum possible one

specified in Table I. Specifically, since $\ell \leq n$, we collect $T = (m+1)(2n+1)-1$ data points for eDDPC. Furthermore, in order for all four schemes to have the same initialization $w_{[t-n, t-1]}$, we set $T_{\text{ini}} = n$ for the sDDPC scheme. Notice that in this considered setting, eDDPC is still more sample-efficient than the other three schemes.

We then implement the four different DDPC schemes in a condensed formulation, i.e., the optimization problems of each scheme are reformulated such that the only decision variables are the regressors, e.g., $\alpha(t)$ in DDPC (6) or $\beta(t)$ in eDDPC (9). The objective is to stabilize the origin. We use a prediction horizon $L = 2n$, and sample the initial internal state from a uniform distribution $\mathcal{U}(-1, 1)^n$ to later obtain the initialization trajectory $w_{[-n, -1]}$. The same quadratic stage costs (with $Q = R = I_{n-2}$) and the following input/output constraints were used in all schemes

$$\mathbb{W} = \{w = \begin{bmatrix} u \\ y \end{bmatrix} \in \mathbb{R}^q \mid -5 \leq w_i \leq 5, \quad \forall i \in \mathbb{Z}_{[1, q]}\}. \quad (10)$$

This process is finally repeated and averaged for all 100 systems (per system dimension). The simulations were done on a standard Intel Core i7-10875H CPU @ 2.30GHz processor with 16 GB of RAM, using Matlab's quadprog with active-set algorithm and warm-start strategy. The corresponding MATLAB codes can be found at [26].

Table II summarizes the results. It can be seen that our proposed eDDPC scheme outperforms the other schemes in terms of number of offline data points. We report both the average (avg) computation time per iteration over 100 systems as well as the maximum (max) computation time (for any MPC iteration out of all 100 systems). It can be seen that eDDPC and SVD-DDPC schemes have the fastest average computation time of all four schemes. Furthermore, their computation times are almost equal, which is expected since they have the same (minimum) number of decision variables among all four schemes. It can be also seen that the computation time of DDPC is of the same order of magnitude as eDDPC and SVD-DDPC. This is because we use the minimum number of data points and, hence, the difference in the number of decision variables is only mn (cf. **D2** in Section III-B). However, if more data points are used, the DDPC's average computation time increases as the dimension of $\alpha(t)$ grows. As for sDDPC, it should be noted that it uses less data points than DDPC and SVD-DDPC, however it has the slowest average computation times due to the increased number of decision variables caused by the segmentation (cf. [17]).

The results of Tables I and II highlight the advantages of our proposed eDDPC approach, specifically in presence of scarce data and/or limited computational resources.

Remark 2: Numerical considerations: As the system com-

TABLE II
NUMERICAL SIMULATIONS ON 100 RANDOM SYSTEMS OF DIFFERENT COMPLEXITIES.

States	DDPC				sDDPC				SVD-DDPC				eDDPC			
	avg [ms]	max [ms]	T	$\dim(\alpha)$	avg [ms]	max [ms]	T	$\sum \dim(g_i)$	avg [ms]	max [ms]	T	$\dim(g)$	avg [ms]	max [ms]	T	$\dim(\beta)$
$n = 4$	1.19	6.98	47	36	1.68	3.94	35	56	1.06	4.22	47	28	1.04	2.94	26	28
$n = 6$	2.76	6.04	119	102	4.67	7.44	89	156	2.40	4.19	119	78	2.37	4.52	64	78
$n = 8$	6.01	10.32	223	200	17.77	39.02	167	304	4.74	8.61	223	152	4.70	12.01	118	152
$n = 10$	20.07	40.55	359	330	66.98	131.23	269	500	14.63	48.00	359	250	14.77	48.30	188	250
$n = 12$	45.70	75.60	527	492	174.32	325.75	395	744	32.36	95.61	527	372	32.18	58.67	274	372
$n = 14$	97.41	161.50	727	686	448.11	695.54	545	1036	67.22	116.83	727	518	67.03	111.76	376	518

plexity (m, n, ℓ) increases, some numerical problems were encountered when constructing the matrix Γ in (5). Specifically, taking the first p rows of the matrix R_d can sometimes result in a matrix Γ with a very large condition number, which might lead to errors in computing $P = \text{null}(\Gamma)$. One explanation is that the kernel representation specified by the rows of R_d is not necessarily minimal. In a minimal kernel representation, the number of rows of R_d is equal to the number of outputs p (cf. [4]), which is the number of rows that are shifted in (5). Since obtaining the minimal kernel representation for multivariable systems is a difficult problem (cf. [22]), we devised a (heuristic) combinatorial method to obtain a well-conditioned matrix Γ . The idea is to try different combinations of p rows of R_d and use them to construct the shifts in Γ and later check if the condition number is smaller than a specified threshold. Note that these steps are done offline and do not increase the computational burden of the eDDPC scheme.

V. CONCLUSIONS

In this paper, we proposed a data-driven predictive control scheme which is both more sample-efficient (uses less offline data) and computationally efficient (uses less decision variables) than existing data-driven predictive control schemes in the literature. We analytically and numerically showed that the proposed scheme outperforms existing schemes from the literature. Future work will focus on providing theoretical guarantees for recursive feasibility and stability of a robust eDDPC scheme in presence of noise.

REFERENCES

- [1] J. Rawlings, D. Mayne, and M. Diehl, *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, 2017.
- [2] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, "Learning-based model predictive control: Toward safe learning in control," *Annu. rev. control robot.*, vol. 3, pp. 269–296, 2020.
- [3] I. Markovsky and F. Dörfler, "Behavioral systems theory in data-driven analysis, signal processing, and control," *Annual Reviews in Control*, vol. 52, pp. 42–64, 2021.
- [4] J. C. Willems, "From time series to linear system—part I: Finite dimensional linear time invariant systems, part II: Exact modelling, part III: Approximate modelling," *Automatica*, vol. 22, no. 5, pp. 561–580, 675–694, 87–115, 1986.
- [5] J. C. Willems, P. Rapisarda, I. Markovsky, and B. L. De Moor, "A note on persistency of excitation," *Systems & Control Letters*, vol. 54, no. 4, pp. 325–329, 2005.
- [6] T. Martin, T. B. Schön, and F. Allgöwer, "Guarantees for data-driven control of nonlinear systems using semidefinite programming: A survey," *Annual Reviews in Control*, p. 100911, 2023.

- [7] H. Yang and S. Li, "A data-driven predictive controller design based on reduced Hankel matrix," in *Asian Control Conf. (ASCC)*, 2015, pp. 1–7.
- [8] J. Coulson, J. Lygeros, and F. Dörfler, "Data-enabled predictive control: In the shallows of the DeePC," in *2019 18th European Control Conference*, 2019, pp. 307–312.
- [9] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "Data-driven model predictive control with stability and robustness guarantees," *IEEE Tran. Automat. Contr.*, vol. 66, no. 4, pp. 1702–1717, 2021.
- [10] —, "Stability in data-driven MPC: an inherent robustness perspective," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, 2022, pp. 1105–1110.
- [11] —, "Linear tracking MPC for nonlinear systems—part II: The data-driven case," *IEEE Tran. Automat. Contr.*, vol. 67, no. 9, pp. 4406–4421, 2022.
- [12] M. Alsalti, V. G. Lopez, J. Berberich, F. Allgöwer, and M. A. Müller, "Data-driven nonlinear predictive control for feedback linearizable systems," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 617–624, 2023.
- [13] C. Verhoek, H. S. Abbas, R. Tóth, and S. Haesaert, "Data-driven predictive control for linear parameter-varying systems," *IFAC-PapersOnLine*, vol. 54, no. 8, pp. 101–108, 2021.
- [14] Y. Lian and C. N. Jones, "Nonlinear data-enabled prediction and control," in *Proceedings of the 3rd Conference on Learning for Dynamics and Control*, vol. 144. PMLR, 07 – 08 June 2021, pp. 523–534.
- [15] G. Pan, R. Ou, and T. Faulwasser, "On a stochastic fundamental lemma and its use for data-driven optimal control," *IEEE Tran. Automat. Contr.*, vol. 68, no. 10, pp. 5922–5937, 2023.
- [16] V. Breschi, A. Chiuso, and S. Formentin, "Data-driven predictive control in a stochastic setting: a unified framework," *Automatica*, vol. 152, p. 110961, 2023.
- [17] E. O'Dwyer, E. C. Kerrigan, P. Falugi, M. Zagorowska, and N. Shah, "Data-driven predictive control with improved performance using segmented trajectories," *IEEE Transactions on Control Systems Technology*, vol. 31, no. 3, pp. 1355–1365, 2023.
- [18] M. Köhler, J. Berberich, M. A. Müller, and F. Allgöwer, "Data-driven distributed MPC of dynamically coupled linear systems," *IFAC-PapersOnLine*, vol. 55, no. 30, pp. 365–370, 2022.
- [19] A. Allibhoy and J. Cortés, "Data-based receding horizon control of linear network systems," *IEEE Contr. Syst. Lett.*, vol. 5, no. 4, pp. 1207–1212, 2021.
- [20] M. Lazar and P. C. N. Verheijen, "Generalized data-driven predictive control: Merging subspace and hankel predictors," *Mathematics*, vol. 11, no. 9, 2023.
- [21] M. Alsalti, I. Markovsky, V. G. Lopez, and M. A. Müller, "Data-based system representations from irregularly measured data," *arXiv: 2307.11589*, 2023.
- [22] I. Markovsky and F. Dörfler, "Identifiability in the behavioral setting," *IEEE Tran. Automat. Contr.*, vol. 68, no. 3, pp. 1667–1677, 2023.
- [23] K. Zhang, Y. Zheng, C. Shang, and Z. Li, "Dimension reduction for efficient data-enabled predictive control," *IEEE Contr. Syst. Lett.*, 2023.
- [24] F. Fiedler and S. Lucia, "On the relationship between data-enabled predictive control and subspace predictive control," in *2021 European Control Conference (ECC)*, 2021, pp. 222–229.
- [25] W. Favoreel, B. D. Moor, and M. Gevers, "SPC: Subspace predictive control," *IFAC Proceedings Volumes*, vol. 32, no. 2, pp. 4004–4009, 1999, 14th IFAC World Congress 1999, Beijing, Chia, 5-9 July.
- [26] <https://doi.org/10.24433/CO.0188721.v1>.