2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

Efficient Recursive Data-enabled Predictive Control

Jicheng Shi, Yingzhao Lian and Colin N. Jones

Abstract— **In the field of model predictive control, Data-enabled Predictive Control (DeePC) offers direct predictive control, bypassing traditional modeling. However, challenges emerge with increased computational demand due to recursive data updates. This paper introduces a novel recursive updating algorithm for DeePC. It emphasizes the use of Singular Value Decomposition (SVD) for efficient low-dimensional transformations of DeePC in its general form, as well as a fast SVD update scheme. Importantly, our proposed algorithm is highly flexible due to its reliance on the general form of DeePC, which is demonstrated to encompass various data-driven methods that utilize Pseudoinverse and Hankel matrices. This is exemplified through a comparison to Subspace Predictive Control, where the algorithm achieves asymptotically consistent prediction for stochastic linear time-invariant systems. Our proposed methodologies' efficacy is validated through simulation studies.**

I. Introduction

In Model Predictive Control (MPC), data-driven techniques have emerged as promising tools to expedite and enhance controller design, offering end-to-end solutions from input-output (I/O) data to fully functional controllers. Among these, Data-enabled Predictive Controller (DeePC) has gained significant attention, leveraging Willems' Fundamental Lemma [1] to bypass traditional modeling steps and establish a direct predictive controller. This method has demonstrated effectiveness across diverse domains, including batteries [2], [3], buildings [4], [5], grids [6], and vehicles [7].

In deterministic Linear Time-invariant (LTI) systems, numerous data-driven approaches have demonstrated the capability to consistently estimate system dynamics using a finite yet sufficiently excited I/O dataset. A paradigmatic method in this regard, Subspace Identification (SID) [8], employs an indirect approach to generate a consistent state-space model, facilitating the subsequent design of an MPC controller. In addition, other direct methods such as DeePC and Subspace Predictive Control (SPC) [9] can leverage this limited data set to directly yield exact trajectory prediction.

The landscape changes slightly for LTI systems affected by stochastic noise. Several data-driven studies have shown that employing infinite open-loop I/O data leads to the estimation of asymptotically consistent models [10]. Building on this foundation, more recent efforts have sought to extend these algorithms to closed-loop data [11]. An innovative proposal in this context, as mentioned in [12], seeks to design the SPC using initial I/O data for system control, with the SPC undergoing recursive updates to enhance performance. DeePC has also witnessed similar extensions, such as the integration of instrumental variables [13], [14], [15], which have also been explored to achieve consistency in predictions utilizing both open [13] and closed-loop data [14].

A major hurdle arises with DeePC's increasing computational complexity as more I/O data are integrated. Recent studies have aimed to mitigate this by reducing DeePC's computational overhead [3], [16], [17]. For instance, [3], [16] employ the Singular Value Decomposition (SVD) of the Hankel matrix to reduce the dimensions of DeePC's decision variables. However, these methods, while promising, often present their own challenges, especially as more data is recursively incorporated into the model. Existing recursive updating methodologies in SID [18] and SPC [12], [19], reliant on the least squares structure, remain unsuited to DeePC and its variations. This gap underscores the pressing need for a generalized and efficient strategy to recursively update DeePC.

Addressing this void, our research introduces an effective recursive updating paradigm within the DeePC framework. Our main contribution describes the equivalency between an SVD-based low-dimensional DeePC and its counterpart in a more general form compared to [3], [16], while also detailing an efficient SVD updating mechanism for recursively updated I/O data. A key advantage of the proposed algorithm is its high degree of flexibility rooted in the general-form DeePC. Our study demonstrates that this form of DeePC can include data-driven methods based on Pseudoinverse and Hankel matrices. We give an example of this through a comparison to SPC, where the algorithm achieves asymptotically consistent prediction. In addition, our proposed algorithm has the potential for broader applications, especially among other adaptive DeePC methods [20], [21].

The paper's structure is as follows: Section II revisits Willems' fundamental lemma and DeePC. Subsequently, Section III delves into the equivalent low-dimensional transformation of DeePC, introducing our efficient recursive updating method. Section IV details an extension to data-driven methods based on Pseudoinverse. The validity of these methods is empirically established through simulations presented in Section V.

Notation: Let $\mathbf{0}$ represent a zero matrix, and I represent an identity matrix. The notation $x := \{x_i\}_{i=1}^{T}$ indicates a sequence of size T. The term x_t represents the measurement of x at the instance t. Additionally, $x_{1:L} := [x_1^\top, x_2^\top, \ldots, x_L^\top]^\top$ signifies a concatenated sequence of x from x_1 to x_L. $M^\dagger$ indicates Moore–Penrose inverse of a matrix M.

This work received support from the Swiss National Science Foundation (SNSF) under the NCCR Automation project, grant agreement 51NF40_180545. (corresponding author: Jicheng Shi)

Extended version: doi.org/10.48550/arXiv.2309.13755

JS, YL and CNJ are with Automatic Laboratory, EPFL, 1015 Lausanne, Switzerland. {jicheng.shi, yingzhao.lian, colin.jones}@epfl.ch

Copyright ©2024 EUCA

II. Preliminaries

Consider a linear time-invariant (LTI) system described by the equations $x_{t+1} = Ax_t + Bu_t$ and $y_t = Cx_t + Du_t$, which we refer to as $\mathfrak{B}(A, B, C, D)$. The system's order is given by n_x with n_u, n_y denoting its input and output dimensions. An L-step trajectory for this system is represented as $\begin{bmatrix} u_{1:L}^\top & y_{1:L}^\top \end{bmatrix}^\top$. The set of all potential L-step trajectories produced by $\mathfrak{B}(A, B, C, D)$ is denoted by $\mathfrak{B}_L(A, B, C, D)$. We define the Hankel matrix H_s of depth L associated with a vector-valued signal sequence $s = \{s_i\}_{i=1}^T$ as:

$$H_s := \begin{bmatrix} s_1 & s_2 & \dots & s_{T-L+1} \\ s_2 & s_3 & \dots & s_{T-L+2} \\ \vdots & \vdots & & \vdots \\ s_L & s_{L+1} & \dots & s_T \end{bmatrix}.$$

The row and column counts of a Hankel matrix H are given by row_H and col_H, respectively. Throughout this paper, the term L is exclusively used to indicate the size of the Hankel matrix. An input measurement sequence defined as $u = \{u_i\}_{i=1}^T$ is termed *persistently exciting* of order L if the Hankel matrix H_u has full row rank.

Utilizing the Hankel matrices H_u and H_y, we introduce the well-established **Willems' Fundamental Lemma**:

Lemma 1: [1, Theorem 1] Given a controllable linear system where $\{u_i\}_{i=1}^T$ is persistently exciting of order $L+n_x$, the condition colspan$(\begin{bmatrix} H_u^\top & H_y^\top \end{bmatrix}^\top) = \mathfrak{B}_L(A, B, C, D)$ is satisfied.

Recent advancements in the data-driven control domain have given rise to schemes like DeePC [22], along with numerous variants, for instance, [23], [20]. Lemma 1 plays a pivotal role in these schemes by facilitating trajectory prediction. In the scope of this paper, our primary aim is to unveil a universally efficient updating algorithm tailored for various controllers under the DeePC paradigm. To exemplify, consider the L2 regularized DeePC (L2-DeePC) detailed in [23]:

$$\min_{\substack{g,\sigma \\ u_{pred}, y_{pred}}} \quad J(u_{pred}, y_{pred}) + \lambda_\sigma \|\sigma\|_2^2 + \lambda_g \|g\|_2^2 \tag{1a}$$

$$\text{s.t. } Hg = \begin{bmatrix} y_{init} + \sigma \\ y_{pred} \\ u_{init} \\ u_{pred} \end{bmatrix}, \tag{1b}$$

$$y_{pred} \in \mathbb{Y}, u_{pred} \in \mathbb{U} \tag{1c}$$

where $H := \begin{bmatrix} H_y \\ H_u \end{bmatrix}$ for simplification. The parameters λ_σ and λ_g represent user-determined regularization cost weights. The elements $J(u_{pred}, y_{pred})$, $\mathbb{Y}$, and $\mathbb{U}$ are defined according to the task at hand. Sequences u_{init} and y_{init} provide n_{init}-step historical data for measured inputs and outputs leading up to the present moment, which aids in current state estimation of the dynamic system [22]. Correspondingly, u_{pred} and y_{pred} denote the predicted sequences of n_{pred} steps from the current timestamp. Consistently, the row dimension of the Hankel matrix is set to $L = n_{init} + n_{pred}$.

The L2-DeePC as presented in (1) forecasts the n_{pred}-step output trajectory y_{pred} based on a provided predictive input sequence u_{pred}. The objective, specified in (1a), is minimized subject to the constraint delineated in (1c). The inclusion of the slack variable σ ensures feasibility for L2-DeePC. Meanwhile, regularization terms are introduced to enhance predictions, especially beneficial when the system is prone to noise or embodies nonlinear elements. For an in-depth discussion and detailed insights, readers are directed to [23].

This paper introduces a data-driven MPC technique under the DeePC framework that is recursively updated with the most recent operational data. We term this approach recursive DeePC and detail it in Algorithm 1.

Algorithm 1 Recursive DeePC

0) Retrieve some persistently excited past I/O data and build the initial DeePC controller, such as (1).
1) Retrieve the recent L-step measurements and update the Hankel matrix as:
$$H \leftarrow \begin{bmatrix} H & \begin{bmatrix} y_{t-L:t-1} \\ u_{t-L:t-1} \end{bmatrix} \end{bmatrix} \tag{2}$$
2) Retrieve the recent t_{init}-step measurements. Solve the DeePC and apply the optimal input as $u_t = u_{pred}^*(1)$.
3) Pause until the subsequent sampling time, update $t \leftarrow t+1$ and revert to step 1.

$u_{pred}^*(1)$ represents the first optimal input in u_{pred}^*

In many applications, empirical evidence suggests that DeePC benefits from larger quantities of I/O data beyond the minimal requirement [4], [6]. Furthermore, research in [13] establishes that infinite open-loop data can ensure consistent prediction in DeePC methods with instrumental variables for stochastic LTI systems. This insight can be broadened to closed-loop data, leveraging techniques from SID [8] and SPC [12]. Notably, by modifying (2) to incorporate a forgetting factor or discard outdated data, Algorithm 1 can be adapted for adaptive DeePC approaches such as those detailed in [20], [21].

III. Efficient recursive updates in the DeePC framework

In this section, we introduce a more computationally efficient version of Algorithm 1 when the DeePC used is in some general form. This improved algorithm relies on two primary components: (1) an equivalent low-dimensional transformation of the DeePC in the general form leveraging SVD, and (2) a fast SVD updating technique. The complete methodology is concluded in Algorithm 3. To detail its operation, we reference the L2-DeePC (1) as a demonstrative example.

A. An equivalent low-dimensional transformation

For the first component of Algorithm 3, we describe an equivalent low-dimensional transformation of a general DeePC problem. This transformation is facilitated by the SVD of the aggregated Hankel matrix, H:

$$H = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} V_1 & V_2 \end{bmatrix}^\top = U_1 \Sigma V_1^\top$$

where $\Sigma \in \mathbb{R}_{r_H, r_H}$ and r_H is the rank of H. A general DeePC problem is defined as:

Problem 1:

$$\min_{\substack{g,\sigma \\ u_{pred},y_{pred}}} \quad f_1(u_{pred}, y_{pred}, \sigma, V_1^\top g) + f_2(V_2^\top g)$$
$$\text{s.t. } Hg = \begin{bmatrix} y_{init} \\ y_{pred} \\ u_{init} \\ u_{pred} \end{bmatrix} + \sigma, \tag{3}$$
$$f_3(y_{pred}, u_{pred}, \sigma) \leq 0$$

Here, functions $f_1(\cdot)$, $f_2(\cdot)$, and $f_3(\cdot)$ are user-specified and vary across different DeePC methodologies tailored for diverse applications, which can cover the L2-DeePC and various DeePC variants [23], [6], [20]. The aforementioned transformation in a lower dimension is defined as:

Problem 2:

$$\min_{\substack{\bar{g},\sigma \\ u_{pred},y_{pred}}} \quad f_1(u_{pred}, y_{pred}, \sigma, \bar{g})$$
$$\text{s.t. } \bar{H}\bar{g} = \begin{bmatrix} y_{init} \\ y_{pred} \\ u_{init} \\ u_{pred} \end{bmatrix} + \sigma, \tag{4}$$
$$f_3(y_{pred}, u_{pred}, \sigma) \leq 0$$

where $\bar{H} := U_1\Sigma$, signifying the transformed version of the Hankel matrix.

***Lemma** 2:* Problem 1 and Problem 2 are equivalent. The optimal values of y_{pred}, u_{pred} for Problem 1 are also optimal for Problem 2, and vice versa.

Proof: Problem 1 can change the decision variable g by:

$$\tilde{g} = \begin{bmatrix} \tilde{g}_1 \\ \tilde{g}_2 \end{bmatrix} = \begin{bmatrix} V_1^\top g \\ V_2^\top g \end{bmatrix} = \begin{bmatrix} V_1 & V_2 \end{bmatrix}^\top g$$

because $\begin{bmatrix} V_1 & V_2 \end{bmatrix}$ is an orthogonal matrix [24]. Then because $Hg = U_1\Sigma V_1^\top g = \bar{H}\tilde{g}_1$, the objects and constraints in the new equivalent problem are separable with respect to $\tilde{g}_1$ and $\tilde{g}_2$:

$$\min_{\substack{\tilde{g},\sigma \\ u_{pred},y_{pred}}} \quad f_1(u_{pred}, y_{pred}, \sigma, \tilde{g}_1) + f_2(\tilde{g}_2)$$
$$\text{s.t. } \bar{H}\tilde{g}_1 = \begin{bmatrix} y_{init} \\ y_{pred} \\ u_{init} \\ u_{pred} \end{bmatrix} + \sigma,$$
$$f_3(y_{pred}, u_{pred}, \sigma) \leq 0$$

Therefore, we can solve them separately by:

$$\min_{\substack{\tilde{g}_1,\sigma \\ u_{pred},y_{pred}}} \quad f_1(u_{pred}, y_{pred}, \sigma, \tilde{g}_1)$$
$$\text{s.t. } \bar{H}\tilde{g}_1 = \begin{bmatrix} y_{init} \\ y_{pred} \\ u_{init} \\ u_{pred} \end{bmatrix} + \sigma,$$
$$f_3(y_{pred}, u_{pred}, \sigma) \leq 0$$
$$\min_{\tilde{g}_2} f_2(\tilde{g}_2)$$

By replacing $\tilde{g}_1$ by $\bar{g}$ in the first sub-problem above, we get Problem 2. ■

Leveraging Lemma 2, we can deduce the low-dimensional version of the L2-DeePC (1). This inference is drawn from the relationship: $\|g\|_2^2 = g^\top \begin{bmatrix} V_1 & V_2 \end{bmatrix} \begin{bmatrix} V_1 & V_2 \end{bmatrix}^\top g = \|V_1 g\|_2^2 + \|V_2 g\|_2^2$:

$$\min_{\substack{\bar{g},\sigma \\ u_{pred},y_{pred}}} \quad J(u_{pred}, y_{pred}) + \lambda_\sigma \|\sigma\|_2^2 + \lambda_g \|\bar{g}\|_2^2$$
$$\text{s.t. } \bar{H}\bar{g} = \begin{bmatrix} y_{init} + \sigma \\ y_{pred} \\ u_{init} \\ u_{pred} \end{bmatrix}, \tag{5}$$
$$y_{pred} \in \mathbb{Y}, u_{pred} \in \mathbb{U}$$

***Remark** 1:* In Problem 2, the decision variable $\bar{g}$, which belongs to $\mathbb{R}_{r_H}$, is independent of the columns of the Hankel matrix. The authors in [3], [16] introduce the same SVD-based transformation and establish the equivalence using KKT conditions [16]. However, their study is limited to L2-DeePC, which is a special case of the more general form of DeePC (3) described in our study. .

B. Efficient SVD updates

In the preceding section, we established that the general-form DeePC (3) can be converted into a more compact, low-dimensional format (4) via SVD. Notably, the dimensionality of the decision variable in (4) is governed solely by the rank of the Hankel matrix H.

Expanding upon this, the current section introduces a rapid SVD updating technique [25], [26]. This method obviates the need for a complete SVD recalculation with each recursive update (2). When the previous SVD components, specifically U_1 and Σ, are available and H undergoes an update as per (2), Algorithm 2 can be leveraged to update the new U_1 and Σ.

Algorithm 2 Fast SVD updating

Given: Current SVD components: U_1, Σ

1) Retrieve the column a to be added to H, i.e. $\begin{bmatrix} y_{t-L:t-1} \\ u_{t-L:t-1} \end{bmatrix}$ at time t. Compute $r = \text{rank}(\Sigma)$.
2) If $r < row_H$, update U_1, Σ by [25]:
   Compute $m = U^\top a$, $p = a - Um$, $R_a = \|p\|_2$, $P = R_a^{-1}p$
   Compute $K = \begin{bmatrix} \Sigma & m \\ 0 & R_a \end{bmatrix}$ and its SVD: $K = C\bar{\Sigma}D^\top$
   If $\text{rank}(\bar{\Sigma}) == r$:
   $U_1 \leftarrow \begin{bmatrix} U_1 & P \end{bmatrix} C(:, 1:r)$, $\Sigma \leftarrow \bar{\Sigma}(1:r, 1:r)$
   If $\text{rank}(\bar{\Sigma}) == r+1$:
   $U_1 \leftarrow \begin{bmatrix} U_1 & P \end{bmatrix} C$, $\Sigma \leftarrow \bar{\Sigma}$
3) If $r == row_H$, update U_1, Σ by [26]:
   Compute $z = U_1^\top a$
   Compute eigendecomposition of $\Sigma^2 + zz^\top$: $C\bar{\Sigma}C^\top$
   Update $U_1 \leftarrow U_1 C$, $\Sigma \leftarrow \sqrt{\bar{\Sigma}}$

***Lemma** 3:* Algorithm 2 calculates U_1 and Σ identical to the results obtained through the direct SVD of H following each recursive update (2). It boasts a computational complexity of $\mathcal{O}(row_H r_H^2)$ and a space requirement of $\mathcal{O}(row_H r_H)$.

Proof: Proofs of equivalence for the conditions in steps 2) and 3) of Algorithm 2 are provided in [25], [26], which is omitted due to the limited space. All the matrix

multiplications requiries a load of $\mathcal{O}(row_H r_H^2)$. In addition, the SVD's load in step 2) is $\mathcal{O}(r_H^3)$ (or $\mathcal{O}(r_H^2)$ due to the special structure [25]), and the eigendecomposition's load is $\mathcal{O}(row_H^3)$. Because $r_H < row_H$ for step 2) and $r_H = row_H$ for step 3), the overall complexity is $\mathcal{O}(row_H r_H^2)$. Finally, the matrices' size directly determines the space requirement. ■

C. Conclusion of the algorithm

A computationally efficient recursive DeePC is summarized in Algorithm 3. The general form of DeePC represented in (3) undergoes a transformation into a low-dimensional equivalent as outlined in (4), using SVD. Moreover, with each successive update as indicated in (2), the new SVD components are rapidly updated.

Algorithm 3 Efficient Recursive DeePC

0) Retrieve some persistently excited past I/O data. Construct H and compute its SVD. Build the initial low-dimensional DeePC controller based on the Problem 2, such as (5).
1) Retrieve the recent L-step measurements. Update the SVD components based on Algorithm 2 and update $\bar{H}$.
2) Retrieve the recent t_{init}-step measurements. Solve the low-dimensional DeePC and apply the optimal input as $u_t = u^*_{pred}(1)$.
3) Pause until the subsequent sampling time, update $t \leftarrow t+1$ and revert to step 1.

Lemma 4: If the DeePC in the form of Problem 1 is used in Algorithm 1, then Algorithm 1 and Algorithm 3 are equivalent.

Proof: At the initial step, Problem 1 and Problem 2 are respectively constructed in two Algorithms, which have been proved to be equivalent in Lemma 2. After the first recursive update (2), the new SVD components are exactly updated by Algorithm 2 proved in Lemma 3. Therefore, Problem 1 and Problem 2 are still equivalent by Lemma 2. The proof is then completed by induction. ■

As a result, the total computational complexity depends mainly on row_H and r_H. Because $row_H = (n_{init} + n_{pred})(n_u + n_y)$, and the decision variables $\bar{g} \in \mathbb{R}_{r_H}$, $u_{pred} \in \mathbb{R}_{n_u(n_{init}+n_{pred})}$, $y_{pred} \in \mathbb{R}_{n_y(n_{init}+n_{pred})}$, $\sigma \in \mathbb{R}_{row_H}$ in Problem 1. Besides, the complexity of the fast SVD updating method is proved in Lemma 3. Therefore, the complexity is fixed after the DeePC's parameter is settled as $r_H \leq row_H$. It's notable that the size of the original recursive DeePC in Algorithm 1 relates to $col_H = T - row_H + 1$, which increases with the addition of more data.

The computational burden of Algorithm 3 is comparable to the recursive SPC method [12]. The latter has a decision variable in its sparse representation of size row_H, which can recursive update using *Recursive Least Square* at a computational complexity of $\mathcal{O}(row_H^2)$[19]. In the next Section, we will prove that SPC is equivalent to a specialized DeePC belonging to the general-from DeePC (3), adaptable as well by Algorithm 3.

Remark 2: Algorithm 3 offers extensions to other adaptive DeePC strategies, typified by references like [20], [21]. These strategies are applicable to slowly time-varying linear systems or approximate dynamics of unknown nonlinear systems across varied operating points. Other fast SVD modifications, such as the integration of forgetting factors and downdating [27], can be incorporated for extensions to these adaptive methods (see Extended Version: Appendix C).

IV. An Extension to Data-driven Methods Based-on Pseudoinverse

A pivotal strength of Algorithm 3 is its versatile nature, rooted in its generic DeePC formulation. Beyond encompassing various DeePC variants [23], [6], [20], it holds potential for extension to various data-driven methodologies that utilize the Hankel matrix, such as simulation [28], physics-based filters [29] and data-driven observers [30], [31].

Among them, A group of researchers utilizes Pseudoinverse to achieve prediction or estimation [6], [28], [30], [31]. For the purpose of this section, we will illustrate that these Pseudoinverse-based methods can be generalized in the form of Problem 1 using a specific data-driven prediction formulation [6]. Next, we will demonstrate that Problem 1 can generalize SPC, which also employs Pseudoinverse and Hankel matrices. Additionally, we present how to achieve asymptotically consistent prediction for stochastic LTI systems through recursive data updates using Algorithm 3.

A. Comparison to Data-driven prediction

Given measured u_{init}, y_{init} and required u_{pred}, a data-driven prediction method based on Pseudoinverse [6] is formulated as:

$$\begin{aligned} y_{pred} &= H_{y,pred} g \\ g &= \begin{bmatrix} H_{y,init} \\ H_u \end{bmatrix}^{\dagger} \begin{bmatrix} y_{init} \\ u_{init} \\ u_{pred} \end{bmatrix} \end{aligned} \tag{6}$$

where the sub-Hankel matrices are derived from the original Hankel matrix by $H_y = \begin{bmatrix} H_{y,init} \\ H_{y,pred} \end{bmatrix}$. The matrix $H_{y,init}$ is of depth n_{init} and the depth of $H_{y,pred}$ is the prediction horizon n_{pred} such that $n_{init} + n_{pred} = L$.

Consider an optimization problem:

$$\begin{aligned} (y_{pred}, g) &= \underset{y_{pred,l}, g_l}{\arg\min} \|g_l\|_2^2 \\ \text{s.t. } H g_l &= \begin{bmatrix} y_{init} \\ y_{pred,l} \\ u_{init} \\ u_{pred} \end{bmatrix} \end{aligned} \tag{7}$$

Assumption 1: The problem (7) is feasible.

Lemma 5: Under Assumption 1, (7) computes the same unique optimal solution as (6).

Proof: Consider a problem:

$$\begin{aligned} y_{pred} &= H_{y,pred} g \\ g &= \underset{g_l}{\arg\min} \|g_l\|^2 \\ \text{s.t. } & \begin{bmatrix} H_{y,init} \\ H_u \end{bmatrix} g_l = \begin{bmatrix} y_{init} \\ u_{init} \\ u_{pred} \end{bmatrix} \end{aligned}$$

Under Assumption 1, the problem remains feasible. Given that the Pseudoinverse offers a unique solution with the minimal L2-norm for least squares problems [32], this problem yields the same prediction as (6). The proof concludes with the observation that this problem and (7) can be equivalently expressed by the same linear equations when applying the KKT conditions. ■

By the fact $\|g\|_2^2 = g^\top \begin{bmatrix} V_1 & V_2 \end{bmatrix} \begin{bmatrix} V_1 & V_2 \end{bmatrix}^\top g = \|V_1 g\|_2^2 + \|V_2 g\|_2^2$ and adding an equality constraint so that u_{pred} is equal to the required value, we can write (7) in the form of Problem 1.

***Remark** 3:* When the input sequence for H_u is persistently excited, Assumption 1 is considered mild. Because then the assumption is satisfied in noiseless conditions as confirmed by Lemma 1. In the presence of noise, this assumption also holds since Hankel matrices are almost surely of full row rank. Alternatively, Extended Version: Appendix D presents some optimization formulations and their low-dimensional equivalents, which are always feasible and provide the same solution as (6). For other Pseudoinverse-based data-driven methods [28], [30], [31], similar results can be derived after little modification of (6).

B. Comparison to SPC

This section describes how to generalize SPC in the form of Problem 1. In addition, Algorithm 3 helps to achieve asymptotically consistent prediction by continuously involving open-loop and closed-loop data online. The SPC controller [9] is formulated as,

$$\begin{aligned} &\min_{u_{pred}, y_{pred}} J(u_{pred}, y_{pred}) \\ &\text{s.t. } y_{pred} \in \mathbb{Y}, u_{pred} \in \mathbb{U} \\ &y_{pred} = K \begin{bmatrix} y_{init} \\ u_{init} \\ u_{pred} \end{bmatrix} \qquad (8a) \\ &K = H_{y,pred} \begin{bmatrix} H_{y,init} \\ H_u \end{bmatrix}^\dagger \qquad (8b) \end{aligned}$$

Based on the specific-form data-driven prediction (7), a bi-level DeePC is defined as:

$$\begin{aligned} &\min_{\substack{g, u_{pred} \\ y_{pred}}} J(u_{pred}, y_{pred}) \\ &\text{s.t. } y_{pred} \in \mathbb{Y}, u_{pred} \in \mathbb{U} \\ &\quad (7) \end{aligned} \tag{9}$$

***Lemma** 6:* Under Assumption 1, SPC (8) and the DeePC (9) are equivalent.

Proof: The only difference between the two controllers is their prediction parts, i.e. (8a) and (8b), (7). The fact that both predictions can be written in the same explicit form finishes the proof: $y_{pred} = H_{y,pred} \begin{bmatrix} H_{y,init} \\ H_u \end{bmatrix}^\dagger \begin{bmatrix} y_{init} \\ u_{init} \\ u_{pred} \end{bmatrix}$ ■

***Remark** 4:* A bi-level DeePC that is equivalent to the SPC (8), without requiring any assumption, is detailed in the Extended Version: Appendix E due to limited space. In addition, some low-dimensional equivalents and an efficient recursive SPC algorithm are introduced.

The analysis of consistent prediction is as follows. Consider a stochastic LTI system defined in innovation form:

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t + Ke_t \\ y_t &= Cx_t + Du_t + e_t \end{aligned} \tag{10}$$

where K denotes the Kalman gain and e_k is a zero-mean white noise signal. The prediction y_{pred} at time t is consistent if its expectation is an unbiased estimation of the real output sequence y_{real} [14], [8], i.e.

$$\mathbb{E}_e(y_{pred} - y_{real}) = \mathbf{0}$$

***Assumption** 2:* The Kalman gain K in (10) ensures that the matrix $A - KC$ is strictly stable. The initial step n_{init} is sufficiently large so that $(A - KC)^{t_{init}} \approx 0$.

***Assumption** 3:* The input is quasi-stationary so that limits of time averages of the input sequence exists.

***Assumption** 4:* The input sequence $\{u_i\}_{i=1}^T$ for Hankel matric H_u is persistently exciting of order $L + n_x$.

***Lemma** 7:* Under Assumptions 2, 3 and 4, (7) constructed by open-loop data provides a consistent prediction when $col_H \to \infty$.

***Lemma** 8:* Assumptions 2, 3 and 4 stand. Assume that $D = 0$ in the LTI system or the I/O data is collected by feedback control with at least one sample time delay. Then (7) with $n_{pred} = 1$ constructed by the closed-loop data provides a consistent prediction when $col_H \to \infty$

The proof of Lemmas 7 and 8 is elaborated in Appendix A. Assumptions 2, 3 and 4 used therein also frequently emerge in consistency analysis in the field of system identification, as seen in [8], [9]. In addition, because Lemma 8 guarantees asymptotically consistent prediction for (7) with closed-loop data when setting $n_{pred} = 1$, one can successively apply (7) with $n_{pred} = 1$ to achieve consistent multi-step output prediction via:

$$\begin{aligned} &\forall i = 1, 2, \ldots, n_{pred}: \\ &(y_{pred}(i), g_i) = \arg\min_{y_{pred,l}, g_l} \|g_l\|_2^2 \\ &\text{s.t. } Hg = \begin{bmatrix} y_{init}(i : n_{pred}) \\ y_{pred}(1 : i-1) \\ y_{pred,l} \\ u_{init}(i : n_{pred}) \\ u_{pred}(1 : i) \end{bmatrix} \end{aligned} \tag{11}$$

where $y_{pred}(i)$ represents the $i-$th output in y_{pred}, and $y_{pred}(i : j)$ captures the vector from the i-th to $j-$th outputs within y_{pred} (with similar notations applied elsewhere). Similar setups have been validated in prior DeePC and SPC studies [33], [12]. A new bi-level DeePC can be defined as:

$$\begin{aligned} &\min_{\substack{g, u_{pred} \\ y_{pred}}} J(u_{pred}, y_{pred}) \\ &\text{s.t. } y_{pred} \in \mathbb{Y}, u_{pred} \in \mathbb{U} \\ &\quad (11) \end{aligned} \tag{12}$$

Notably, one can directly apply Algorithm 3 to recursively update the bi-level DeePC (12). A tractable computation method for this bi-level DeePC is to transform the lower-level problem into linear constraints by KKT conditions. More details can be referred to our previous work [20]. According to Lemma 8, with an infinite length of closed-loop data,

it's feasible to obtain an unbiased output prediction for the stochastic LTI system. Nonetheless, it's important to note that there may not be a monotonic improvement in prediction and control performance throughout the update cycle.

In addition, one can adapt Algorithm 3 to update the data-driven prediction online in the two bi-level DeePCs (9) and (12) with data from other controllers. To achieve this, the modification required for (11) (or (7)) is simply replacing the DeePC in step 3) of Algorithm 2 with other closed-loop controllers (or open-loop control signals).

V. Simulation

In this section, we evaluate the effectiveness of the proposed efficient recursive DeePC methodology through simulation studies. We utilize a discrete-time LTI system, as detailed in [9], which models two circular plates coupled with flexible shafts. The system's matrices, conforming to (10), are provided:

$$A = \begin{bmatrix} 4.4 & 1 & 0 & 0 & 0 \\ -8.09 & 0 & 1 & 0 & 0 \\ 7.83 & 0 & 0 & 1 & 0 \\ -4 & 0 & 0 & 0 & 1 \\ 0.86 & 0 & 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0.00098 \\ 0.01299 \\ 0.01859 \\ 0.0033 \\ -0.00002 \end{bmatrix}, K = \begin{bmatrix} 2.3 \\ -6.64 \\ 7.515 \\ -4.0146 \\ 0.86336 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix}, D = 0$$

During the simulation, the noise variance is set to $\text{var}(e_t) = 0.1$, and the input is restricted to $|u_t| \leq 10$. The optimization problems in the following simulation are solved by the solver QUADPROG in MATLAB with Intel Core i7-1165G7 2.80 GHz processor.

A. Validation of Algorithm 3

To initiate, a 200-step trajectory is generated with the input defined as a zero-mean white noise signal, having $\text{var}(u_t) = 1$. Employing this initial trajectory, a L2-DeePC (1) is established, targeting the objective $J(u_{pred}, y_{pred}) = \|y_{pred} - ref\|_2^2 + 0.001\|u_{pred}\|_2^2$. Initially, the reference is set at 10 for 1000 steps and subsequently adjusted to 0 for the ensuing 1000 steps. The parameters are designated as $\lambda_\sigma = 10^6$, $\lambda_g = 10^4$, and $n_{init} = n_{pred} = 10$. These parameters aren't meticulously tuned, as our primary interest lies in evaluating the efficiency of Algorithm 3. The L2-DeePC controls the system and is recursively updated by Algorithm 1. For comparative analysis, an identical procedure is employed utilizing Algorithm 3, integrated with the equivalent low-dimensional L2-DeePC, as delineated in (5).

Table I provides the statistical results from 10 Monte Carlo simulations across different noise scenarios. Both algorithms yield almost identical input and output signals, with only slight numerical errors, Algorithm 3 proves to be faster in execution than Algorithm 1. For a closer look, Figure 1 shows the resulting trajectories for a specific noise scenario. The input and output trajectories validate the equivalence between the two algorithms. Additionally, we analyze the computational time required for each recursive update and optimization for both algorithms. We notice that the time for Algorithm 1 increases superlinearly as more data are added, whereas the time for Algorithm 3 stays relatively steady, highlighting its efficiency.

TABLE I: Statistical results of 10 Monte Carlo runs: the differences of input ($e_u = |u_1 - u_3|$) and output ($e_y = |y_1 - y_3|$)), the computational time of each recursive step ($time$), where $\cdot_1$ and $\cdot_3$ respectively indicate the data from Algorithm 1 and 3.

Average e_u	Average e_y	Average $time_1$	Average $time_3$
6.7×10^{-12}	5.2×10^{-12}	0.1557 [s]	0.0026 [s]

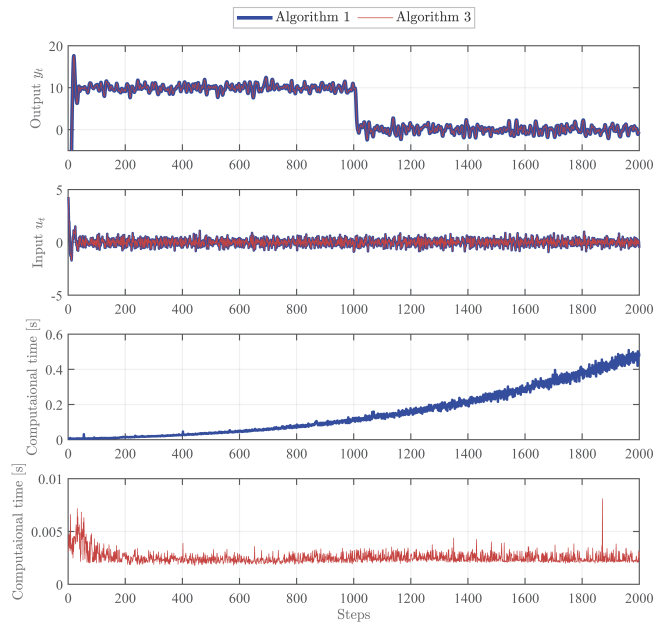

Fig. 1: Comparison of Algorithm 1 and Algorithm 3.

B. Comparison to SPC

This section evaluates the asymptotic consistency of the data-driven prediction in equations (9) and (12) by using Algorithm 3. We will refer them to as DDP1 and DDP2 for brevity. The equivalence between (9) and SPC (8) is also tested. Given that all the data-driven prediction methods and the ground truth (see the Extended Version: Appendix B) can be expressed in the matrix form:

$$y_{pred} = K_{y,init} y_{init} + K_{u,init} u_{init} + K_{u,pred} u_{pred} \quad (13)$$

, consistency is tested by comparing discrepancies among the involved matrices. In this study, we set $n_{init} = n_{pred} = 50$, with a large n_{init} ensuring compliance with Assumption 2

Firstly, an open-loop trajectory spanning 10000 steps is generated using an input characterized as a zero-mean white noise signal with a variance $\text{var}(u_t) = 1$. DPP1, DDP2, and SPC are initialized using 150 steps to assemble the Hankel matrix. They are then efficiently updated in a recursive manner, leveraging a variant of Algorithm 3 as outlined at the end of Section IV-B. Figure 2 depicts average outcomes from 10 Monte Carlo simulations, wherein the deviation from the ground truth is calculated at each iteration. As more open-loop data are incorporated into the three prediction methods, the matrix discrepancies consistently diminish, reinforcing their validity. Furthermore, the equivalence between SPC and DPP1 is validated.

The subsequent experiment employs a 25000-step closed-loop trajectory, controlled by a static DeePC constructed from the open-loop trajectory of the previous test. Average results over 10 random noise signal realisations are show-

cased in Figure 3. The matrix discrepancies from the ground truth, as observed in the SPC and DDP1, initially decline but later stabilize. Conversely, DDP2 continually exhibits a reduction in matrix differences as it integrates more closed-loop data. However, it's notable that when the Hankel matrix lacks sufficient data, matrix discrepancies in DDP2 exceed others', and its improvement rate lags behind that observed in Figure 2. Future work will focus on optimizing the closed-loop controller design to expedite improvements in DDP2.

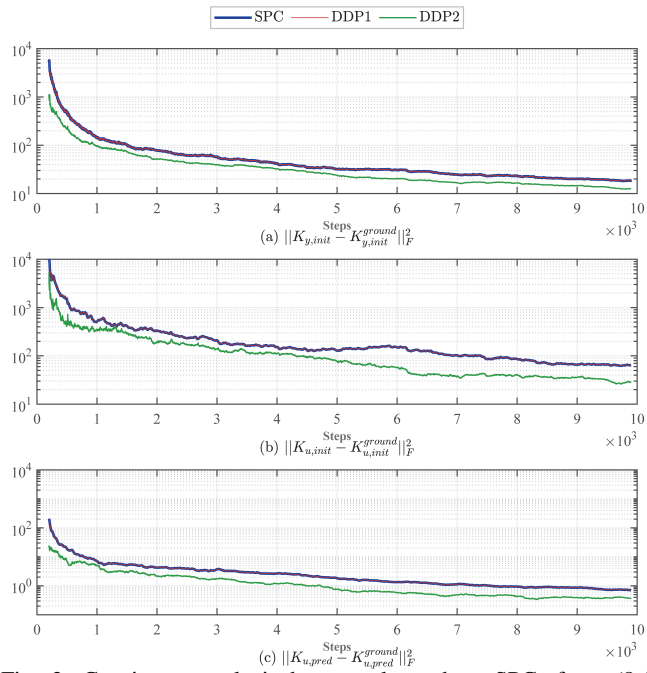

Fig. 2: Consisteny analysis by open-loop data. SPC: from (8a) and (8b); DDP1: from (9); DDP2: from (12). K^{ground} indicates the matrix from the ground truth.

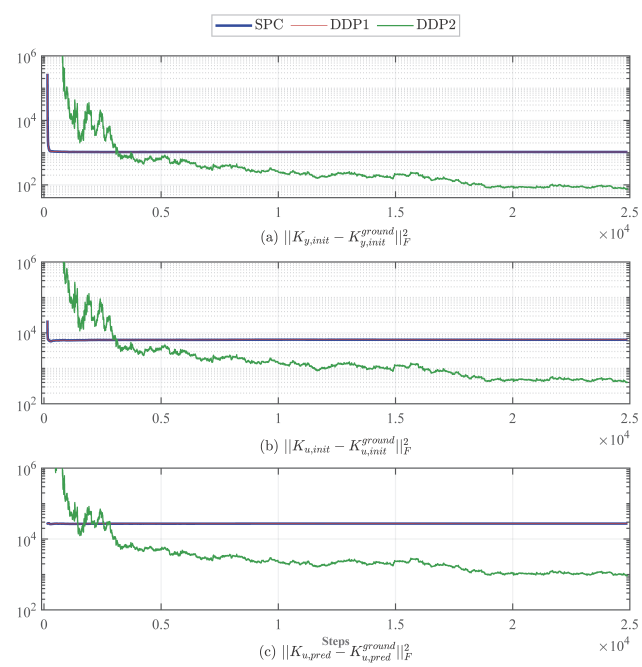

Fig. 3: Consisteny analysis by closed-loop data

VI. Conclusion

In conclusion, this paper presents a novel recursive updating algorithm for DeePC to efficiently handle computational challenges. The algorithm utilizes SVD for low-dimensional transformations and fast updates. It is flexible, accommodating various data-driven methods that use Pseudoinverse and Hankel matrices, as demonstrated through a comparison to Subspace Predictive Control.

Appendix

A. *Proofs of Lemmas 7 and 8*

First, we derive a relationship from the stochastic LTI system (10). By propagating the dynamics from time t, we can formulate the next n_{pred}-step output as:

$$y_{real} = \Gamma x_t + K_1 u_{pred} + K_2 e_{pred} \tag{14}$$

where $e_{pred} := e_{t:t+n_{pred}-1}$ and

$$\Gamma = \begin{bmatrix} C^\top & (CA)^\top & (CA^2)^\top & \cdots & (CA^{n_{pred}-1})^\top \end{bmatrix}^\top,$$

$$K_1 = \begin{bmatrix} D & 0 & 0 & \cdots & 0 \\ CB & D & 0 & \cdots & 0 \\ CAB & CB & D & \cdots & 0 \\ \cdots & \cdots & \ddots & \ddots & \vdots \\ CA^{n_{pred}-2}B & CA^{n_{pred}-3}B & \cdots & CB & D \end{bmatrix},$$

$$K_2 = \begin{bmatrix} I & 0 & 0 & \cdots & 0 \\ CK & I & 0 & \cdots & 0 \\ CAK & CB & I & \cdots & 0 \\ \cdots & \cdots & \ddots & \ddots & \vdots \\ CA^{n_{pred}-2}K & CA^{n_{pred}-3}K & \cdots & CK & I \end{bmatrix}$$

By replacing $e_t = y_t - Cx_t - Du_t$ in the state propagation in (10), a predictor-form state-space model can be formulated as: $x_{t+1} = \tilde{A}x_t + \tilde{B}u_t + Ky_t,\ y_t = Cx_t + Du_t + e_t$, where $\tilde{A} = A - KC$ and $\tilde{B} = B - KD$. From this model, we can find a relation between x_t and $x_{t-n_{init}}$ by:

$$x_t = \tilde{A}^P x_{t-n_{init}} + K_3 u_{init} + K_4 y_{init}$$

By replacing the above equation in (14), we can find:

$$\begin{aligned} y_{real} = &\begin{bmatrix} \Gamma K_4 & \Gamma K_3 & K_1 \end{bmatrix} \begin{bmatrix} y_{init} \\ u_{init} \\ u_{pred} \end{bmatrix} \\ &+ \tilde{A}^P x_{t-n_{init}} + K_2 e_{pred} \end{aligned} \tag{15}$$

The above linear relation can be extended to the Hankel matrices H_u, H_y by

$$\begin{aligned} H_{y,pred} = &\begin{bmatrix} \Gamma K_4 & \Gamma K_3 & K_1 \end{bmatrix} \begin{bmatrix} H_{y,init} \\ H_u \end{bmatrix} \\ &+ \tilde{A}^P X_{-n_{init}} + K_2 H_{e,pred} \end{aligned} \tag{16}$$

where $H_{e,pred}$ represents the prediction part in H_e, similar to the definition of $H_{y,pred}$. Besides, $X_{-n_{init}} := \begin{bmatrix} x_{1-n_{init}} & x_{2-n_{init}} & \cdots & x_{T-L+1-n_{init}} \end{bmatrix}$, where the time corresponds to that of $\{u_i\}_{i=1}^T$ and $\{y_i\}_{i=1}^T$ for constructing H_u and H_y.

Proof: [For Lemma 7] The data-driven prediction (7) can be written in the explicit form: $y_{pred} = H_{y,pred} Z^\dagger \begin{bmatrix} y_{init} \\ u_{init} \\ u_{pred} \end{bmatrix}$ by defining $Z := \begin{bmatrix} H_{y,init} \\ H_u \end{bmatrix}$ for sim-

plicity. It can be rewritten as

$$y_{pred} = \frac{1}{T} H_{y,pred} Z^\top (\frac{1}{T} Z Z^\top)^{-1} \begin{bmatrix} y_{init} \\ u_{init} \\ u_{pred} \end{bmatrix} \tag{17}$$

under Assumption 4, which almost surely ensures that the inverse exists for stochastic LTI systems. Besides, Assumption 3 ensures the existence of matrix correlation in (17).

By replacing $H_{y,pred}$ by (16) to (17), we can find the following result:

$$\begin{aligned} &\lim_{T\to\infty} \frac{1}{T} H_{y,pred} Z^\top (\frac{1}{T} Z Z^\top)^{-1} \\ &= \begin{bmatrix} \Gamma K_3 & K_1 & \Gamma K_4 \end{bmatrix} + \\ &\lim_{T\to\infty} \frac{1}{T} (\tilde{A}^P X_{-n_{init}} + K_2 H_{e,pred}) Z^\top (\frac{1}{T} Z Z^\top)^{-1} \\ &= \begin{bmatrix} \Gamma K_3 & K_1 & \Gamma K_4 \end{bmatrix} \end{aligned} \tag{18}$$

where the latter term in the second equation vanishes due to Assumption 2 and the lack of correlation between the u_t and e_t in open-loop data. Referring to (15), and (18) and Assumption 2, we can demonstrate consistency in the prediction made by (17) as $T \to \infty$ by:

$$\mathbb{E}_e(y_{pred} - y_{real}) = \mathbb{E}_e(\tilde{A}^P x_{t-n_{init}} + K_2 e_{pred}) = \mathbf{0}$$

■

Proof: [For Lemma 8] The proof is very similar to the one for For Lemma 7. The only difference is that $\lim_{T\to\infty} \frac{1}{T} K_2 H_{e,pred}) Z^\top \neq 0$ in general for closed-loop data. However, under the specific setup, i.e. $D = 0$ in the LTI system or the I/O data is collected by feedback control with at least one sample time delay, we again have $\lim_{T\to\infty} \frac{1}{T} K_2 H_{e,pred}) Z^\top = 0$. ■

REFERENCES

[1] 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.

[2] L. Schmitt, J. Beerwerth, M. Bahr, and D. Abel, "Data-driven predictive control with online adaption: Application to a fuel cell system," *IEEE Transactions on Control Systems Technology*, 2023.

[3] K. Chen, K. Zhang, X. Lin, Y. Zheng, X. Yin, X. Hu, Z. Song, and Z. Li, "Data-enabled predictive control for fast charging of lithium-ion batteries with constraint handling," *arXiv preprint arXiv:2209.12862*, 2022.

[4] J. Shi, Y. Lian, C. Salzmann, and C. N. Jones, "Adaptive data-driven predictive control as a module in building control hierarchy: A case study of demand response in switzerland," *arXiv preprint arXiv:2307.08866*, 2023.

[5] M. Yin, H. Cai, A. Gattiglio, F. Khayatian, R. S. Smith, and P. Heer, "Data-driven predictive control for demand side management: Theoretical and experimental results," *Applied Energy*, vol. 353, p. 122101, 2024.

[6] L. Huang, J. Coulson, J. Lygeros, and F. Dörfler, "Data-enabled predictive control for grid-connected power converters," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 8130–8135, IEEE, 2019.

[7] J. Wang, Y. Zheng, J. Dong, C. Chen, M. Cai, K. Li, and Q. Xu, "Implementation and experimental validation of data-driven predictive control for dissipating stop-and-go waves in mixed traffic," *IEEE Internet of Things Journal*, 2023.

[8] P. Van Overschee and B. De Moor, *Subspace identification for linear systems: Theory—Implementation—Applications*. Springer Science & Business Media, 2012.

[9] W. Favoreel, B. De Moor, and M. Gevers, "Spc: Subspace predictive control," *IFAC Proceedings Volumes*, vol. 32, no. 2, pp. 4004–4009, 1999.

[10] S. J. Qin, "An overview of subspace identification," *Computers & chemical engineering*, vol. 30, no. 10-12, pp. 1502–1513, 2006.

[11] G. Van der Veen, J.-W. van Wingerden, M. Bergamasco, M. Lovera, and M. Verhaegen, "Closed-loop subspace identification methods: an overview," *IET Control Theory & Applications*, vol. 7, no. 10, pp. 1339–1358, 2013.

[12] J. Dong, M. Verhaegen, and E. Holweg, "Closed-loop subspace predictive control for fault tolerant mpc design," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 3216–3221, 2008.

[13] J.-W. van Wingerden, S. P. Mulders, R. Dinkla, T. Oomen, and M. Verhaegen, "Data-enabled predictive control with instrumental variables: the direct equivalence with subspace predictive control," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, pp. 2111–2116, IEEE, 2022.

[14] R. Dinkla, S. P. Mulders, J. W. van Wingerden, and T. A. Oomen, "Closed-loop aspects of data-enabled predictive control," in *IFAC 22st Triennial World Congress*, 2023.

[15] Y. Wang, Y. Qiu, M. Sader, D. Huang, and C. Shang, "Data-driven predictive control using closed-loop data: An instrumental variable approach," *arXiv preprint arXiv:2309.05916*, 2023.

[16] K. Zhang, Y. Zheng, and Z. Li, "Dimension reduction for efficient data-enabled predictive control," *arXiv preprint arXiv:2211.03697*, 2022.

[17] S. Baros, C.-Y. Chang, G. E. Colon-Reyes, and A. Bernstein, "Online data-enabled predictive control," *Automatica*, vol. 138, p. 109926, 2022.

[18] M. Lovera, T. Gustafsson, and M. Verhaegen, "Recursive subspace identification of linear and non-linear wiener state-space models," *Automatica*, vol. 36, no. 11, pp. 1639–1650, 2000.

[19] P. Verheijen, G. R. G. da Silva, and M. Lazar, "Recursive data–driven predictive control with persistence of excitation conditions," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, pp. 467–473, IEEE, 2022.

[20] Y. Lian, J. Shi, M. Koch, and C. N. Jones, "Adaptive robust data-driven building control via bilevel reformulation: An experimental result," *IEEE Transactions on Control Systems Technology*, 2023.

[21] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "Linear tracking mpc for nonlinear systems—part ii: The data-driven case," *IEEE Transactions on Automatic Control*, vol. 67, no. 9, pp. 4406–4421, 2022.

[22] J. Coulson, J. Lygeros, and F. Dörfler, "Data-enabled predictive control: In the shallows of the deepc," in *2019 18th Eur. Control Conf. (ECC)*, pp. 307–312, IEEE, 2019.

[23] J. Coulson, J. Lygeros, and F. Dörfler, "Regularized and distributionally robust data-enabled predictive control," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 2696–2701, IEEE, 2019.

[24] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[25] M. Brand, "Fast low-rank modifications of the thin singular value decomposition," *Linear algebra and its applications*, vol. 415, no. 1, pp. 20–30, 2006.

[26] J. R. Bunch and C. P. Nielsen, "Updating the singular value decomposition," *Numerische Mathematik*, vol. 31, no. 2, pp. 111–129, 1978.

[27] M. Gu and S. C. Eisenstat, "Downdating the singular value decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 16, no. 3, pp. 793–810, 1995.

[28] I. Markovsky and P. Rapisarda, "Data-driven simulation and control," *International Journal of Control*, vol. 81, no. 12, pp. 1946–1959, 2008.

[29] Y. Lian, J. Shi, and C. N. Jones, "Physically consistent multiple-step data-driven predictions using physics-based filters," *IEEE Control Systems Letters*, 2023.

[30] M. S. Turan and G. Ferrari-Trecate, "Data-driven unknown-input observers and state estimation," *IEEE Control Systems Letters*, vol. 6, pp. 1424–1429, 2021.

[31] J. Shi, Y. Lian, and C. N. Jones, "Data-driven input reconstruction and experimental validation," *IEEE Control Systems Letters*, vol. 6, pp. 3259–3264, 2022.

[32] R. Penrose, "On best approximate solutions of linear matrix equations," in *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 52, pp. 17–19, Cambridge University Press, 1956.

[33] E. O'Dwyer, E. C. Kerrigan, P. Falugi, M. Zagorowska, and N. Shah, "Data-driven predictive control with improved performance using segmented trajectories," *IEEE Trans. Control Syst. Technol.*, 2022.