Closed-loop data-enabled predictive control and its equivalence with closed-loop subspace predictive control

Dinkla, Rogier; Oomen, Tom; Mulders, Sebastiaan Paul; van Wingerden, Jan Willem

DOI
10.1016/j.automatica.2025.112556

Publication date
2026

Document Version
Final published version

Published in
Automatica

Citation (APA)
Dinkla, R., Oomen, T., Mulders, S. P., & van Wingerden, J. W. (2026). Closed-loop data-enabled predictive control and its equivalence with closed-loop subspace predictive control. *Automatica*, *183*, Article 112556. https://doi.org/10.1016/j.automatica.2025.112556

Automatica 183 (2026) 112556

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Closed-loop data-enabled predictive control and its equivalence with closed-loop subspace predictive control☆

Rogier Dinkla [a,*], Tom Oomen [a,b], Sebastiaan Paul Mulders [a], Jan-Willem van Wingerden [a]

[a] Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2628CD Delft, The Netherlands
[b] Control Systems Technology Group, Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands

ARTICLE INFO

Article history:
Received 26 January 2024
Received in revised form 12 June 2025
Accepted 28 July 2025

Keywords:
Data-driven predictive control
Data-enabled predictive control
Closed-loop identification

ABSTRACT

Factors like growing data availability and increasing system complexity have sparked interest in data-driven predictive control (DDPC) methods like Data-enabled Predictive Control (DeePC). However, closed-loop identification bias arises in the presence of noise, which reduces the effectiveness of obtained control policies. In this paper we propose Closed-loop Data-enabled Predictive Control (CL-DeePC), a framework that unifies different approaches to address this challenge. To this end, CL-DeePC incorporates instrumental variables (IVs) to synthesize and sequentially apply consistent single or multi-step-ahead predictors. Furthermore, a computationally efficient CL-DeePC implementation is developed that reveals an equivalence with Closed-loop Subspace Predictive Control (CL-SPC). Time marching simulations of DeePC and CL-DeePC are conducted using Hankel matrices of past data that are updated at every time step to induce potentially troublesome closed-loop correlations between inputs and noise. Compared to DeePC, CL-DeePC simulations demonstrate superior reference tracking, with a sensitivity study finding a 48% lower susceptibility to noise-induced reference tracking performance degradation.

1. Introduction

Trends of increasing data availability and system complexity provide opportunities for data driven-driven control (Hou & Wang, 2013). In sharp contrast to the use of data in indirect data-driven approaches to synthesize a model by means of system identification, direct data-driven control approaches are promising because of their ability to derive a control law directly from data without having to realize an explicit system model as an often expensive intermediate step (Hjalmarsson, 2005).

A direct Data-driven Predictive Control (DDPC) method called Data-enabled Predictive Control (DeePC) is developed in Coulson et al. (2019a) and has recently seen considerable development and successful implementation (see, e.g., Markovsky et al. (2023), Huang et al. (2019), and Breschi et al. (2023)). DeePC relies on Willems' Fundamental Lemma in a receding horizon optimal control framework. This lemma reveals that for a deterministic, controllable linear time-invariant (LTI) system, any sufficiently persistently exciting input–output trajectory parameterizes all possible future input–output trajectories (Willems et al., 2005).

For non-deterministic systems, care has to be taken with such a parameterization in terms of only past input–output trajectories because this does not consider the effects of noise. If the block-Hankel data matrix in which the noisy input–output trajectories are stored is full rank then unattainable future trajectories may be predicted (Markovsky et al., 2023). On the other hand, if the data matrix is rank-deficient, then the DeePC problem may be infeasible. To deal with noise, slack variables and regularization initially served as heuristic measures to introduce robustness (Coulson et al., 2019a), and subsequently have been motivated formally to, e.g., account for insights from subspace identification (Dörfler et al., 2023), provide robust closed-loop stability guarantees (Berberich et al., 2021), distributional robustness (Coulson et al., 2019b), and robustness to structured uncertainty (Huang et al., 2023). Furthermore, Chiuso et al. (2025) recently demonstrates a separation principle for DDPC by which regularization accounts for uncertainty in output predictions. Other approaches to handle noise include averaging techniques (Sassella et al., 2022a), singular value based thresholding (Sassella et al., 2022b), and the use of maximum likelihood estimation (Yin et al., 2023). See also Sassella et al. (2023) for a discussion of such methods, or Verheijen et al. (2023) for a practical review of several DDPC methods.

To fundamentally address the consequences of noise in a data-driven setting, an instrumental variable (IV) approach is

☆ The material in this paper was not presented at any conference.This paper was recommended for publication in revised form by Associate Editor Simone Formentin under the direction of Editor Alessandro Chiuso.

* Corresponding author.
E-mail addresses: r.t.o.dinkla@tudelft.nl (R. Dinkla), t.a.e.oomen@tudelft.nl (T. Oomen), s.p.mulders@tudelft.nl (S. P. Mulders), j.w.vanwingerden@tudelft.nl (J.W. van Wingerden).

https://doi.org/10.1016/j.automatica.2025.112556

presented in van Wingerden et al. (2022). Using IVs to mitigate noise, the aforementioned approach of van Wingerden et al. demonstrates the equivalence between DeePC and a subspace identification-inspired direct DDPC method called Subspace Predictive Control (SPC) from Favoreel et al. (1999)[1]. The established equivalence provides opportunities for in-depth analysis and further development of the direct DDPC techniques by the strong fundamental basis of subspace identification methods.

The correlation between inputs and noise that arises from feedback results in closed-loop identification bias (Ljung & McKelvey, 1996). Such closed-loop correlations are practically unavoidable for systems that require feedback to be stabilized or for which open-loop experimentation is too costly. Moreover, these closed-loop correlations naturally arise with, e.g., adaptive DDPC applications like the one developed by Baros et al. (2022), whereby the data matrices that are used to characterize the system behavior are adapted over time to reflect time-varying or nonlinear dynamics. This paper employs fully adaptive DDPC implementations (for which the data matrices are updated at each time step) to study the effect of closed-loop correlation between inputs and noise. The resulting closed-loop identification bias has been shown to potentially degrade the performance of SPC and (given the aforementioned equivalence) DeePC in a batch-wise adaptive[2] setting (Dinkla et al., 2023).

To tackle the closed-loop identification issue, Dinkla et al. (2023) suggest using IVs and sequential step-ahead predictions, the use of which is further confirmed by work of Wang et al. (2023) and Shi et al. (2024). Drawing on subspace identification methods, the idea of sequential step-ahead predictions is also used by Closed-loop Subspace Predictive Control (CL-SPC) (Dong et al., 2008).

Although data-driven control algorithms have seen considerable development, to date, optimal noise mitigation under feedback is not completely addressed. The aim of this paper is to develop a direct DDPC technique named Closed-loop Data-enabled Predictive Control (CL-DeePC)[3] that addresses closed-loop identification bias, and provides a unifying framework of different solution strategies. Using particular design choices within the unified CL-DeePC framework uncovers a specific case in which there is an equivalence with CL-SPC. Although the relative merits of CL-SPC and (different design choices within) the unified CL-DeePC framework would make an interesting subject of study, this investigation focuses on demonstrating the mechanisms and use of CL-DeePC to address closed-loop identification bias. To this end, the resilience of CL-DeePC to this identification bias w.r.t. DeePC is illustrated in simulations with noisy closed-loop data.

The main contributions of this paper are:

1. to formally establish Closed-loop Data-enabled Predictive Control (CL-DeePC) as a new DDPC framework,
2. to incorporate IVs in CL-DeePC as a systematic noise-mitigation technique to provide consistent and causal single-step-ahead predictors or consistent multiple-step-ahead predictors,
3. to present a unified CL-DeePC framework that solves the closed-loop identification problem that arises in the presence of feedback and noise by sequential application of such consistent predictors,
4. to present a computationally efficient CL-DeePC technique that reveals an equivalence between CL-DeePC and CL-SPC,
5. to show the superior performance of CL-DeePC compared to DeePC in simulation.

This paper is structured as follows. The rest of this section introduces the used system model, notation, and fundamental relations called data equations. Section 2 subsequently clarifies the problem that is addressed in this paper and its specific formulation. In Section 3 CL-DeePC is developed using IVs to obtain a unified DDPC framework that encompasses both consistent sequential single and multi-step-ahead predictions. Section 4 introduces a computationally efficient CL-DeePC implementation for which Section 5 subsequently reveals an equivalence with CL-SPC. Section 6 presents simulation results that facilitate a comparison between the performance of DeePC and CL-DeePC. Finally, conclusions are presented in Section 7.

Preliminaries

This section presents the employed system model, notation, and fundamental relations called data equations.

1.1. System model

Consider a non-deterministic discrete LTI system $\mathcal{S}$ whose dynamics is described in the so-called innovation form by

$$\mathcal{S}_{\mathcal{I}} \begin{cases} x_{k+1} = Ax_k + Bu_k + Ke_k, & \text{(1a)} \\ y_k = Cx_k + Du_k + e_k, & \text{(1b)} \end{cases}$$

in which the subscript $k \in \mathbb{Z}$ denotes the discrete time index, $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^r$, $y_k \in \mathbb{R}^l$, $e_k \in \mathbb{R}^l$ respectively represent states, inputs, outputs, and zero-mean white innovation noise with variance $\Sigma(e_k) \succ 0$, and $\{A, B, C, D, K\}$ are system matrices of compatible dimensions. It is assumed that the input and output sequences of $\mathcal{S}$ are (jointly) quasi-stationary second-order ergodic stochastic processes. This ensures that limits of time averages involving these sequences exist and that sample correlations approach a finite true correlation with probability one as the number of samples goes to infinity (Ljung, 1999). Without loss of generality it is assumed that the data is generated by a minimal system realization. Moreover, K represents a Kalman filter gain matrix that renders $\tilde{A} = A - KC$ asymptotically stable such that $\rho = \max|\text{eig}(\tilde{A})| < 1$ (see, e.g., Verhaegen and Verdult (2007, Sec. 5.7)). Rearranging and substituting (1b) into (1a) obtains the equivalent predictor form

$$\mathcal{S}_{\mathcal{P}} \begin{cases} x_{k+1} = \tilde{A}x_k + \tilde{B}u_k + Ky_k, & \text{(2a)} \\ y_k = Cx_k + Du_k + e_k, & \text{(2b)} \end{cases}$$

in which $\tilde{B} = B - KD$.

1.2. Notation and definitions

This section introduces useful notation and definitions. Several block-Toeplitz matrices are defined by

$$\mathcal{T}_s(\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}) = \begin{bmatrix} \mathcal{D} & 0 & 0 & \cdots & 0 \\ \mathcal{C}\mathcal{B} & \mathcal{D} & 0 & \cdots & 0 \\ \mathcal{C}\mathcal{A}\mathcal{B} & \mathcal{C}\mathcal{B} & \mathcal{D} & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \mathcal{C}\mathcal{A}^{s-2}\mathcal{B} & \mathcal{C}\mathcal{A}^{s-3}\mathcal{B} & \cdots & \mathcal{C}\mathcal{B} & \mathcal{D} \end{bmatrix}, \quad (3)$$

in which the matrices $\{\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}\}$ are of compatible dimensions. Let $s \in \mathbb{Z}_{>0}$ denote a generic strictly-positive integer. As a subscript, s here indicates the number of block-rows. Let

[1] The equivalence between SPC and DeePC is also shown by Fiedler and Lucia (2021) and Breschi et al. (2023) in a noiseless setting and using regularizations respectively.

[2] With batch-wise adaptive operation subsequent controllers employ closed-loop data from only the preceding controller as a result of less frequent updating of the data matrices than is the case with fully adaptive operation.

[3] The notion of closed-loop DDPC employed in this paper refers to DDPC that uses closed-loop data. This is different from the notion of closed-loop in model predictive control literature, where it refers to the use of feedback to handle disturbances.

$I_s \in \mathbb{R}^{s\times s}$ represent an identity matrix. Equation (3) then defines the block-Toeplitz matrices

$$\mathcal{T}_s^{\mathrm{u}} = \mathcal{T}_s(A, B, C, D), \quad \mathcal{H}_s = \mathcal{T}_s(A, K, C, I_l),$$
$$\widetilde{\mathcal{T}}_s^{\mathrm{u}} = \mathcal{T}_s(\tilde{A}, \tilde{B}, C, D), \quad \widetilde{\mathcal{H}}_s = \mathcal{T}_s(\tilde{A}, K, -C, I_l).$$

In addition, for a generic s and specific past data window length $p \in \mathbb{Z}_{>0}$, the extended observability matrices Γ_s and $\widetilde{\Gamma}_s$ as well as extended reversed controllability matrices $\widetilde{\mathcal{K}}^{\mathrm{u}}$, $\widetilde{\mathcal{K}}^{\mathrm{y}}$ are defined by

$$\begin{aligned}
\Gamma_s &= \begin{bmatrix} C^\top & (CA)^\top & \cdots & (CA^{s-1})^\top \end{bmatrix}^\top && \in \mathbb{R}^{sl\times n},\\
\widetilde{\Gamma}_s &= \begin{bmatrix} C^\top & (C\tilde{A})^\top & \cdots & (C\tilde{A}^{s-1})^\top \end{bmatrix}^\top && \in \mathbb{R}^{sl\times n},\\
\widetilde{\mathcal{K}}^{\mathrm{u}} &= \begin{bmatrix} \tilde{A}^{p-1}\tilde{B} & \tilde{A}^{p-2}\tilde{B} & \cdots & \tilde{A}\tilde{B} & \tilde{B} \end{bmatrix} && \in \mathbb{R}^{n\times pr},\\
\widetilde{\mathcal{K}}^{\mathrm{y}} &= \begin{bmatrix} \tilde{A}^{p-1}K & \tilde{A}^{p-2}K & \cdots & \tilde{A}K & K \end{bmatrix} && \in \mathbb{R}^{n\times pl}.
\end{aligned}$$

Data vectors are denoted as

$$\underline{u}_{k,s} = \begin{bmatrix} u_k^\top & u_{k+1}^\top & \cdots & u_{k+s-1}^\top \end{bmatrix}^\top \in \mathbb{R}^{sr},$$

which represents a vector of ordered input data starting at time index k, and containing a number of samples s.

A block-Hankel data matrix is defined by[4]

$$U_{k,s,N} = \frac{1}{\sqrt{N}} \begin{bmatrix} u_k & u_{k+1} & \cdots & u_{k+N-1} \\ u_{k+1} & u_{k+2} & \cdots & u_{k+N} \\ \vdots & \vdots & \ddots & \vdots \\ u_{k+s-1} & u_{k+s} & \cdots & u_{k+N+s-2} \end{bmatrix} \in \mathbb{R}^{sr\times N},$$

in which $N \in \mathbb{Z}_{>0}$ represents the number of columns with s data samples each, starting from time index k. A particular type of data matrix that concatenates input and output block-Hankel matrices is defined by

$$\Psi_{k,s,N} = \begin{bmatrix} U_{k,p+s,N}^\top & Y_{k,p,N}^\top \end{bmatrix}^\top \in \mathbb{R}^{((p+s)r+pl)\times N}, \tag{4}$$

in which k indicates the starting index, and s and N parameterize the dimensions of the concatenated matrix together with p. To improve readability the second subscript of matrices like $U_{k,s,N}$ and $\Psi_{k,s,N}$ is omitted if $s = 1$. We indicate vectors and matrices that are composed entirely or partly of predictions by respectively $\hat{(\cdot)}$ or $\overline{(\cdot)}$.

Block-Hankel data matrices are employed to define the notion of persistency of excitation below.

Definition 1 (*Verhaegen & Verdult, 2007, Def. 10.1*)**.** A signal consisting of samples $w_j \in \mathbb{R}^v$, $j \in [k, k+s+N-2]$ is persistently exciting of order s if the associated block-Hankel matrix $W_{k,s,N} \in \mathbb{R}^{sv\times N}$ is full row rank.

Furthermore, the expectation of a variable with a stochastic component is denoted by $\mathbb{E}[\cdot]$. For sequences of scalar random variables a_N and b_N, $a_N = o(b_N)$ indicates that $\lim_{N\to\infty} a_N/b_N = 0$. Likewise $a_N = o_P(b_N)$ stipulates that $\forall\delta > 0$, $\lim_{N\to\infty} P[|a_N/b_N| > \delta] = 0$.

In addition, the well-posedness of a closed-loop system is defined as follows.

Definition 2 (*Van Overschee & De Moor, 1997*)**.** A closed-loop system composed of an LTI plant and controller with respectively direct feedthrough matrices D and D_c is well-posed if $I_l + DD_c$ is invertible.

The outputs of a well-posed closed-loop system are uniquely defined by the reference and the states of the plant and controller. Note that a practical sufficient condition for well-posedness is that either $D = 0$ or $D_c = 0$.

[4] This relates to the Hankel matrix operator $\mathscr{H}_s()$ from, e.g., Coulson et al. (2019a) as $U_{k,s,N} = \frac{1}{\sqrt{N}}\mathscr{H}_s(\underline{u}_{k,N+s-1})$.

1.3. The data equations

This section applies the introduced notation to the system models (1) and (2), thereby deriving several fundamental relations called data equations. These relations are essential for later consistency proofs and derivations.

To this end, iterative application of respectively (1) and (2) leads to

$$Y_{k_p,s,N} = \Gamma_s X_{k_p,N} + \mathcal{T}_s^{\mathrm{u}} U_{k_p,s,N} + \mathcal{H}_s E_{k_p,s,N}, \tag{5}$$
$$Y_{k_p,s,N} = \widetilde{\Gamma}_s X_{k_p,N} + \widetilde{\mathcal{T}}_s^{\mathrm{u}} U_{k_p,s,N} + E_{k_p,s,N} + (I_{sl} - \widetilde{\mathcal{H}}_s) Y_{k_p,s,N}, \tag{6}$$

in which we employ a recurring shorthand for time indices exemplified by $k_p = k + p$. Furthermore, the initial states can be rewritten in terms of preceding states and input–output data using (2) as

$$X_{k_p,N} = \tilde{A}^p X_{k,N} + \widetilde{\mathcal{K}}^{\mathrm{u}} U_{k,p,N} + \widetilde{\mathcal{K}}^{\mathrm{y}} Y_{k,p,N}. \tag{7}$$

Substitute (7) into (5) and (6) to find the *data equations*

$$Y_{k_p,s,N} = L_s \Psi_{k,s,N} + \mathcal{H}_s E_{k_p,s,N} + \Gamma_s \tilde{A}^p X_{k,N}, \tag{8}$$
$$Y_{k_p,s,N} = \widetilde{L}_s \Psi_{k,s,N} + E_{k_p,s,N} + (I_{sl} - \widetilde{\mathcal{H}}_s) Y_{k_p,s,N} + \widetilde{\Gamma}_s \tilde{A}^p X_{k,N}, \tag{9}$$

in which $L_s, \widetilde{L}_s \in \mathbb{R}^{sl\times((p+s)r+pl)}$ are defined as

$$L_s = \begin{bmatrix} \Gamma_s\widetilde{\mathcal{K}}^{\mathrm{u}} & \mathcal{T}_s^{\mathrm{u}} & \Gamma_s\widetilde{\mathcal{K}}^{\mathrm{y}} \end{bmatrix}, \quad \widetilde{L}_s = \begin{bmatrix} \widetilde{\Gamma}_s\widetilde{\mathcal{K}}^{\mathrm{u}} & \widetilde{\mathcal{T}}_s^{\mathrm{u}} & \widetilde{\Gamma}_s\widetilde{\mathcal{K}}^{\mathrm{y}} \end{bmatrix}.$$

2. Problem formulation

This section clarifies the problem addressed in this paper.

For a future horizon length f over $k \in [\hat{i}_p, \hat{i}_p + f)$, consider a user-defined cost function $\mathcal{J}(\underline{u}_{\hat{i}_p,f}, \underline{\hat{y}}_{\hat{i}_p,f})$. Approaches typically seek to minimize such a cost function , potentially subject to input and output constraints, for which purpose an accurate predictor is desirable. One may attempt to obtain a predictor from a noise-affected dataset $\mathcal{D} = \{(u_k, y_k), k = i, \ldots, i + \bar{N} - 1\}$ of $\bar{N}$ past input and output samples that were collected in closed-loop. However, the synthesis of an accurate data-driven predictor from $\mathcal{D}$ is complicated by the fact that its data was both obtained in closed-loop and was affected by noise, as is explained next.

2.1. Closed-loop identification bias

This section illustrates the issue of closed-loop identification bias for DDPC along the lines of SPC as treated in Dinkla et al. (2023).

SPC employs an output predictor of the form

$$\underline{\hat{y}}_{\hat{i}_p,f} = \widehat{L}_f \Psi_{i,f,1}. \tag{10}$$

The least-squares estimate $\widehat{L}_f$ is obtained from N sufficiently persistently exciting such past input–output trajectories from $k \in [i, \hat{i}_p)$ as

$$\widehat{L}_f = \arg\min_L \|Y_{i_p,f,N} - L\Psi_{i,f,N}\|_{\mathrm{F}}^2 = Y_{i_p,f,N}\Psi_{i,f,N}^\dagger,$$

where F denotes the Frobenius norm and † indicates the Moore–Penrose pseudoinverse. Applying (8) assuming a negligible initial state contribution,[5] this becomes

$$\widehat{L}_f = L_f + \mathcal{H}_f \underbrace{E_{i_p,f,N}\Psi_{i,f,N}^\top} \left(\Psi_{i,f,N}\Psi_{i,f,N}^\top\right)^{-1}. \tag{11}$$

The underbraced term contains the sample correlation matrix formed by $E_{i_p,f,N}U_{i_p,f,N}^\top$. In closed-loop, inputs become correlated with noise such that this correlation matrix does not go to zero

[5] Formally motivated later by Assumption 1

asymptotically as the total number of samples $\bar{N}$ increases with $N \to \infty$. Hence, (11) illustrates that the estimate $\widehat{L}_f$ is inconsistent, leading to an inconsistent predictor[6] (10) with potentially deteriorated performance.

2.2. Closed-loop direct DDPC problem formulation

Given the issue posed by closed-loop identification bias this paper considers the following kind of optimal control problem that is solved at each time step:

$$\min_{\underline{u}_{\hat{i}_p,f}} \mathcal{J}(\underline{u}_{\hat{i}_p,f}, \underline{\hat{y}}_{\hat{i}_p,f}) \tag{12a}$$

$$\text{s.t.} \quad \underline{u}_{\hat{i}_p,f} \in \mathcal{U}_f, \underline{\hat{y}}_{\hat{i}_p,f} \in \mathcal{Y}_f, \tag{12b}$$

$$\underline{\hat{y}}_{\hat{i}_p,f} = y_f\big(\underline{u}_{\hat{i}_p,f}, \mathcal{D}\big), \tag{12c}$$

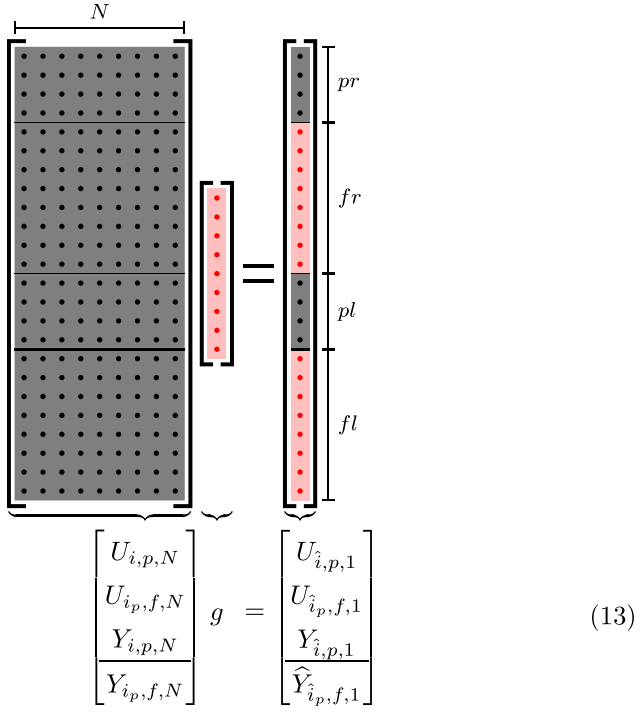
$$\lim_{\bar{N}\to\infty} \mathbb{E}\left[\underline{\hat{y}}_{\hat{i}_p,f} - \underline{y}_{\hat{i}_p,f}\right] = 0, \tag{12d}$$

where $y_f(\cdot)$ represents the form of the output predictor, and $\mathcal{U}_f$ and $\mathcal{Y}_f$ respectively represent sets of allowable future input and output trajectories.

The principal focus of this paper is on how a *consistent* data-driven predictor can be obtained for *direct* DDPC in the presence of noise using past *closed-loop data*. With respect to (12) this paper thus specifically seeks to develop a *non-parametric predictor in (12c) that inherently satisfies (12d) with noise-affected closed-loop data.*

The focus on the development of a DeePC-inspired direct, non-parametric (as opposed to an indirect, parametric) DDPC method is motivated by several benefits. As a non-parametric method, DeePC offers a more versatile formulation than its parametric equivalent SPC. DeePC's more versatile formulation is thereby more conducive to various extensions like γ-DDPC (Breschi et al., 2023) and regularization (Coulson et al., 2019a; Dörfler et al., 2023; Huang et al., 2019; Markovsky et al., 2023), each of which that has shown promising results w.r.t. indirect or parametric DDPC methods. Furthermore, non-parametric methods avoid an intermediate – often costly – step to obtain a particular parametric system realization (Hjalmarsson, 2005). Moreover, the combination of system identification and model-based control in indirect DDPC is often complicated by the use of different parametric or uncertainty descriptions and the need to choose a particular model structure (Markovsky et al., 2023).

[6] We shall refer to predictors as 'consistent' if their bias (i.e. expected error) asymptotically goes to zero as $N \to \infty$, and 'inconsistent' otherwise.



(13)

Fig. 1. Visualization of known (black) and unknown (red) variables in DeePC without IVs. Dots represent an input $u_k \in \mathbb{R}^r$, output $y_k \in \mathbb{R}^l$, or element of the vector g. An f-step-ahead predictor is formed directly by taking a linear combination of past input and output data.

3. Closed-loop Data-enabled Predictive Control

In this section, CL-DeePC is established, thereby providing Contribution 1. The concept of single-step-ahead predictors as a means to prevent closed-loop identification bias is explained first. Thereafter, sequential application of such predictors is explained before a rigorous exposition introduces IVs to assure their consistency. Results are subsequently generalized to the sequential application of multi-step-ahead predictors, thereby providing Contributions 2 and 3.

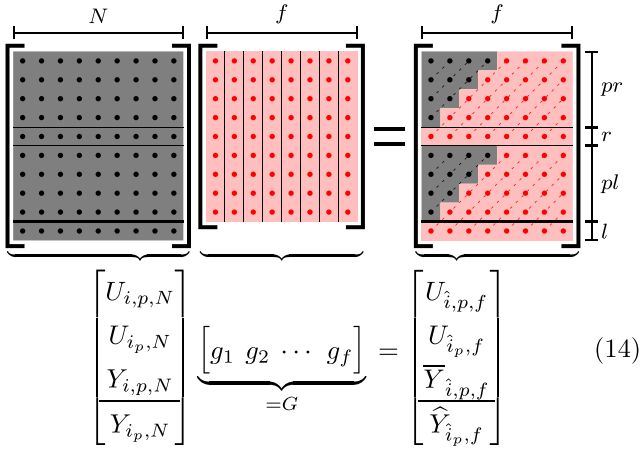
3.1. Preventing closed-loop identification bias: $f = 1$

As explained in Section 2.1 the closed-loop identification bias that we seek to address arises from correlation between inputs and noise contained in the underbraced term of (11). For controllers without direct feedthrough, inputs are correlated only with preceding noise (i.e. $\mathbb{E}\left[e_k u_j^\top\right] = 0\ \forall k \geq j$ and $\mathbb{E}\left[e_k u_j^\top\right] \neq 0\ \forall k < j$). Studying (11), note that such nonzero correlations between inputs and preceding noise only appear if $f > 1$. If instead $f = 1$, only correlations between inputs and concurrent noise feature, which are zero. This motivates the use of single-step-ahead predictors for which $f = 1$.

3.2. The DeePC-based predictor

The predictor used in DeePC is briefly introduced here.

An example of such a predictor is given by Fig. 1, where (13) describes an input–output trajectory (on the right-hand side) as a linear combination g of sufficiently persistently exciting past input–output data (black dots on the left-hand side). The



(14)

Fig. 2. Visualization of known (black) and unknown (red) variables in CL-DeePC without IVs. The composition of block-Hankel matrices on the right hand side (note the dashed anti-diagonals) results from sequential application of DeePC with $f = 1$.

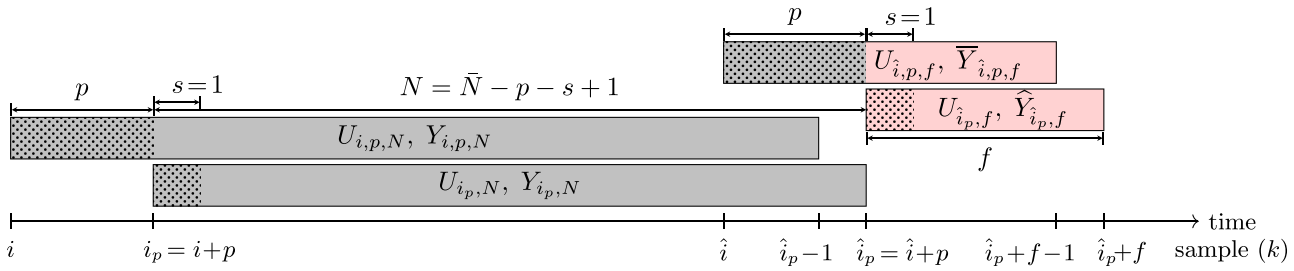


Fig. 3. Visualization of data use by CL-DeePC as in (14) with past data $[i, \hat{i}_p)$ in grey and future data $[\hat{i}_p, \hat{i}_p+f)$ in red. Block-Hankel data matrices occupy black-outlined rectangles that indicate the range of the contained data. The basic building block of CL-DeePC is a single-step-ahead ($s = 1$) output predictor (see the dotted patterns left and right). Based on N past corresponding input–output trajectories (bottom left two grey rectangles) the single single-step-ahead predictor is sequentially applied f times for a total prediction horizon of f samples (see top right two rectangles). For the fully adaptive implementation used here, $\hat{i}_p = i + \bar{N}$, where $\bar{N}$ is the number of past input or output samples that is contained by the lower left two rectangles. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

input–output trajectory on the right-hand side of (13) contains past input–output data (black dots) that encodes information on an initial state, future inputs, and output predictions. For more information on DeePC see Coulson et al. (2019a).

3.3. CL-DeePC as sequential DeePC with $f = 1$

This section shows how sequential single-step-ahead predictions with DeePC lead to a novel CL-DeePC formulation that addresses closed-loop identification bias.

CL-DeePC, as visualized by Fig. 2, sequentially applies DeePC with $f = 1$. Successive vectors g_k in G of (14) take linear combinations of the same past data matrix on the left hand side to obtain input–output trajectories on the right hand side that are shifted by a single time step further into the future for successive columns. Fig. 3 clarifies the resulting data use. The sequential application of a single-step-ahead predictor is apparent in Fig. 2 from the introduction of a single new unknown (red) input and output prediction by subsequent columns on the right hand side. This results in the composition of block-Hankel matrices on the right hand side of (14), as visualized by the dashed anti-diagonals. For a derivation of Eqs. (13) and (14) based on the data equation (8) see Section 3.5.1.

Although a single-step-ahead predictor (i.e. $f = 1$ in (13)) prevents closed-loop identification bias as described in Section 3.1, for receding horizon optimal control settings a longer prediction horizon ($f > 1$) is often desirable to, e.g., achieve closed-loop stability. To obtain such a longer prediction horizon a one-step-ahead DeePC-predictor can be applied sequentially.

Although (13) and (14) are equivalent if $f = 1$, for $f > 1$ there are important differences. For example, consider that the dimension of G in (14) is then notably larger than that of g in (13). Since this entails a considerable increase in the number of unknown optimization variables, this motivates the development of an efficient CL-DeePC in Section 4. Furthermore, important advantages of CL-DeePC over DeePC are the former's suitability for closed-loop noise-affected data (see also the next section), and its ability to, by the sequential use of single-step-ahead predictors, enforce causality (see also Remark 6 and reference therein) and use data efficiently. With regards to the efficient use of data note the size of the data matrix on the left hand sides of Fig. 1 and Fig. 2, which is considerably smaller for the latter.

3.4. Noise mitigation using instrumental variables

The CL-DeePC formulation that was introduced in the previous subsection mitigates the closed-loop identification issue that arises from correlation between inputs and noise. However, sampling of noise in the past data matrix by the columns of G in (14) may, similarly to DeePC, still lead to predictions of unattainable of input–output trajectories (Markovsky et al., 2023). This necessitates further noise mitigation strategies. The strategy that is presented here incorporates IVs in a similar fashion as in van Wingerden et al. (2022).

3.4.1. Desirable properties of IVs

As typically employed in estimation problems, IVs are typically chosen to have two desirable properties that guarantee the consistency of the estimated quantity. Assuming sufficiently persistently exciting data such that $\Psi_{i,N}$ is full row rank, these conditions on the IV matrix $\mathcal{Z} \in \mathbb{R}^{n_z \times N}$ are (Verhaegen & Verdult, 2007)

$$\operatorname{rank}\Big(\lim_{N\to\infty} \hat{\Sigma}_{\psi z}\Big) = (p+1)r + pl, \tag{15}$$

$$\lim_{N\to\infty} \hat{\Sigma}_{ez} := 0, \tag{16}$$

in which $\hat{\Sigma}_{\psi z} = \Psi_{i,N}\mathcal{Z}^\top$ and $\hat{\Sigma}_{ez} = E_{i_p,N}\mathcal{Z}^\top$ represent sample correlations that approach their respective true counterparts $\Sigma_{\psi z}$ and Σ_{ez} in the limit $N \to \infty$. The former condition preserves the informativity of the employed data and the latter equation stipulates that the IV is uncorrelated with noise. This work considers a generic $\mathcal{Z}$ for the sake of generality. For examples of suitable choices for $\mathcal{Z}$ see Wang et al. (2023) and Lemma 2.

3.4.2. Assumptions

Before moving on to the main results of this work, three assumptions are presented.

Assumption 1. The past window length p and the number of columns used to construct the block-Hankel data matrices N go to infinity such that, with scalars $d, \alpha \in \mathbb{R}$, and $\rho = \max|\operatorname{eig}(A)|$,

$$p \geq -\frac{d \log N}{2 \log|\rho|} \quad 1 < d < \infty,$$
$$p = o((\log N)^\alpha) \quad \alpha < \infty.$$

For the system $\mathcal{S}$ from Section 1.1 with $\rho < 1$ this assumption is required for consistency of estimators that are based on closed-loop data because it ensures that errors due to mis-specification of the initial condition are $o_P(1/\sqrt{N})$ (and may therefore be neglected), and that sample correlations approach true correlations that are well-defined (Bauer & Ljung, 2002; Chiuso, 2006). This assumption will prove to be instrumental to demonstrate the closed-loop consistency of predictors.

Remark 1. The number of past samples in a matrix like $\Psi_{i,s,N}$ from (4) is $\bar{N} = N + p + s - 1$. By Assumption 1, the limit $\bar{N} \to \infty$ in (12d) shall require both $p, N \to \infty$.

Assumption 2. The known past data matrix of the form $\Psi_{i,s,N}$, as defined by (4), is full row rank.

This assumption entails that used past input–output data is sufficiently persistently exciting, which is an important requirement for many data-driven applications (see for example van Wingerden et al. (2022)).

Remark 2. While the rank condition in Assumption 2 is easy to verify, its satisfaction cannot be guaranteed for unknown noise (consider degenerate noise cases like $e_k = 0\,\forall k$). Potentially problematic cases for Assumption 2 are when closed-loop data derives from low-order (e.g., proportional) controllers (Söderström & Stoica, 1989, Ch. 10), and when inputs are (nearly) constant (e.g., due to input constraints or tracking of a constant reference). An additional random input disturbance can enhance persistency of excitation, but may compromise control performance. This motivates research on optimal closed-loop experiment design (Gevers, 2005; Hallouzi & Verhaegen, 2008; Jansson & Hjalmarsson, 2005). This work does not consider such extensions because such approaches obscure the effects of closed-loop correlations that are studied here.

Assumption 3. The IV matrix $\mathcal{Z} \in \mathbb{R}^{n_z \times N}$ is chosen such that (15) and (16) hold.

Under this assumption the chosen IV matrix exhibits the desirable properties described in Section 3.4.1.

3.4.3. Using IVs and sequential step-ahead predictions

This section presents results that pertain to the use of IVs by CL-DeePC to obtain a consistent predictor.

Lemma 1. *Consider input–output data generated (either in open or a well-posed closed-loop) by the minimal discrete LTI system $\mathcal{S}$ from Section 1.1. By Assumption 2 the known data matrix $\Psi_{i,N}$ has full row rank. Furthermore, consider a partially known data matrix $\overline{\Psi}_{\hat{i},f}$ and output predictor $\widehat{Y}^{\mathrm{IV}}_{\hat{i}_p,f}$ as on the right hand side of Fig. 2, and an IV matrix $\mathcal{Z} \in \mathbb{R}^{n_z \times N}$ such that $\hat{\Sigma}_{\psi z} = \Psi_{i,N}\mathcal{Z}^\top$ is full row rank, then $\exists G^{\mathrm{IV}} \in \mathbb{R}^{n_z \times f}$ such that*

$$\begin{bmatrix} \Psi_{i,N} \\ Y_{i_p,N} \end{bmatrix} \mathcal{Z}^\top G^{\mathrm{IV}} = \begin{bmatrix} \overline{\Psi}_{\hat{i},f} \\ \widehat{Y}^{\mathrm{IV}}_{\hat{i}_p,f} \end{bmatrix}. \tag{17}$$

Proof. To start, note that any solution to the top row of (17) is consistent with the bottom row since $\widehat{Y}^{\mathrm{IV}}_{\hat{i}_p,f}$ is unknown a priori. In addition, since by assumption $\hat{\Sigma}_{\psi z} = \Psi_{i,N}\mathcal{Z}^\top$ is full row rank, based on the top row of (17), solutions for G^{IV} are feasible and given by

$$G^{\mathrm{IV}} = \hat{\Sigma}^{\dagger}_{\psi z}\overline{\Psi}_{\hat{i},f} + \hat{\Pi}^{\perp}_{\psi z} W, \tag{18}$$

where $\hat{\Pi}^{\perp}_{\psi z} = I_{n_z} - \hat{\Sigma}^{\dagger}_{\psi z}\hat{\Sigma}_{\psi z}$ is a matrix that forms an orthogonal projection onto the column space of $\hat{\Sigma}_{\psi z}$, and $W \in \mathbb{R}^{n_z \times f}$ is a matrix of optimization variables. □

Remark 3. Choosing $\mathcal{Z}$ such that $\hat{\Sigma}_{\psi z} = \Psi_{i,N}\mathcal{Z}^\top$ is square and invertible ensures that, with reference to (18), G^{IV} is uniquely determined by $\overline{\Psi}_{\hat{i},f}$. Moreover, by (17), $\widehat{Y}^{\mathrm{IV}}_{\hat{i}_p,f}$ is then also uniquely determined by $\overline{\Psi}_{\hat{i},f}$.

Lemma 1 effectively indicates that the system of equations provided by (17) is consistent provided that $\hat{\Sigma}_{\psi z}$ is full row rank. The following theorem concerns the consistency of the resulting output predictor when using data that has been collected in closed-loop.

Theorem 1. *Consider the minimal discrete LTI system $\mathcal{S}$ from Section 1.1 with $\rho < 1$ to generate input–output data in a well-posed closed-loop configuration by means of a causal controller, known past data matrix $\Psi_{i,N}$ of full row rank by Assumption 2, and partially-known data matrix $\overline{\Psi}_{\hat{i},f}$. Furthermore, let $p, N \to \infty$ as specified by Assumption 1 and let $\mathcal{Z} \in \mathbb{R}^{n_z \times N}$ satisfy Assumption 3. Then the predictor $\widehat{Y}^{\mathrm{IV}}_{\hat{i}_p,f}$ specified by (17) is consistent, i.e.:*

$$\lim_{p,N\to\infty} \mathbb{E}\left[\widehat{Y}^{\mathrm{IV}}_{\hat{i}_p,f} - Y_{\hat{i}_p,f}\right] = 0. \tag{19}$$

Proof. Starting with finite p and N, by application of Lemma 1, (17) forms a consistent system of equations that specifies an output predictor $\widehat{Y}^{\mathrm{IV}}_{\hat{i}_p,f}$. Expanding the bottom row of (17) by substitution of $Y_{i_p,N}$ as obtained from (9) results in

$$\widehat{Y}^{\mathrm{IV}}_{\hat{i}_p,f} = \widetilde{L}_1 \underbrace{\Psi_{i,N}\mathcal{Z}^\top G^{\mathrm{IV}}}_{=\overline{\Psi}_{\hat{i},f}} + \underbrace{E_{i_p,N}\mathcal{Z}^\top}_{=\hat{\Sigma}_{ez}} G^{\mathrm{IV}} + \widetilde{\Gamma}_1 \tilde{A}^p X_{i,N}\mathcal{Z}^\top G^{\mathrm{IV}}, \tag{20}$$

wherein $\hat{\Sigma}_{ez}$ is recognized from its definition and $\overline{\Psi}_{\hat{i},f}$ is recognized from the top part of (17). By comparison, from (9), the true output is

$$Y_{\hat{i}_p,f} = \widetilde{L}_1 \Psi_{\hat{i},f} + E_{\hat{i}_p,f} + \widetilde{\Gamma}_1 \tilde{A}^p X_{\hat{i},f}. \tag{21}$$

Under Assumption 1, as $p, N \to \infty$ the sample correlations go to well-defined true correlations and the contribution of the initial condition may be neglected (Bauer & Ljung, 2002; Chiuso, 2006). In addition, by Assumption 3, $\mathcal{Z}$ satisfies (16) such that it is uncorrelated with noise. Hence, computing the bias of the output predictor from (20) and (21) in the aforementioned limits yields

$$\lim_{p,N\to\infty} \mathbb{E}\left[\widehat{Y}^{\mathrm{IV}}_{\hat{i}_p,f} - Y_{\hat{i}_p,f}\right] = \lim_{p,N\to\infty} \widetilde{L}_1 \mathbb{E}\left[\overline{\Psi}_{\hat{i},f} - \Psi_{\hat{i},f}\right]. \tag{22}$$

The remaining consistency proof for (19) is sequential. First, note the structure of $\overline{\Psi}_{\hat{i},f}$ and $\Psi_{\hat{i},f}$ as visualized by Fig. 2. The leftmost columns of $\overline{\Psi}_{\hat{i},f}$ and $\Psi_{\hat{i},f}$ are equal because they contain no estimates. Therefore, the leftmost column of the predictor $\widehat{Y}^{\mathrm{IV}}_{\hat{i}_p,f}$ is consistent. Moreover, (22) shows that for subsequent columns in the output predictor the asymptotic bias is a linear combination of the asymptotic bias of preceding columns. Hence, since the leftmost column of the output predictor is consistent, so are the other columns of the output predictor, resulting in (19). □

This section has presented (17): a novel, non-parametric formulation of a predictor that Theorem 1 demonstrates provides consistent predictions with noisy closed-loop data. To this end, Theorem 1 is inspired by the use of sequential single step-ahead predictors (Ljung & McKelvey, 1996) and IVs (see, e.g., Gilson and van den Hof (2005)) to address closed-loop identification bias. Moreover, the employed Assumption 1 presents insights from closed-loop subspace identification (Chiuso, 2006) that are used here to assure consistency of predictors with closed-loop data. The subsequent section presents a specific suitable choice for the IV matrix $\mathcal{Z}$ in Theorem 1.

3.4.4. Exemplary IV that leads to a consistent predictor

The following lemma suggests a specific suitable IV matrix for Theorem 1 that satisfies Assumption 3, thus enabling consistent output predictions with (17).

Lemma 2. *For Theorem 1, if the data-generating controller employs no direct feedthrough, a suitable choice of IV matrix that satisfies Assumption 3 is $\mathcal{Z} = \Psi_{i,N}$.*

Proof. This proof requires it to be shown that the choice $\mathcal{Z} = \Psi_{i,N}$ satisfies conditions (15) and (16) from Assumption 3 when the controller lacks direct feedthrough. Upon inspection, the rank condition specified by (15) is indeed satisfied by this choice because $\Psi_{i,N}$ is (by Assumption 2 in Theorem 1) full row rank. With regards to (16), substituting the choice of IV and applying Assumption 1 from Theorem 1 obtains

$$\begin{aligned}\lim_{p,N\to\infty} \hat{\Sigma}_{ez} &= \lim_{p,N\to\infty} \frac{1}{N} \sum_{k=i_p}^{i_p+N-1} e_k \begin{bmatrix} \underline{u}_{k-p,p+1}^\top & \underline{y}_{k-p,p}^\top \end{bmatrix} \\ &= \Sigma_{ez} = \mathbb{E}\left[e_k \begin{bmatrix} \underline{u}_{k-p,p+1}^\top & \underline{y}_{k-p,p}^\top \end{bmatrix} \right] = 0.\end{aligned} \tag{23}$$

The last equality in (23) holds due to the (assumed) lack of direct feedthrough of the used causal controller, due to which inputs are correlated with preceding noise ($\mathbb{E}[e_k u_j^\top] \neq 0,\ \forall j > k$), but inputs are uncorrelated with concurrent and subsequent noise ($\mathbb{E}[e_k u_j^\top] = 0,\ \forall j \leq k$). Moreover, the innovation noise is also uncorrelated with preceding outputs ($\mathbb{E}[e_k y_j^\top] = 0,\ \forall j < k$). □

Remark 4. For G in (14) consider the use of IVs such that $G = \mathcal{Z}^\top G^{\mathrm{IV}}$ as in (17). Based on (18), with full row rank $\Psi_{i,N}$, the choice $\mathcal{Z} = \Psi_{i,N}$ induces a minimum norm least squares solution for G.

Lemma 2 suggests a choice of IV matrix $\mathcal{Z}$ that ensures a consistent output predictor by Theorem 1. With the sequential application of one-step-ahead predictors as per (17), the choice $\mathcal{Z} = \Psi_{i,N}$ is useful because it relies only on past input–output data, keeps the number of optimization variables in (17) to a minimum (see Remark 3), and enforces causality of the predictor at finite N. This last point will become clear in Section 4.

3.5. Unified CL-DeePC framework: incorporating IVs and sequential multi-step-ahead predictors

This section generalizes Theorem 1 to incorporate the sequential use of multi-step-ahead predictors. To this end, a non-parametric predictor that generalizes different design choices of s and $\mathcal{Z}$ is first derived. Thereafter, the derived formulation is shown to obtain a consistent predictor with closed-loop data.

3.5.1. Deriving the unified CL-DeePC predictor

Based on (8), data equations for past outputs $Y_{i_p,s,N}$, and ideal predictions $\widehat{Y}^*_{\hat{i}_p,s,b}$ based on past data and idyllic system knowledge are

$$Y_{i_p,s,N} = L_s \Psi_{i,s,N} + \Gamma_s \tilde{A}^p X_{i,N} \ + \mathcal{H}_s E_{i_p,s,N} \tag{24}$$

$$\widehat{Y}^*_{\hat{i}_p,s,b} = L_s \overline{\Psi}^*_{\hat{i},s,b} + \Gamma_s \tilde{A}^p X_{\hat{i},b}, \tag{25}$$

where $b \in \mathbb{Z}_{>0}$, and the bar and superscript $*$ in $\overline{\Psi}^*_{\hat{i},s,b}$ reflects the fact that it is itself comprised in part of unknown, ideal output predictions.

Defining $q \in \mathbb{Z}_{>0}$, we let $b = (q-1)s + 1$ and select every s^{th} column of (25) starting from the first one (i.e. columns $1, s+1, 2s+1$, etc.). We denote this column selection by the superscript m to write

$$\widehat{Y}^{*,\mathrm{m}}_{\hat{i}_p,s,q} = L_s \overline{\Psi}^{*,\mathrm{m}}_{\hat{i},s,q} + \Gamma_s \tilde{A}^p X^{\mathrm{m}}_{\hat{i},q}, \tag{26}$$

where the last subscripts are updated to reflect the q remaining columns. Accordingly we define using MATLAB notation $\widehat{Y}^{*,\mathrm{m}}_{\hat{i}_p,s,q} = \widehat{Y}^*_{\hat{i}_p,s,b}(:, 1:s:b)$, $\overline{\Psi}^{*,\mathrm{m}}_{\hat{i},s,q} = \overline{\Psi}^*_{\hat{i},s,b}(:, 1:s:b)$, and $X^{\mathrm{m}}_{\hat{i},q} = X_{\hat{i},b}(:, 1:s:b)$.

Rearranging (24) and (26) whilst post-multiplying the former by a transposed IV matrix $\mathcal{Z} \in \mathbb{R}^{n_z \times N}$ obtains

$$\begin{bmatrix} -L_s & I_{ls} \end{bmatrix} \underbrace{\begin{bmatrix} \Psi_{i,s,N} \\ Y_{i_p,s,N} \end{bmatrix}}_{=\mathcal{M}_1} \mathcal{Z}^\top = \Gamma_s \tilde{A}^p X_{i,N} \mathcal{Z}^\top + \mathcal{H}_s E_{i_p,s,N} \mathcal{Z}^\top, \tag{27}$$

$$\begin{bmatrix} -L_s & I_{ls} \end{bmatrix} \underbrace{\begin{bmatrix} \overline{\Psi}^{*,\mathrm{m}}_{\hat{i},s,q} \\ \widehat{Y}^{*,\mathrm{m}}_{\hat{i}_p,s,q} \end{bmatrix}}_{=\mathcal{M}_2} = \Gamma_s \tilde{A}^p X^{\mathrm{m}}_{\hat{i},q}, \tag{28}$$

where $\mathcal{M}_1$ and $\mathcal{M}_2$ are defined as shown for brevity.

Taking the limits $p, N \to \infty$ in accordance with Assumption 1 ensures that the right hand sides of (27) and (28) go to zero asymptotically. Intuitively, this can be understood because for the initial state contributions $\tilde{A}^p \to 0$ as $p \to \infty$ since all eigenvalues of $\tilde{A}$ lie strictly within the unit circle ($\rho < 1$). With respect to the noise term in (27), the fundamental property of an IV that it is not to be correlated with noise ensures that $E_{i_p,s,N} \mathcal{Z}^\top \to 0$ as $N \to \infty$.

In the aforementioned limits $p, N \to \infty$ the right hand sides of (27) and (28) become zero and $\mathcal{M}_1$ and $\mathcal{M}_2$ lie in the nullspace of $[-L_s\ I_{ls}]$. The fundamental idea behind DeePC is that if additionally $\Psi_{i,s,N} \mathcal{Z}^\top$ is full row rank[7] then this nullspace is completely spanned by the columnspace of $\mathcal{M}_1$. Consequently, there then exist linear combinations of $\mathcal{M}_1$, parameterized by the columns of a generic matrix $G^{\mathrm{IV}} \in \mathbb{R}^{n_z \times q}$, such that

$$\begin{aligned} &\lim_{p,N\to\infty} \mathcal{M}_1 G^{\mathrm{IV}} = \lim_{p,N\to\infty} \mathcal{M}_2, \\ &\lim_{p,N\to\infty} \begin{bmatrix} \Psi_{i,s,N} \\ Y_{i_p,s,N} \end{bmatrix} \mathcal{Z}^\top G^{\mathrm{IV}} = \lim_{p,N\to\infty} \begin{bmatrix} \overline{\Psi}^{*,\mathrm{m}}_{\hat{i},s,q} \\ \widehat{Y}^{*,\mathrm{m}}_{\hat{i}_p,s,q} \end{bmatrix}. \end{aligned} \tag{29}$$

Practical applications of (29) rely on a finite p and N, which obtains the below unified CL-DeePC formulation:

$$\begin{bmatrix} \Psi_{i,s,N} \\ Y_{i_p,s,N} \end{bmatrix} \mathcal{Z}^\top G^{\mathrm{IV}} = \begin{bmatrix} \overline{\Psi}^{\mathrm{m}}_{\hat{i},s,q} \\ \widehat{Y}^{\mathrm{IV,m}}_{\hat{i}_p,s,q} \end{bmatrix}. \tag{30}$$

The existence of a G^{IV} that satisfies (30) trivially follows from the full row rank of $\Psi_{i,s,N} \mathcal{Z}^\top$.

In the unified CL-DeePC predictor (30), s is the multi-step-ahead predictor length, and q is the number of sequential applications thereof, thus obtaining a total prediction length $f = sq$. The superscript m indicates that subsequent columns are shifted not by a single sample, as in Fig. 2, but by s samples.

The formulation provided by (30) unifies different potential choices of s, q and $\mathcal{Z}$ in a single framework. Note that if $s = 1$ the superscript m can be omitted and with $q = f$ one then obtains the CL-DeePC formulation using sequential single step-ahead predictions of (17). The superscript m can likewise be ignored if $q = 1$ such that with $s = f$ one obtains a generic IV-based DeePC formulation of the type introduced in van Wingerden et al. (2022). Formulations that do not employ IVs (i.e. $\mathcal{Z} = I_N$) like (13) ($s = f$, $q = 1$) and (14) ($s = 1$, $q = f$) lack this structural means of noise mitigation, potentially leading to detrimental noise sampling by the columns of $G = \mathcal{Z}^\top G^{\mathrm{IV}}$.

Remark 5. Note that DeePC with IVs as in van Wingerden et al. (2022) and Wang et al. (2023) is recovered from (30) by the single application ($q = 1$) of a multi-step-ahead predictor of length $s = f$. See the latter work for three examples of closed-loop suitable IV matrices $\mathcal{Z}$.

[7] Full row rank $\Psi_{i,s,N}$ should suffice because an appropriate choice of $\mathcal{Z}$ retains rank of $\Psi_{i,s,N} \mathcal{Z}^\top$ (see Section 3.4.1).

Remark 6. Note that the unified framework obtains causal predictors only when it employs single-step-ahead predictors (i.e. $s = 1$), which are inherently causal. For multi-step-ahead predictors ($s > 1$) the presented unified framework does not ensure causality. However, causality can be enforced by altering the result of an intermediate LQ-decomposition of the data as in Sader et al. (2025). Therein the authors furthermore show that this is conceptually equivalent to imposing a causality-informed constraint on the least-squares identification problem of an SPC-type predictor.

3.5.2. Consistency of the unified CL-DeePC predictor

For the unified formulation of a CL-DeePC predictor given by (30), we present the following theorem pertaining to its consistency when using closed-loop data.

Theorem 2. *Consider the well-posed closed-loop system from Theorem 1 to generate input–output data, with $\Psi_{i,s,N}$ full row rank by Assumption 2 and partially-known data matrix $\overline{\Psi}^{\mathrm{m}}_{i,s,q}$. Furthermore, choose $\mathcal{Z} \in \mathbb{R}^{n_z \times N}$ such that, with $p, N \to \infty$ as specified by Assumption 1, $\Psi_{i,s,N}\mathcal{Z}^\top$ is full row rank and $E_{i_p,s,N}\mathcal{Z}^\top \to 0$. Then the predictor $\widehat{Y}^{\mathrm{IV,m}}_{\hat{i}_p,s,q}$ specified by (30) is consistent, i.e.:*

$$\lim_{p,N\to\infty} \mathbb{E}\left[\widehat{Y}^{\mathrm{IV,m}}_{\hat{i}_p,s,q} - Y^{\mathrm{m}}_{\hat{i}_p,s,q}\right] = 0.$$

Proof. Proof of this result follows the proof of Theorem 1, only using a generic $s \in \mathbb{Z}_{>0}$ instead of $s = 1$ and replacing the notation f by q. □

Theorem 2 presents the consistency of predictors obtained by the unified CL-DeePC formulation (30). Suitable choices of the IV matrix $\mathcal{Z}$ retain data informativity and are uncorrelated with noise as stipulated by Theorem 2.

Remark 7. Note that with the choice $\mathcal{Z} = \Psi_{i,m,N}$, the condition $E_{i_p,s,N}\mathcal{Z}^\top \to 0$ is violated for $m > 1$, resulting in an inconsistent output predictor. Moreover, since $\Psi_{i,s,N}\mathcal{Z}^\top \in \mathbb{R}^{((p+s)r+pl)\times n_z}$ must also be full row rank such that $n_z \geq (p+s)r + pl$, this implies that for $s > 1$ the IV matrix $\mathcal{Z}$ must consist of more instruments than the past inputs and outputs contained by $\Psi_{i,N}$.

Remark 8. Without IVs (i.e. $\mathcal{Z} = I_N$), the use of a square and invertible data matrix $\Psi_{i,s,N} \in \mathbb{R}^{((p+s)r+pl)\times N}$ may seem appealing based on Remark 3. However, the dimensions of a square data matrix $\Psi_{i,s,N}$ violate Assumption 1. Since this assumption is a necessary assumption (see discussion in Chiuso (2007)), this implies inconsistency of the predictor based on Theorem 2. This highlights the use of IVs to obtain consistent output predictors.

The special case with $q = f$ sequential applications of a one-step-ahead predictor ($s = 1$) reduces (30) to (17), for which a computationally efficient implementation is presented in the subsequent section.

4. Computationally efficient CL-DeePC

This section presents an implementation of (17) that reduces the number of optimization variables to improve the computational efficiency of CL-DeePC, thus providing part of Contribution 4.

The use of such an efficient method can be understood by comparing the number of unknown, unequal optimization variables (in red) in Fig. 2, as summarized in Table 1 for the case with IVs. Table 1 shows that for CL-DeePC based on (17) the influence of the number of unknowns in G^{IV} is quite large due to the relatively large minimum number of instruments $n_z^{\min}$. To this end, this section will eliminate G^{IV} from the formulation provided by (17), considerably reducing the number of unknowns.

Table 1
Number of unequal unknowns in (17) and their origin: G^{IV}, inputs, and outputs. Full row rank of $\hat{\Sigma}_{\psi z}$ implies $n_z \geq n_z^{\min}$, which has a large influence on the total number of unknowns.

G^{IV}	inputs	outputs	total	$n_z^{\min}$
$n_z f$	fr	fl	$f(r+l+n_z)$	$p(r+l)+r$

In line with Remark 3, we choose $\mathcal{Z} \in \mathbb{R}^{n_z \times N}$ with $n_z = n_z^{\min}$ from Table 1 such that $\hat{\Sigma}_{\psi z}$ is square and invertible, and consequently, $G^{\mathrm{IV}} = \hat{\Sigma}_{\psi z}{}^{-1}\overline{\Psi}_{\hat{i},f}$ by (18). Substituting this result for G^{IV} in (17) yields

$$\widehat{Y}^{\mathrm{IV}}_{\hat{i}_p,f} = \hat{\Sigma}_{yz}\hat{\Sigma}_{\psi z}{}^{-1}\overline{\Psi}_{\hat{i},f}, \tag{31}$$

in which $\hat{\Sigma}_{yz} = Y_{i_p,N}\mathcal{Z}^\top$.

The sequential nature of CL-DeePC can be exploited by the subsequent columns of $\overline{\Psi}_{\hat{i},f}$ in (31). For columns indexed by $k \in [\hat{i}, \hat{i}_p + f)$ this obtains

$$\hat{y}_k = \begin{bmatrix}\tilde{\beta}_1 & \cdots & \tilde{\beta}_{p+1}\end{bmatrix}\underline{u}_{k-p,p+1} + \begin{bmatrix}\tilde{\theta}_1 & \cdots & \tilde{\theta}_p\end{bmatrix}\underline{\bar{y}}_{k-p,p}, \tag{32}$$

in which $\tilde{\beta}_j \in \mathbb{R}^{l\times r}$, and $\tilde{\theta}_j \in \mathbb{R}^{l\times l}$ are determined from $\hat{\Sigma}_{yz}\hat{\Sigma}_{\psi z}{}^{-1}$ in (31). Sequential application of the one-step-ahead predictor (32) leads to

$$\underline{\hat{y}}_{\hat{i}_p,f} = \begin{bmatrix}\widetilde{\mathcal{L}}^{\mathrm{u}} & \widetilde{\mathcal{G}}^{\mathrm{u}}\end{bmatrix}\begin{bmatrix}\underline{u}_{\hat{i},p} \\ \underline{u}_{\hat{i}_p,f}\end{bmatrix} + \begin{bmatrix}\widetilde{\mathcal{L}}^{\mathrm{y}} & \widetilde{\mathcal{G}}^{\mathrm{y}}\end{bmatrix}\begin{bmatrix}\underline{y}_{\hat{i},p} \\ \underline{\hat{y}}_{\hat{i}_p,f}\end{bmatrix}, \tag{33}$$

in which

$$\begin{bmatrix}\widetilde{\mathcal{L}}^{\mathrm{u}} & \widetilde{\mathcal{G}}^{\mathrm{u}}\end{bmatrix} = \begin{bmatrix} \tilde{\beta}_1 & \cdots & \tilde{\beta}_p & \tilde{\beta}_{p+1} & 0 & 0 & \cdots & 0 \\ 0 & \tilde{\beta}_1 & \cdots & \tilde{\beta}_p & \tilde{\beta}_{p+1} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \tilde{\beta}_1 & \cdots & \tilde{\beta}_p & \tilde{\beta}_{p+1} & 0 \\ 0 & \cdots & 0 & 0 & \tilde{\beta}_1 & \cdots & \tilde{\beta}_p & \tilde{\beta}_{p+1} \end{bmatrix},$$

$$\begin{bmatrix}\widetilde{\mathcal{L}}^{\mathrm{y}} & \widetilde{\mathcal{G}}^{\mathrm{y}}\end{bmatrix} = \begin{bmatrix} \tilde{\theta}_1 & \cdots & \tilde{\theta}_p & 0 & 0 & 0 & \cdots & 0 \\ 0 & \tilde{\theta}_1 & \cdots & \tilde{\theta}_p & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \tilde{\theta}_1 & \cdots & \tilde{\theta}_p & 0 & 0 \\ 0 & \cdots & 0 & 0 & \tilde{\theta}_1 & \cdots & \tilde{\theta}_p & 0 \end{bmatrix},$$

with $\widetilde{\mathcal{L}}^{\mathrm{u}} \in \mathbb{R}^{fl\times pr}$, $\widetilde{\mathcal{G}}^{\mathrm{u}} \in \mathbb{R}^{fl\times fr}$, $\widetilde{\mathcal{L}}^{\mathrm{y}} \in \mathbb{R}^{fl\times pl}$, and $\widetilde{\mathcal{G}}^{\mathrm{y}} \in \mathbb{R}^{fl\times fl}$.

Remark 9. The form of (33) is comparable to, e.g., the iterated step-ahead predictor in Chiuso et al. (2025), which considers cases without direct feedthrough ($\tilde{\beta}_{p+1} = 0$) and is also not restricted to open-loop data.

Notice that the predicted outputs $\underline{\hat{y}}_{\hat{i}_p,f}$ feature on both sides of (33). Solving for these outputs yields

$$\underline{\hat{y}}_{\hat{i}_p,f} = \begin{bmatrix}\mathcal{L}^{\mathrm{u}} & \mathcal{L}^{\mathrm{y}}\end{bmatrix}\begin{bmatrix}\underline{u}_{\hat{i},p} \\ \underline{y}_{\hat{i},p}\end{bmatrix} + \mathcal{G}^{\mathrm{u}}\underline{u}_{\hat{i}_p,f}, \tag{34}$$

in which $\mathcal{L}^{\mathrm{u}}$, $\mathcal{G}^{\mathrm{u}}$, and $\mathcal{L}^{\mathrm{y}}$ are uniquely defined by

$$\begin{bmatrix}\mathcal{L}^{\mathrm{u}} & \mathcal{G}^{\mathrm{u}} & \mathcal{L}^{\mathrm{y}}\end{bmatrix} = \left(I_{fl} - \widetilde{\mathcal{G}}^{\mathrm{y}}\right)^{-1}\begin{bmatrix}\widetilde{\mathcal{L}}^{\mathrm{u}} & \widetilde{\mathcal{G}}^{\mathrm{u}} & \widetilde{\mathcal{L}}^{\mathrm{y}}\end{bmatrix}. \tag{35}$$

The block-lower-triangular structure of $\widetilde{\mathcal{G}}^{\mathrm{y}}$ guarantees the invertibility of $I_{fl} - \widetilde{\mathcal{G}}^{\mathrm{y}}$ such that (35) can be solved directly to construct the predictor (34).

An efficient sequential procedure is also possible to solve (35) that exploits the structure of $I_{fl} - \widetilde{\mathcal{G}}^{\mathrm{y}}$. For this sequential solution procedure, define the f block-rows of $\begin{bmatrix}\widetilde{\mathcal{L}}^{\mathrm{u}} & \widetilde{\mathcal{G}}^{\mathrm{u}} & \widetilde{\mathcal{L}}^{\mathrm{y}}\end{bmatrix}$ and $\begin{bmatrix}\mathcal{L}^{\mathrm{u}} & \mathcal{G}^{\mathrm{u}} & \mathcal{L}^{\mathrm{y}}\end{bmatrix}$

by respectively $\tilde{\alpha}_j$, $\alpha_j \in \mathbb{R}^{l\times p(r+l)+fr}$, with j here representing the index of the block row: $j = 0, 1, \ldots, f-1$. Based on (35) it is shown in Appendix A that with $a = \max(1, j-p+1)$ the formulation

$$\alpha_j = \begin{cases} \tilde{\alpha}_j & \text{if } j = 0 \\ \tilde{\alpha}_j + \sum_{m=a}^{j} \tilde{\theta}_{p-j+m}\alpha_{m-1} & \text{if } j \geq 1 \end{cases}, \tag{36}$$

allows efficient sequential construction of $\begin{bmatrix}\mathcal{L}^{\mathrm{u}} & \mathcal{G}^{\mathrm{u}} & \mathcal{L}^{\mathrm{y}}\end{bmatrix}$ starting from $j = 0$.

With reference to (33) we note that $\widetilde{\mathcal{G}}^{\mathrm{u}}$ is block-lower-triangular, just like $I_{fl} - \widetilde{\mathcal{G}}^{\mathrm{y}}$. Hence, by (35), $\mathcal{G}^{\mathrm{u}}$ will also be block-lower-triangular. This block-lower-triangular structure enforces causality of the predictor (34). This is unlike DeePC, as shown by the fact that equivalent SPC methods do not enforce causality of the predictor.

In addition, from the subsequent section it will become clear that $\mathcal{G}^{\mathrm{u}}$ is not just block-lower-triangular, but also a block-Toeplitz matrix. This means that $\mathcal{G}^{\mathrm{u}}$ is fully parameterized by its leftmost block-column, allowing one to solve only for this part of $\mathcal{G}^{\mathrm{u}}$ in (35) using (36). The complete matrix $\mathcal{G}^{\mathrm{u}}$ can then be constructed from its leftmost block-column thereafter.

The subsequent section provides the remainder of Contribution 4 by demonstrating a specific equivalence between CL-DeePC and CL-SPC.

5. Specific equivalence to Closed-loop SPC

This section demonstrates an equivalence between CL-SPC as developed in Dong et al. (2008) and CL-DeePC based on (30) *with specific design choices*: $s = 1$ (recovering (17)) and $\mathcal{Z} = \Psi_{i,N}$. This equivalence is revealed based on the efficient CL-DeePC implementation discussed in the previous section, thus providing the remainder of Contribution 4. The CL-SPC algorithm is briefly explained first, based upon which the equivalence is demonstrated thereafter.

5.1. Closed-loop SPC

To understand this equivalence, consider the data Eqs. (8) and (9). As is possible with CL-DeePC, CL-SPC uses $s = 1$ to avoid closed-loop correlation between inputs and noise. CL-SPC estimates the dynamic matrix $\widetilde{L}_1$ by least squares regression on past data[8]:

$$\widehat{\widetilde{L}}_1 = \begin{bmatrix}\widehat{C\widetilde{\mathcal{K}}^{\mathrm{u}}} & \widehat{D} & \widehat{C\widetilde{\mathcal{K}}^{\mathrm{y}}}\end{bmatrix} = Y_{i_p,N}\Psi_{i,N}^{\dagger}. \tag{37}$$

Assuming that p is sufficiently large such that $\widetilde{A}^p = 0$, estimates of the predictor Markov parameters contained in $\widehat{\widetilde{L}}_1$ allow the construction of estimates of $\widetilde{\Gamma}_f\widetilde{\mathcal{K}}_p^{\mathrm{u}}$, $\widetilde{\Gamma}_f\widetilde{\mathcal{K}}_p^{\mathrm{y}}$ and $\widetilde{\mathcal{T}}_f^{\mathrm{u}}$, which make up $\widetilde{L}_f$, as well as $\widetilde{\mathcal{H}}_f$. In line with (9) this allows the construction of a predictor as

$$\begin{aligned}\underline{\hat{y}}_{\hat{i}_p,f} = &\begin{bmatrix}\widehat{\widetilde{\Gamma}_f\widetilde{\mathcal{K}}^{\mathrm{u}}} & \widehat{\widetilde{\mathcal{T}}_f^{\mathrm{u}}}\end{bmatrix}\begin{bmatrix}\underline{u}_{\hat{i},p} \\ \underline{u}_{\hat{i}_p,f}\end{bmatrix} + \\ &\begin{bmatrix}\widehat{\widetilde{\Gamma}_f\widetilde{\mathcal{K}}^{\mathrm{y}}} & (I_{fl} - \widehat{\widetilde{\mathcal{H}}}_f)\end{bmatrix}\begin{bmatrix}\underline{y}_{\hat{i},p} \\ \underline{\hat{y}}_{\hat{i}_p,f}\end{bmatrix}.\end{aligned} \tag{38}$$

Note that the predicted output is contained on both sides of this equation. To solve (38) for the predicted outputs first make note of the fact that (Houtzager et al., 2012)

$$\begin{bmatrix}\Gamma_f\widetilde{\mathcal{K}}^{\mathrm{u}} & \mathcal{T}_f^{\mathrm{u}} & \Gamma_f\widetilde{\mathcal{K}}^{\mathrm{y}}\end{bmatrix} = \widetilde{\mathcal{H}}_f^{-1}\begin{bmatrix}\widetilde{\Gamma}_f\widetilde{\mathcal{K}}^{\mathrm{u}} & \widetilde{\mathcal{T}}_f^{\mathrm{u}} & \widetilde{\Gamma}_f\widetilde{\mathcal{K}}^{\mathrm{y}}\end{bmatrix}, \tag{39}$$

as is also visible from the combination of (8) and (9) for $s = f$. Using estimates in (39) to solve (38) for the output predictions (which can be done efficiently in a sequential manner as with CL-DeePC) yields

$$\underline{\hat{y}}_{\hat{i}_p,f} = \begin{bmatrix}\widehat{\Gamma_f\widetilde{\mathcal{K}}^{\mathrm{u}}} & \widehat{\Gamma_f\widetilde{\mathcal{K}}^{\mathrm{y}}}\end{bmatrix}\begin{bmatrix}\underline{u}_{\hat{i},p} \\ \underline{y}_{\hat{i},p}\end{bmatrix} + \widehat{\mathcal{T}_f^{\mathrm{u}}}\underline{u}_{\hat{i}_p,f}, \tag{40}$$

which is in line with (8) and can be used in a receding horizon optimal control framework.

5.2. Equivalence between CL-DeePC and CL-SPC

This section formalizes an equivalence between CL-SPC and a form of CL-DeePC by means of the below lemma.

Lemma 3. *Given full row rank $\Psi_{i,N}$, the predictors (17) and (40) of respectively CL-DeePC with $\mathcal{Z} = \Psi_{i,N}$ and CL-SPC are equivalent.*

Proof. For (17) there exists the equivalent, but computationally more efficient implementation from Section 4, which obtains the predictor (34). By (31) and (32) with full row rank $\mathcal{Z} = \Psi_{i,N}$, such an efficient CL-DeePC implementation uses $\begin{bmatrix}\tilde{\beta}_1 & \cdots & \tilde{\beta}_{p+1} & \tilde{\theta}_1 & \cdots & \tilde{\theta}_p\end{bmatrix} = Y_{i_p,N}\Psi_{i,N}^{\dagger}$. Comparing this to (37) shows that the building blocks $\tilde{\beta}_j$ and $\tilde{\theta}_j$ in this CL-DeePC implementation are equal to the estimated predictor Markov parameters that CL-SPC uses:

$$\tilde{\beta}_j = \begin{cases} \widehat{C\tilde{A}^{p-j}\tilde{B}} & 1 \leq j \leq p, \\ \widehat{D} & j = p+1, \end{cases} \tag{41a}$$

$$\tilde{\theta}_j = \widehat{C\tilde{A}^{p-j}K} \qquad 1 \leq j \leq p. \tag{41b}$$

Since CL-SPC effectively assumes $\tilde{A}^p = 0$, (41) then entails an equivalence between the constructed matrices in CL-SPC and the efficient sequential CL-DeePC method that is succinctly described by

$$\begin{bmatrix}\widetilde{\mathcal{L}}^{\mathrm{u}} & \widetilde{\mathcal{G}}^{\mathrm{u}} & \widetilde{\mathcal{L}}^{\mathrm{y}} & (I_{fl} - \widetilde{\mathcal{G}}^{\mathrm{y}})\end{bmatrix} = \begin{bmatrix}\widehat{\widetilde{\Gamma}_f\widetilde{\mathcal{K}}^{\mathrm{u}}} & \widehat{\widetilde{\mathcal{T}}_f^{\mathrm{u}}} & \widehat{\widetilde{\Gamma}_f\widetilde{\mathcal{K}}^{\mathrm{y}}} & \widehat{\widetilde{\mathcal{H}}}_f\end{bmatrix}. \tag{42}$$

It follows from (35), (39), and (42) that, likewise,

$$\begin{bmatrix}\mathcal{L}^{\mathrm{u}} & \mathcal{G}^{\mathrm{u}} & \mathcal{L}^{\mathrm{y}}\end{bmatrix} = \begin{bmatrix}\widehat{\Gamma_f\widetilde{\mathcal{K}}^{\mathrm{u}}} & \widehat{\mathcal{T}_f^{\mathrm{u}}} & \widehat{\Gamma_f\widetilde{\mathcal{K}}^{\mathrm{y}}}\end{bmatrix}. \tag{43}$$

This concludes the proof since (43) entails that (34), and therefore (17), is equivalent to (40). □

The foregoing lemma indicates that for CL-SPC and CL-DeePC along the lines of (17) with $\mathcal{Z} = \Psi_{i,N}$, the predictors are equivalent. Hence, the analytical solutions of these control methods are also equivalent. In practice we note that numerical differences may arise from implementations with optimizers due to the relatively large number of optimization variables introduced by G^{IV} in (17), as discussed in Section 4. Notwithstanding the specific equivalence presented here we emphasize the potential advantages, discussed in Section 2.2, of a direct closed-loop DDPC method like CL-DeePC over an indirect method like CL-SPC. Moreover, we note that CL-DeePC predictors are more generic than the CL-SPC predictor (40) because the former type considers different options as IV matrix $\mathcal{Z}$ (see Remark 5), and facilitates the use of multi-step-ahead predictors as per (30).

6. Results

This section presents simulation results of DeePC and CL-DeePC that facilitate the comparison of their performance, thereby providing Contribution 5. We will first describe the simulation setup before presenting a reference tracking example. A subsequent demonstration of closed-loop correlation between inputs and noise is followed by a consistency analysis. Lastly, a parametric sensitivity analysis is performed.

[8] To incorporate direct-feedthrough one may apply $B = \tilde{B} + KD$ to the recursive component of the CL-SPC algorithm of Dong et al. (2008), which does not consider direct-feedthrough.

6.1. Simulation setup

This section presents the simulation setup. The simulated discrete plant is a marginally stable fifth-order model of the form (1) that represents two circular plates that are spun by a motor with non-rigid shafts with (van Wingerden et al., 2022)

$$A = \begin{bmatrix} 4.40 & 1 & 0 & 0 & 0 \\ -8.09 & 0 & 1 & 0 & 0 \\ 7.83 & 0 & 0 & 1 & 0 \\ -4.00 & 0 & 0 & 0 & 1 \\ 0.86 & 0 & 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0.00098 \\ 0.01299 \\ 0.01859 \\ 0.0033 \\ -0.00002 \end{bmatrix}, C = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}^\top,$$
$$K = \begin{bmatrix} 2.3 & -6.64 & 7.515 & -4.0146 & 0.86336 \end{bmatrix}^\top, D = 0.$$

Three different predictive controllers are implemented. The optimal control problem solved by each of these controllers involves (12a)–(12b) but differs in terms of the employed predictor. For this purpose, one controller uses DeePC with IVs as proposed in van Wingerden et al. (2022)[9]. Another uses CL-DeePC with IVs, implemented as described in Section 4 to reduce computation time. As a benchmark, an oracle model predictive controller is used. This oracle employs the predictor

$$\underline{\hat{y}}_{\hat{i}_p,f} = \Gamma_f x_{\hat{i}_p} + \mathcal{T}_f^{\mathrm{u}} \underline{u}_{\hat{i}_p,f},$$

for which it has complete knowledge of the initial state $x_{\hat{i}_p}$ and the system matrices Γ_f and $\mathcal{T}_f^{\mathrm{u}}$ (as defined in Section 1.2). The exact form of the cost function that is minimized in (12a) by all of the controllers is given by

$$\mathcal{J}(\underline{u}_{\hat{i}_p,f}, \underline{\hat{y}}_{\hat{i}_p,f}) = \|\underline{\hat{y}}_{\hat{i}_p,f} - \underline{y}^{\mathrm{ref}}_{\hat{i}_p,f}\|_Q^2 + \|\Delta \underline{u}_{\hat{i}_p,f}\|_{R^\Delta}^2, \tag{44}$$

in which $\Delta \underline{u}_{\hat{i}_p,f} = \underline{u}_{\hat{i}_p,f} - \underline{u}_{\hat{i}_p-1,f}$, $\underline{y}^{\mathrm{ref}}_{\hat{i}_p,f}$ contains a future output reference trajectory, $Q \in \mathbb{R}^{fl \times fl}$ is a positive semi-definite weighting matrix, and $R^\Delta \in \mathbb{R}^{fr \times fr}$ is a positive definite weighting matrix.

All of the simulations are carried out in MATLAB[10], using CasADi (Andersson et al., 2019) to formulate a quadratic program that is solved with IPOPT (Wächter & Biegler, 2006). Unless indicated otherwise $p = f = 20$, the employed cost function weights of (44) are $Q = 100$, and $R^\Delta = 10$, and employed constraints are $|u_k - u_{k-1}| \leq 3.75$, $|u_k| \leq 15$, $|y_k| < 1000$. If no solution is found, to prevent infeasibility these constraints are relaxed using quadratically penalized slack variables (with weighting $\lambda = 10^{15}$) and the problem is solved again. This appears to be necessary with DeePC, particularly for $\bar{N} < 200$.

All simulations use a square wave reference signal that varies between 0 and 100 with a period of 200 time steps. Simulations start by data collection in open-loop to initialize the data-driven controllers, which are then active together with the oracle for 1800 time steps. Closed-loop correlation between inputs and noise is induced by operating the data-driven controllers fully adaptively. As mentioned in Section 1 this means that the data matrices are updated at every time step after their initialization. With regards to Assumption 2, we seek to avoid problematic conditions mentioned in Remark 2. Furthermore, validation of Assumption 2 was performed for the parametric sensitivity analysis carried out in Section 6.3.3 to find that the assumption was always satisfied. Unless otherwise stated, the input variance for the open-loop trajectories is $\Sigma(u_k) = 1$ and the innovation noise has a variance of $\Sigma(e_k) = 1$ throughout.

[9] The implementation relies on an equivalent, but computationally more efficient SPC formulation.

[10] The employed code and resulting data can respectively be found via doi.org/10.5281/zenodo.10573259 and doi.org/10.5281/zenodo.10573874.

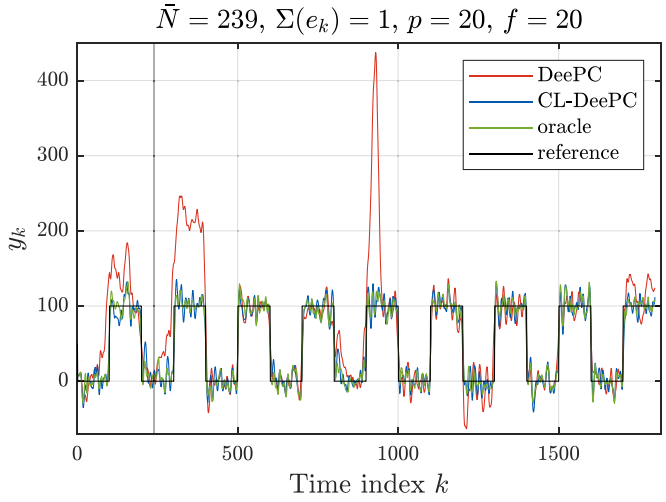


Fig. 4. Reference tracking by adaptive DeePC and CL-DeePC using IVs. After the vertical line at $\bar{N}$ all used data originates from operation in closed-loop. CL-DeePC displays superior reference tracking performance with respect to DeePC, in part due to a closed-loop identification issue.

6.2. Example of a reference tracking case

To compare the performance of DeePC and CL-DeePC in an adaptive setting this section presents a reference tracking example by Fig. 4. CL-DeePC clearly manages considerably better reference tracking performance w.r.t. DeePC, the former performing comparably to the oracle. At first, all of the data that the data-driven controllers rely on derives from open-loop operation. Due to the adaptive nature of the controller implementation, after $\bar{N}$ time steps, all of the employed data derives from closed-loop operation. The reference tracking ability of DeePC appears to decrease as the amount of employed closed-loop data increases. Thereafter, DeePC only manages to track the reference well intermittently. This intermittency may be understood by the following process. First, akin to cases where the data matrices are not updated at every time step, closed-loop identification bias results in large reference tracking errors (see also Dinkla et al. (2023)). In this work, wherein the data matrices are updated at each time step, these large tracking errors can momentarily cause increased excitation and signal-to-noise ratio of the employed data. This thereafter temporarily improves the obtained reference tracking performance, until the identification issue leads to worse tracking again, and the process repeats itself.

6.2.1. Correlation between inputs and noise

This section demonstrates the existence of correlation between inputs and preceding noise during closed-loop operation in an adaptive setting by means of simulations. Following the preceding consistency proof in Section 3.4.3, the correlation matrix of interest is given by $E_{i_p,s,N}\begin{bmatrix} U_{i,p,N}^\top & U_{i_p,s,N}^\top \end{bmatrix}$, with $s = f$ for DeePC, and $s = 1$ for CL-DeePC. Fig. 5 shows the mean of this correlation matrix based on the closed-loop data of 120 different noise realizations. Based on the figure, between DeePC and CL-DeePC, only DeePC experiences the correlation that it induces between inputs and noise. Note that the stochastic variability of subsequent control policies, which arises from the adaptive implementation in the presence of noise, is insufficient to mitigate the input-noise correlation experienced by DeePC.

6.2.2. Consistency analysis

This section demonstrates the consistency (or lack thereof) of the estimators employed by DeePC and CL-DeePC as a result of

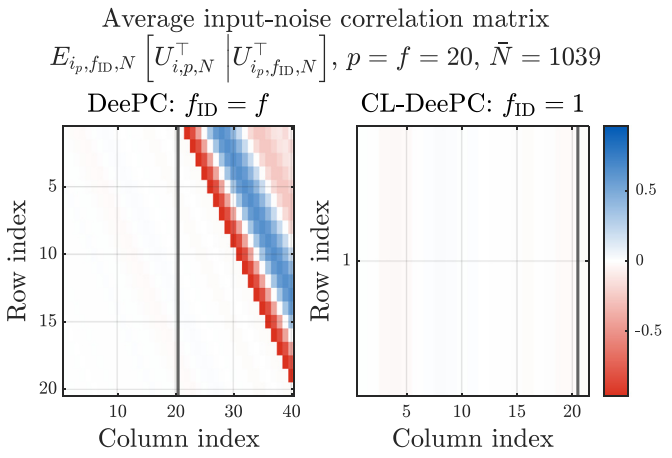


Fig. 5. Noise-input correlation matrix for DeePC and CL-DeePC averaged over the closed-loop data from 120 noise realizations. In contrast to DeePC, CL-DeePC makes use of input data that is uncorrelated with preceding noise.

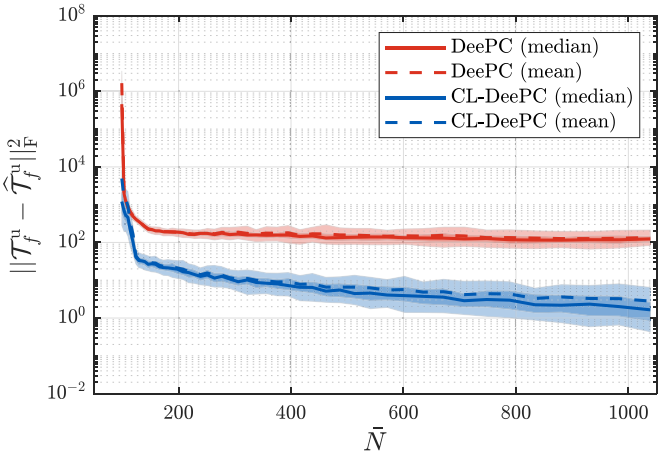


Fig. 6. Bias of the implicitly estimated matrix $\mathcal{T}_f^{\mathrm{u}}$ based exclusively on adaptive closed-loop operation. Shaded regions indicate the 10th, 30th, 70th and 90th percentiles of 120 simulations with different noise realizations.

experienced closed-loop input-noise correlation. Along the lines of the consistency analysis in Section 3.4.3 and Dinkla et al. (2023) it is to be expected that the implicit matrix estimate of $\mathcal{T}_s^{\mathrm{u}}$ is inconsistent for DeePC ($s=f$) and consistent for CL-DeePC ($s=1$). For a fair comparison, the error in $\widehat{\mathcal{T}}_f^{\mathrm{u}}$ is shown for both controllers by Fig. 6. Section 4 is followed to construct this estimate for CL-DeePC.

As expected, as the number of employed past data points $\bar{N}$ increases (together with the number of columns $N=\bar{N}-p-s+1$), the bias of the CL-DeePC estimate keeps decreasing, which indicates that the employed estimate is indeed consistent. In contrast, for DeePC the bias does not keep decreasing noticeably beyond around $\bar{N}=400$, indicating that the implicitly employed estimate $\widehat{\mathcal{T}}_f^{\mathrm{u}}$ is inconsistent.

6.3. Parametric sensitivity analysis

This section performs a parametric sensitivity study on the reference tracking performance of DeePC, CL-DeePC, and the oracle. Investigated parameters are the number of past data points $\bar{N}$, the innovation noise variance $\Sigma(e_k)$, and the window lengths $p=f$. In addition, computation times of the employed implementations of these algorithms are shown for

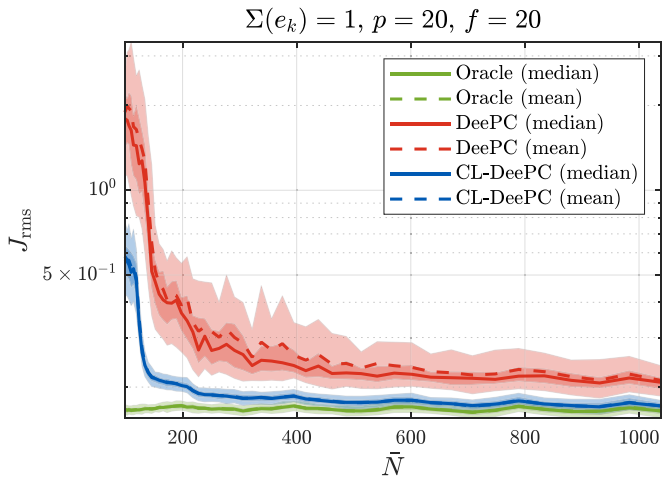


Fig. 7. Effect of $\bar{N}$ on reference tracking performance. Shaded regions indicate the 10th, 30th, 70th and 90th percentiles of 120 simulations.

varying $p=f$. As a measure of the reference tracking performance a scaled root mean square of the reference tracking error is used: $J_{\mathrm{rms}}=\sqrt{\left(\sum_k (y_k^{\mathrm{ref}})^2\right)^{-1}\sum_k (y_k-y_k^{\mathrm{ref}})^2}$. To reflect purely the effect on performance during adaptive closed-loop operation, the first $\bar{N}$ control actions are excluded. For the performance metric J_{rms} both the mean and median over 120 noise realizations will be shown to convey skewness of the distribution of the obtained performance. Note that it is possible for DeePC and CL-DeePC to outperform the oracle because the oracle does also not account for noise.

6.3.1. Number of past data samples: $\bar{N}$

The effect of a varying number of past data samples $\bar{N}$ is shown in Fig. 7. Slight fluctuations of the displayed oracle performance are an artifact that is attributable to the exclusion of the first $\bar{N}$ control actions in the calculation of J_{rms} to exclude control actions that are based on open-loop data. Using CL-DeePC the reference tracking performance approaches the oracle's performance at a relatively low $\bar{N}$. For DeePC the picture is different, with the median reference tracking performance remaining 26% higher than its oracle counterpart at $\bar{N}=1039$. By comparison this value is only 4% for CL-DeePC.

Fig. 7 demonstrates that compared to DeePC, CL-DeePC is generally able to use fewer past data samples to obtain as good or better performance, hence referred to as better sample efficiency. Compared to DeePC, the better sample efficiency of CL-DeePC is attributable to the use of a shorter future prediction window length for identification ($s=1$ as opposed to $s=f$). This leaves more columns $N=\bar{N}-p-s+1$ to approximate the relevant correlation matrices that are used implicitly by both DeePC and CL-DeePC. In addition, the discussed closed-loop identification issue entails that even if $\bar{N}$, and therefore N, is large such that these correlation matrices are approximated well, CL-DeePC outperforms DeePC.

6.3.2. Noise level: $\Sigma(e_k)$

The effect of the noise level, as quantified by a varying innovation noise variance $\Sigma(e_k)$, is shown in Fig. 8. In the absence of noise, DeePC and Model Predictive Control (MPC) are equivalent (Coulson et al., 2019a). In the noiseless case, the closed-loop identification issue does not arise so the performance of the three algorithms is identical. As the noise level increases, correlation between inputs and preceding noise increases because $|e_k|$ is typically larger, and more control effort is needed to perform

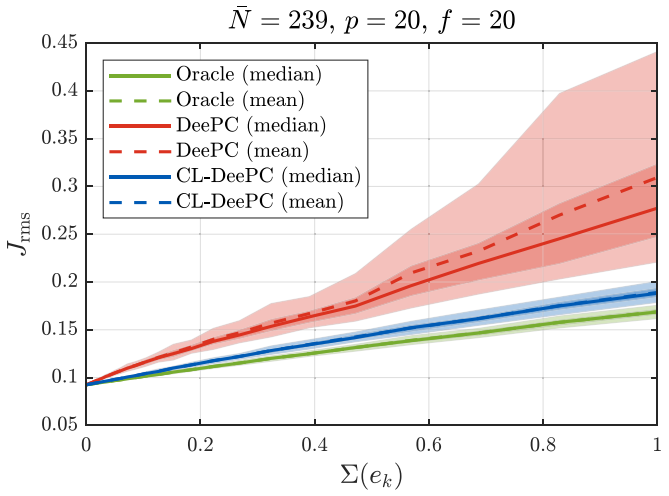


Fig. 8. Effect of $\Sigma(e_k)$ on reference tracking performance. Shaded regions indicate the 10th, 30th, 70th and 90th percentiles over 120 simulations.

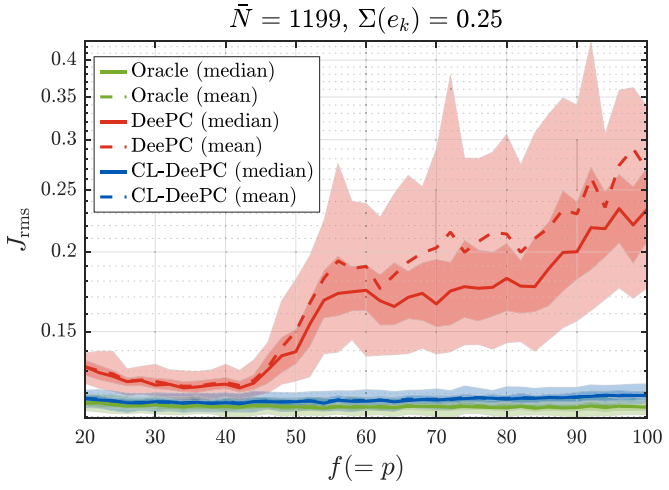


Fig. 9. Effect of $f = p$ on reference tracking performance. Shaded regions indicate the 10th, 30th, 70th and 90th percentiles of 120 simulations.

noise rejection. Consequently, the closed-loop identification issue becomes more troublesome for DeePC at higher noise levels. Note that the performance of all methods decreases with an increasing noise level because all methods lack the ability to immediately compensate noise disturbances. Using the slopes of the median performance as a measure for the susceptibility to noise-induced performance deterioration, CL-DeePC is 48% less susceptible to noise-induced performance deterioration compared to DeePC.

6.3.3. Data window lengths: $p = f$

This section investigates the effect of the window lengths with $p = f$. The choice $p = f$ is made for the sake of simplicity and is further motivated by its frequent use in the closely related field of subspace identification (van der Veen et al., 2013). We discern two competing mechanisms to explain the results shown in Fig. 9 for CL-DeePC and an additional detrimental effect on performance for DeePC. Increasing p is beneficial in terms of reducing the bias of the predictor. However, as with increasing s, increasing p entails implicitly estimating more parameters of the predictor, thereby increasing its variance. This is more pronounced for DeePC since $s = f$ when compared to CL-DeePC for

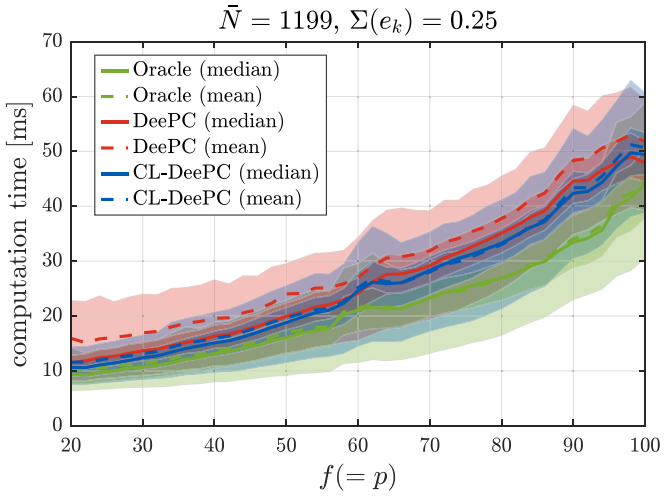


Fig. 10. Effect of $f = p$ on computation time of the IPOPT solver, as implemented in CasADi with MATLAB 2021b. Shaded regions indicate the 10th and 90th percentiles of 120 simulations.

which $s = 1$. The shorter future window length used for identification s is also beneficial for CL-DeePC because a higher degree of collinearity may be expected for larger s, potentially leading to ill-conditioning of the identification task (Chiuso & Picci, 2004). Together with the aforementioned closed-loop identification issue these effects induce up to a 49% lower reference tracking cost at $p = f = 100$ for CL-DeePC compared to DeePC. Moreover, note that the performance of DeePC deteriorates rapidly with increasing $p = f$, whereas the impact on the performance of CL-DeePC is far less pronounced.

Another relevant measure to quantify performance of the different controllers is to consider the computation times they require. To this end, the effect of varying windows $p = f$ on the time taken by the solver for different controllers is shown in Fig. 10. Each time-marching simulation was executed on a single core of an Intel Xeon E5-6248R 24-core 3.0 GHz CPU. As is to be expected, the computation time of each optimal control problem increases with increasing $f = p$. Moreover, one may observe that the employed efficient CL-DeePC implementation from Section 4 appears to typically be slightly more efficient than the employed efficient DeePC implementation (i.e. SPC). Such an effect can be explained by the larger degree of sparsity that the efficient CL-DeePC implementation introduces in the optimal control problem as a result of a causal predictor.

7. Conclusion

This article establishes CL-DeePC to address a closed-loop identification issue that is inherent to adaptive DeePC implementations. By incorporating IVs to mitigate noise it is shown that CL-DeePC employs a consistent output predictor when relying on data obtained in closed-loop. Moreover, to limit the number of optimization variables, an efficient sequential procedure is established to construct the output predictor in practice. This sequential procedure reveals the equivalence of the developed CL-DeePC method and CL-SPC.

Simulations confirm that CL-DeePC relies on data that does not exhibit correlation between inputs and preceding noise, facilitating the use of a consistent output predictor. Furthermore simulations demonstrate superior reference tracking performance of CL-DeePC compared to DeePC. A sensitivity analysis illustrates

that CL-DeePC is more sample efficient than DeePC, less sensitive to high noise levels, and performs better for a wide range of window lengths $p = f$.

Given the practical difficulties posed to persistency of excitation, outlined in Remark 2, with closed-loop data and adaptive implementations, future work may investigate optimal adaptive and closed-loop experiment design for DDPC applications. Furthermore, while this work has considered a single specific choice of IV ($\mathcal{Z} = \Psi_{i,N}$), future work may consider different IVs in CL-DeePC to further reduce the effects of noise. Moreover, where simulations have employed the computationally efficient implementation of (17), future work may further explore the merits of different settings (not $s = 1$ and $q = f$) of the unified CL-DeePC approach provided by (30). Other interesting avenues of investigation include the reduction of the number of optimization variables akin to γ-DDPC (Breschi et al., 2023), and the potential use of regularization to handle noise within the CL-DeePC framework. Using such techniques, and those introduced in this article, it furthermore remains to be shown under what conditions CL-DeePC may outperform CL-SPC.

Appendix A. Recursive inversion procedure

This section derives the recursive inversion procedure represented by (36) to solve (35).

To this end, rewriting (35) without taking an inverse obtains

$$\left(I_{fl} - \widetilde{\mathcal{G}}^{\mathrm{y}}\right)\begin{bmatrix}\mathcal{L}^{\mathrm{u}} & \mathcal{G}^{\mathrm{u}} & \mathcal{L}^{\mathrm{y}}\end{bmatrix} = \begin{bmatrix}\widetilde{\mathcal{L}}^{\mathrm{u}} & \widetilde{\mathcal{G}}^{\mathrm{u}} & \widetilde{\mathcal{L}}^{\mathrm{y}}\end{bmatrix},$$

which with the definitions of α_j, $\tilde{\alpha}_j$, and $\tilde{\theta}_j$ from Section 4 amounts to

$$\left[\begin{array}{cccccccc} I_l & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ \hline -\tilde{\theta}_p & I_l & 0 & 0 & 0 & 0 & \cdots & 0 \\ -\tilde{\theta}_{p-1} & -\tilde{\theta}_p & I_l & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & 0 & 0 & \cdots & 0 \\ -\tilde{\theta}_1 & -\tilde{\theta}_2 & \cdots & -\tilde{\theta}_p & I_l & 0 & \cdots & 0 \\ \hline 0 & -\tilde{\theta}_1 & -\tilde{\theta}_2 & \cdots & -\tilde{\theta}_p & I_l & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -\tilde{\theta}_1 & -\tilde{\theta}_2 & \cdots & -\tilde{\theta}_p & I_l \end{array}\right]\left[\begin{array}{c}\alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_p \\ \alpha_{p+1} \\ \vdots \\ \alpha_{f-1}\end{array}\right] = \left[\begin{array}{c}\tilde{\alpha}_0 \\ \tilde{\alpha}_1 \\ \tilde{\alpha}_2 \\ \vdots \\ \tilde{\alpha}_p \\ \tilde{\alpha}_{p+1} \\ \vdots \\ \tilde{\alpha}_{f-1}\end{array}\right]. \tag{A.1}$$

From (A.1) it is possible to distinguish three different kind of conditions, which are described by

$$\tilde{\alpha}_j = \begin{cases} \alpha_j & \text{if } j = 0 \\ \alpha_j - \displaystyle\sum_{m=1}^{j} \tilde{\theta}_{p-j+m}\alpha_{m-1} & \text{if } 1 \le j \le p \\ \alpha_j - \displaystyle\sum_{m=j-p+1}^{j} \tilde{\theta}_{p-j+m}\alpha_{m-1} & \text{if } j \ge p+1 \end{cases}.$$

Solving this for α_j, and defining $a = \max(1, j-p+1)$ to describe the two bottom conditions with a single sum obtains

$$\alpha_j = \begin{cases} \tilde{\alpha}_j & \text{if } j = 0 \\ \tilde{\alpha}_j + \displaystyle\sum_{m=a}^{j} \tilde{\theta}_{p-j+m}\alpha_{m-1} & \text{if } j \ge 1 \end{cases}. \tag{36}$$

References

Andersson, J. A. E., Gillis, J., Horn, G., Rawlings, J. B., & Diehl, M. (2019). CasADi: A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, *11*(1), 1–36.

Baros, S., Chang, C. Y., Colón-Reyes, G. E., & Bernstein, A. (2022). Online data-enabled predictive control. *Automatica*, *138*, Article 109926.

Bauer, D., & Ljung, L. (2002). Some facts about the choice of the weighting matrices in Larimore type of subspace algorithms. *Automatica*, *38*(5), 763–773.

Berberich, J., Köhler, J., Müller, M. A., & Allgöwer, F. (2021). Data-driven model predictive control with stability and robustness guarantees. *IEEE Transactions on Automatic Control*, *66*(4), 1702–1717.

Breschi, V., Chiuso, A., & Formentin, S. (2023). Data-driven predictive control in a stochastic setting: A unified framework. *Automatica*, *152*, Article 110961.

Chiuso, A. (2006). Asymptotic equivalence of certain closed loop subspace identification methods. *IFAC Proceedings Volumes*, *39*(1), 297–302.

Chiuso, A. (2007). The role of vector autoregressive modeling in predictor-based subspace identification. *Automatica*, *43*(6), 1034–1048.

Chiuso, A., Fabris, M., Breschi, V., & Formentin, S. (2025). Harnessing uncertainty for a separation principle in direct data-driven predictive control. *Automatica*, *173*, Article 112070.

Chiuso, A., & Picci, G. (2004). On the ill-conditioning of subspace identification with inputs. *Automatica*, *40*(4), 575–589.

Coulson, J., Lygeros, J., & Dörfler, F. (2019a). Data-enabled predictive control: In the shallows of the DeePC. In *2019 18th European control conference* (pp. 307–312).

Coulson, J., Lygeros, J., & Dörfler, F. (2019b). Regularized and distributionally robust data-enabled predictive control. In *2019 IEEE 58th conference on decision and control* (pp. 2696–2701).

Dinkla, R., Mulders, S. P., van Wingerden, J. W. ., & Oomen, T. (2023). Closed-loop aspects of data-enabled predictive control. *IFAC-PapersOnLine*, *56*(2), 1388–1393.

Dong, J., Verhaegen, M., & Holweg, E. (2008). Closed-loop subspace predictive control for fault tolerant MPC design. In *IFAC proceedings volumes*: *Vol. 41*, (pp. 3216–3221).

Dörfler, F., Coulson, J., & Markovsky, I. (2023). Bridging direct and indirect data-driven control formulations via regularizations and relaxations. *IEEE Transactions on Automatic Control*, *68*(2), 883–897.

Favoreel, W., De Moor, B., & Gevers, M. (1999). SPC: subspace predictive control. *IFAC Proceedings Volumes*, *32*(2), 4004–4009.

Fiedler, F., & Lucia, S. (2021). On the relationship between data-enabled predictive control and subspace predictive control. In *2021 European control conference* (pp. 222–229). Rotterdam, Netherlands: IEEE.

Gevers, M. (2005). Identification for control: From the early achievements to the revival of experiment design. *European Journal of Control*, *11*(4), 335–352.

Gilson, M., & van den Hof, P. M. J. (2005). Instrumental variable methods for closed-loop system identification. *Automatica*, *41*(2), 241–249.

Hallouzi, R., & Verhaegen, M. (2008). Persistency of excitation in subspace predictive control. *IFAC Proceedings Volumes*, *41*(2), 11439–11444.

Hjalmarsson, H. (2005). From experiment design to closed-loop control. *Automatica*, *41*(3), 393–438.

Hou, Z. S., & Wang, Z. (2013). From model-based control to data-driven control: Survey, classification and perspective. *Information Sciences*, *235*, 3–35.

Houtzager, I., van Wingerden, J. W., & Verhaegen, M. (2012). Recursive predictor-based subspace identification with application to the real-time closed-loop tracking of flutter. *IEEE Transactions on Control Systems Technology*, *20*(4), 934–949.

Huang, L., Coulson, J., Lygeros, J., & Dörfler, F. (2019). Data-enabled predictive control for grid-connected power converters. In *2019 IEEE 58th conference on decision and control* (pp. 8130–8135).

Huang, L., Zhen, J., Lygeros, J., & Dörfler, F. (2023). Robust data-enabled predictive control: Tractable formulations and performance guarantees. *IEEE Transactions on Automatic Control*, *68*(5), 3163–3170.

Jansson, H., & Hjalmarsson, H. (2005). Optimal experiment design in closed loop. *IFAC Proceedings Volumes*, *38*(1), 488–493.

Ljung, L. (1999). *System identification: theory for the user* (2nd ed.). Prentice Hall.

Ljung, L., & McKelvey, T. (1996). Subspace identification from closed loop data. *Signal Processing*, *52*(2), 209–215.

Markovsky, I., Huang, L., & Dörfler, F. (2023). Data-driven control based on the behavioral approach: From theory to applications in power systems. *IEEE Control Systems Magazine*, *43*(5), 28–68.

Sader, M., Wang, Y., Huang, D., Shang, C., & Huang, B. (2025). Causality-informed data-driven predictive control. *IEEE Transactions on Control Systems Technology*, 1–8.

Sassella, A., Breschi, V., & Formentin, S. (2022a). Data-driven design of explicit predictive controllers. In *2022 IEEE 61st conference on decision and control* (pp. 2821–2826).

Sassella, A., Breschi, V., & Formentin, S. (2022b). Noise handling in data-driven predictive control: a strategy based on dynamic mode decomposition. In *Proceedings of the 4th annual learning for dynamics and control conference* (pp. 74–85). PMLR.

Sassella, A., Breschi, V., & Formentin, S. (2023). A practitioner's guide to noise handling strategies in data-driven predictive control. *IFAC-PapersOnLine*, *56*(2), 1382–1387.
Shi, J., Lian, Y., & Jones, C. N. (2024). Efficient recursive data-enabled predictive control. In *2024 European control conference* (pp. 880–887).
Söderström, T., & Stoica, P. (1989). *System identification*. Prentice Hall.
Van Overschee, P., & De Moor, B. (1997). Closed loop subspace system identification. In *Proceedings of the 36th IEEE conference on decision and control*: *Vol. 2*, (pp. 1848–1853).
van der Veen, G., van Wingerden, J. W., Bergamasco, M., Lovera, M., & Verhaegen, M. (2013). Closed-loop subspace identification methods: An overview. *IET Control Theory & Applications*, *7*(10), 1339–1358.
Verhaegen, M., & Verdult, V. (2007). *Filtering and system identification: a least squares approach*. Cambridge ; New York: Cambridge University Press.
Verheijen, P. C. N., Breschi, V., & Lazar, M. (2023). Handbook of linear data-driven predictive control: Theory, implementation and design. *Annual Reviews in Control*, *56*, Article 100914.
Wächter, A., & Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, *106*(1), 25–57.
Wang, Y., Qiu, Y., Sader, M., Huang, D., & Shang, C. (2023). Data-driven predictive control using closed-loop data: An instrumental variable approach. *IEEE Control Systems Letters*, *7*, 3639–3644.
Willems, J. C., Rapisarda, P., Markovsky, I., & De Moor, B. L. (2005). A note on persistency of excitation. *Systems & Control Letters*, *54*(4), 325–329.
van Wingerden, J. W., Mulders, Sebastiaan P., Dinkla, Rogier, Oomen, Tom, & Verhaegen, Michel (2022). Data-enabled predictive control with instrumental variables: The direct equivalence with subspace predictive control. In *2022 IEEE 61st conference on decision and control* (pp. 2111–2116).
Yin, M., Iannelli, A., & Smith, R. S. (2023). Maximum likelihood estimation in data-driven modeling and control. *IEEE Transactions on Automatic Control*, *68*(1), 317–328.

Rogier Dinkla obtained his B.Sc. degree in Mechanical Engineering (2016) and M.Sc. degree in Systems and Control (2021) at Delft University of Technology (TU Delft) in The Netherlands. He is currently pursuing a Ph.D. at the Delft Center for Systems and Control at the TU Delft. The primary focus of his research concerns closed-loop aspects of data-driven predictive control.

Tom Oomen is a full professor in the Department of Mechanical Engineering at Eindhoven University of Technology, The Netherlands. He has also held academic positions at KTH Royal Institute of Technology (Sweden), The University of Newcastle (Australia), and Delft University of Technology. He is the recipient of several awards, including the 7th Grand Nagamori Award, the Corus Young Talent Graduation Award, and NWO Veni and Vidi personal grants. He currently serves as Senior Editor of IEEE Control Systems Letters (L-CSS) and Co-Editor-in-Chief of IFAC Mechatronics. His research spans all aspects from data to closed-loop control, and his application expertise includes precision mechatronics.

Sebastiaan P. Mulders received his degree in Systems and Control in 2015 and the Ph.D. degree in Wind Turbine Control in 2020, both from Delft University of Technology (TU Delft), The Netherlands. He continued at TU Delft as a postdoctoral researcher in 2021 and was appointed Assistant Professor in 2022. His research focuses on advanced control strategies for large-scale and flexible wind turbines. His work covers operational control, periodic load mitigation, and learning-based control strategies for flexible wind turbines. He has developed novel control approaches that explicitly account for blade and tower (periodic) deformations. These efforts extend turbine lifetime and enable enhanced energy capture while satisfying structural load constraints. He also develops data-driven learning algorithms for the online calibration of internal turbine models.

Jan-Willem van Wingerden was born in Ridderkerk, The Netherlands, in 1980. He received the M.Sc. degree in mechanical engineering and the Ph.D. degree (both cum laude) in control engineering from the Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands, in 2004 and 2008, respectively. Since 2017 he is a full Professor with the Delft University of Technology. His current research interests include the development of data-driven controllers for wind turbines and wind farms.