Proceedings of Machine Learning Research vol vvv:1–20, 2024

Convex Approximations for a Bi-level Formulation of Data-Enabled Predictive Control

Xu Shang X3SHANG@UCSD.EDU
Yang Zheng ZHENGY@UCSD.EDU
Department of Electrical and Computer Engineering, University of California San Diego

Abstract

The Willems' fundamental lemma, which characterizes linear time invariant (LTI) systems using input and output trajectories, has found many successful applications. Combining this with receding horizon control leads to a popular Data-EnablEd Predictive Control (`DeePC`) scheme. `DeePC` is first established for LTI systems and has been extended and applied for practical systems beyond LTI settings. However, the relationship between different `DeePC` variants, involving regularization and dimension reduction, remains unclear. In this paper, we first introduce a new bi-level optimization formulation that combines a data pre-processing step as an inner problem (system identification) and predictive control as an outer problem (online control). We next discuss a series of convex approximations by relaxing some hard constraints in the bi-level optimization as suitable regularization terms, accounting for an implicit identification. These include some existing `DeePC` variants as well as two new variants, for which we establish their equivalence under appropriate settings. Notably, our analysis reveals a novel variant, called `DeePC-SVD-iter`, which has remarkable empirical performance of direct methods on systems beyond deterministic LTI settings.
Keywords: Data-driven Control, Bi-level optimization, Convex approximation

1. Introduction

There has been a surging interest in utilizing data-driven techniques to control systems with unknown dynamics (Pillonetto et al., 2014; Markovsky and Dörfler, 2021). Existing methods can be generally categorized into indirect and direct data-driven control techniques: indirect data-driven control approaches typically include sequential system identification (system ID) (Ljung, 1998; Chiuso and Pillonetto, 2019) and model-based control (Kouvaritakis and Cannon, 2016), while direct data-driven control methods bypass system ID and directly design control strategies from input and output measured data (Markovsky and Dörfler, 2021).

In particular, Data-EnablEd Predictive Control (`DeePC`) (Coulson et al., 2019; Markovsky and Dörfler, 2021) that combines behavioral theory with receding horizon control has received increasing attention. It utilizes Willem's fundamental lemma (Willems, 2007) to construct a data-driven representation of a dynamic system and incorporates it with receding horizon control. `DeePC` is first established for deterministic linear time-invariant (LTI) systems, and its equivalence with subspace predictive control (SPC) has been discussed in (Fiedler and Lucia, 2021). Berberich et al. (2020) further investigate conditions for its closed-loop stability. The `DeePC` approach has shown promising results for the control of practical systems beyond LTI settings (Wang et al., 2023; Elokda et al., 2021; Shang et al., 2023; Lian et al., 2023). For non-deterministic or nonlinear systems, suitable regularizations are necessary for `DeePC`; see Breschi et al. (2023); Dörfler et al. (2022).

There are different regularization strategies for `DeePC`, ranging from some heuristics in Coulson et al. (2019) to principled analysis via bi-level formulations in Dörfler et al. (2022). Notably, indirect data-driven control is first formulated as a bi-level optimization problem involving both control and identification in Dörfler et al. (2022). Many regularized versions (such as l_1 or l_2 norms)

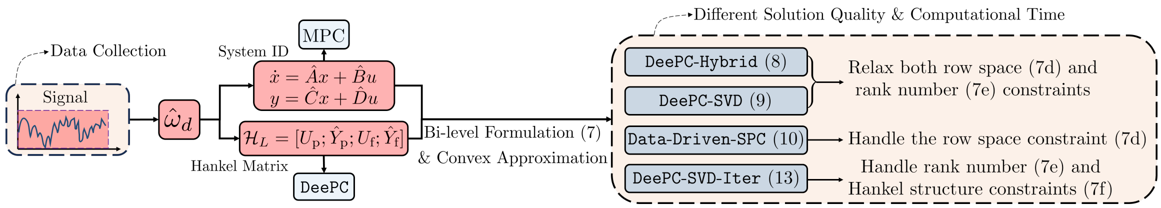


Figure 1: Schematic of data-driven control, which starts by collecting data (usually noisy) from the real system. In-direct methods identify a parametric model, while DeePC forms a Hankel matrix as the trajectory library for predictive control. Our bi-level formulation (7) integrates system ID techniques to DeePC. We introduce a series of convex approximations (8), (9), (10), and (13) that relax the bi-level formulation.

of DeePC can be considered as convex relaxations of the bi-level optimization. Beyond regularization, some recent approaches aim to decrease the optimization dimensions in DeePC and improve computational efficiency (Zhang et al., 2023; Alsalti et al., 2023). One simple strategy is to use a singular value decomposition (SVD) to pre-process the data-driven representation, which has shown promising performance (Zhang et al., 2023). However, the relationship between the recent variants of DeePC for non-deterministic and nonlinear systems, involving regularization and dimension reduction, remains unclear, and there is no analysis and comparison for their solution qualities.

In this paper, we introduce a new bi-level formulation incorporating both system ID techniques and predictive control, and discuss how existing and new variants of DeePC can be considered as convex approximations of this bi-level formulation; Figure 1 illustrates the overall process. Specifically, in our bi-level formulation for DeePC, the data pre-processing step is viewed as an inner optimization problem (identification), and the predictive control is viewed as an outer optimization problem (online control). Constraints for the inner optimization problem are derived from system ID methods (e.g., SPC Favoreel et al. (1999), and low-rank approximation Markovsky (2016)). We further discuss a series of convex approximations by relaxing some hard constraints as suitable regularization terms. In this process, we derive two new variants of DeePC by adapting existing methods: 1) Data-Driven-SPC is derived from classical SPC with the same structure as DeePC, and 2) DeePC-SVD-Iter refines the data-driven representation in DeePC-SVD from Zhang et al. (2023) and provide superior performance. We also investigate the equivalence of DeePC-Hybrid (Dörfler et al., 2022), DeePC-SVD (Zhang et al., 2023), and Data-Driven-SPC. Our analysis is more general than Zhang et al. (2023); Fiedler and Lucia (2021); Breschi et al. (2023). Numerical experiments confirm our analysis and show the superior performance of DeePC-SVD-Iter.

The rest of this paper is structured as follows. Section 2 introduces preliminaries and the problem statement of our bi-level formulation. A series of convex approximations are introduced in Section 3, their relationship is established in Section 4. Section 5 compares their control performance via numerical simulations. Finally, we conclude the paper in Section 6. Some notations and background are listed in Appendix A.1. We use $\mathrm{col}(A_1, A_2, \ldots, A_m) = [A_1^\mathsf{T}, A_2^\mathsf{T}, \ldots, A_m^\mathsf{T}]^\mathsf{T}$.

2. Preliminaries and Problem Statement

2.1. Preliminaries

We consider a linear time-invariant (LTI) system in the discrete-time domain:

$$\begin{cases} x(k+1) = Ax(k) + Bu(k), \\ \quad\;\; y(k) = Cx(k) + Du(k), \end{cases} \tag{1}$$

where the state, input, output at time k are $x(k) \in \mathbb{R}^n$, $u(k) \in \mathbb{R}^m$, and $y(k) \in \mathbb{R}^p$, respectively. Given a desired reference trajectory $y_{\mathrm{r}} \in \mathbb{R}^{pN}$ with horizon $N > 0$, input constraint set $\mathcal{U} \subseteq \mathbb{R}^m$, output constraint set $\mathcal{Y} \subseteq \mathbb{R}^p$, we aim to design control inputs such that the system output tracks the reference trajectory. In particular, we consider the well-known receding horizon predictive control

$$
\begin{aligned}
\min_{x,u,y} \quad & \sum_{k=t}^{t+N-1} \left(\|y(k) - y_{\mathrm{r}}(k)\|_Q^2 + \|u(k)\|_R^2\right) \\
\text{subject to} \quad & x(k+1) = Ax(k) + Bu(k), \quad k \in [t, t+N-1] && \text{(2a)} \\
& y(k) = Cx(k) + Du(k), \quad k \in [t, t+N-1] && \text{(2b)} \\
& x(t) = x_{\mathrm{ini}}, && \text{(2c)} \\
& u(k) \in \mathcal{U}, \ y(k) \in \mathcal{Y}, \quad k \in [t, t+N-1], && \text{(2d)}
\end{aligned}
$$

where $x_{\mathrm{ini}} \in \mathbb{R}^n$ is the initial state of system (1), and $\|u(k)\|_R^2$ denotes the quadratic norm $u(k)^{\mathsf{T}} Ru(k)$ (similarly for $\|\cdot\|_Q$) with $R \in \mathbb{S}_+^m$ and $Q \in \mathbb{S}_+^p$. We assume $\mathcal{U}$ and $\mathcal{Y}$ are convex sets. Without loss of generality, we consider a regulation problem (i.e., $y_{\mathrm{r}} = \mathbb{0}_{pN}$) from the rest of the discussions.

It is clear that (2) is a convex optimization problem (it is indeed a quadratic program for simple $\mathcal{U}$ and $\mathcal{Y}$), which admits an efficient solution when the model for the system (1) is known, i.e., matrices A, B, C and D are known. In this work, we focus on the case when the system model and the initial condition x_{ini} are unknown. Instead, we have access to 1) offline data, i.e., a length-T pre-collected input/output trajectory of (1), and 2) online data, i.e., the most recent past input/output sequence of length-T_{ini}. Then, (2) can be solved by either indirect system ID and model-based control (Åström and Eykhoff, 1971) or the recent emerging direct data-driven control, such as `DeePC` and its related approaches (Dörfler et al., 2022; Markovsky and Dörfler, 2021). As discussed in Dörfler et al. (2022), the indirect system ID approach is superior in the case of "variance" noise, while `DeePC` with suitable regularization terms has better performance in the case of "bias" errors.

2.2. Data-Enabled Predictive Control

We here review the basic setup of `DeePC`. First, let us introduce a notion of persistent excitation.

Definition 1 (Persistently Exciting) *A sequence of signal $\omega = \mathrm{col}(\omega(1), \omega(2), \ldots, \omega(T))$ of length T ($T \in \mathbb{N}$) is persistently exciting of order L ($L < T$) if its associated Hankel matrix with depth L, defined below, has full row rank,*

$$
\mathcal{H}_L(\omega) = \begin{bmatrix} \omega(1) & \omega(2) & \cdots & \omega(T-L+1) \\ \omega(2) & \omega(3) & \cdots & \omega(T-L+2) \\ \vdots & \vdots & \ddots & \vdots \\ \omega(L) & \omega(L+1) & \cdots & \omega(T) \end{bmatrix}.
$$

Lemma 1 (Fundamental Lemma; Willems et al. (2005)) *Suppose that system (1) is controllable. Given a length T input/output trajectory: $u_{\mathrm{d}} = \mathrm{col}(u_{\mathrm{d}}(1), \ldots, u_{\mathrm{d}}(T)) \in \mathbb{R}^{mT}$, $y_{\mathrm{d}} = \mathrm{col}(y_{\mathrm{d}}(1), \ldots, y_{\mathrm{d}}(T)) \in \mathbb{R}^{pT}$ where u_{d} is persistently exciting of order $L + n$, then a length L input/output sequence $(u_{\mathrm{s}}, y_{\mathrm{s}})$ is a valid trajectory of (1) if and only if there exists a $g \in \mathbb{R}^{T-L+1}$ such that*

$$
\begin{bmatrix} \mathcal{H}_L(u_{\mathrm{d}}) \\ \mathcal{H}_L(y_{\mathrm{d}}) \end{bmatrix} g = \begin{bmatrix} u_{\mathrm{s}} \\ y_{\mathrm{s}} \end{bmatrix}. \tag{3}
$$

If length L is not smaller than the lag of the system, matrix $\mathrm{col}(\mathcal{H}_L(u_{\mathrm{d}}), \mathcal{H}_L(y_{\mathrm{d}}))$ has rank $mL + n$.

The `DeePC` approach in Coulson et al. (2019) employs (3) to build a predictor based on the pre-collected data. In particular, the Hankel matrix formed by the offline data is partitioned as

$$\begin{bmatrix} U_\mathrm{P} \\ U_\mathrm{F} \end{bmatrix} := \mathcal{H}_L(u_\mathrm{d}), \quad \begin{bmatrix} Y_\mathrm{P} \\ Y_\mathrm{F} \end{bmatrix} := \mathcal{H}_L(y_\mathrm{d}), \tag{4}$$

where U_P and U_F consist the first T_ini rows and the last N rows of $\mathcal{H}_L(u_\mathrm{d})$, respectively (similarly for Y_P and Y_F; so $L = T_\mathrm{ini} + N$). We denote the most recent past input trajectory of length T_ini and the future input trajectory of length N as $u_\mathrm{ini} = \mathrm{col}(u(t - T_\mathrm{ini}), u(t - T_\mathrm{ini} + 1), \ldots, u(t - 1))$ and $u = \mathrm{col}(u(t), u(t + 1), \ldots, u(t + N - 1))$, respectively (similarly for y_ini, y).

Then, Lemma 1 ensures that the sequence $\mathrm{col}(u_\mathrm{ini}, y_\mathrm{ini}, u, y)$ is a valid trajectory of (1) if and only if there exists $g \in \mathbb{R}^{T-T_\mathrm{ini}-N+1}$ such that (5) holds. For notational simplicity, we further denote the matrix $\mathrm{col}(U_\mathrm{P}, Y_\mathrm{P}, U_\mathrm{F}, Y_\mathrm{F})$ associated with pre-collected data as H.

$$\begin{bmatrix} U_\mathrm{P} \\ Y_\mathrm{P} \\ U_\mathrm{F} \\ Y_\mathrm{F} \end{bmatrix} g = \begin{bmatrix} u_\mathrm{ini} \\ y_\mathrm{ini} \\ u \\ y \end{bmatrix}. \tag{5}$$

Note that H can be considered as a trajectory library since each of its columns is a valid trajectory of system (1). If T_ini is larger or equal to the lag of system (1), y is unique given any u_ini, y_ini and u in (5). The most basic version of `DeePC` (Coulson et al., 2019) utilizes the predictor (5) as the data-driven representation of (2a) to (2c) and reformulate the problem (2) as

$$\begin{aligned} \min_{g,u,y} \quad & \sum_{k=t}^{t+N-1} \left(\|y(k) - y_r(k)\|_Q^2 + \|u(k)\|_R^2\right) \\ \text{subject to} \quad & (5), u \in \mathcal{U}, y \in \mathcal{Y} \end{aligned} \tag{6}$$

where we slightly abuse the notation and use $u \in \mathcal{U}, y \in \mathcal{Y}$ to denote input/output constraints (2d).

2.3. A bi-level formulation beyond deterministic LTI systems

It is not difficult to show that for LTI systems with noise-free data, problems (2) and (6) are fully equivalent (cf. Coulson et al., 2019, Theorem 5.1). However, for the case beyond deterministic LTI systems, there exist different regularization terms or data pre-processing techniques that extends the basic `DeePC` (6). Indeed, an extensive discussion on bridging indirect and direct data-driven control was presented in Dörfler et al. (2022), where two different bi-level formulations were discussed.

Motivated by Dörfler et al. (2022), we propose a new bi-level formulation that incorporates data pre-processing techniques from system ID. In practice, the data predictor H in (5) may be corrupted by "variance" noises and/or "bias" errors. The key idea of our bi-level formulation is to pre-process the raw data H and construct a new trajectory library $\tilde{H}$ satisfying specific structures in system ID:

$$\underset{g,\sigma_y,u\in\mathcal{U},y\in\mathcal{Y}}{\text{minimize}} \quad \sum_{k=t}^{t+N-1} \left(\|y(k) - y_r(k)\|_Q^2 + \|u(k)\|_R^2\right) + \lambda_y \|\sigma_y\|_2^2 \tag{7a}$$

$$\text{subject to} \quad \tilde{H}^* g = \mathrm{col}(u_\mathrm{ini}, y_\mathrm{ini} + \sigma_y, u, y), \tag{7b}$$

$$\text{where} \quad \tilde{H}^* \in \arg\min_{\tilde{H}} \quad J(\tilde{H}, H), \tag{7c}$$

$$\text{subject to} \quad \tilde{Y}_\mathrm{F} = Y_\mathrm{F}/\mathrm{col}(\tilde{U}_\mathrm{P}, \tilde{Y}_\mathrm{P}, \tilde{U}_\mathrm{F}) \quad \text{(Row Space)}, \tag{7d}$$

$$\mathrm{rank}(\tilde{H}) = mL + n \quad \text{(Rank Number)}, \tag{7e}$$

$$\tilde{H} \in \mathcal{H} \quad \text{(Hankel Structure)}. \tag{7f}$$

This bi-level problem structure in (7), which is consistent with those in Dörfler et al. (2022), reflects the sequential ID and control tasks, where we first fit a model $\tilde{H}$ from the raw data H (4) in the inner system ID before using the model for `DeePC` in the outer problem.

In the outer problem (7a)-(7b), we have introduced a slack variable σ_y and its regularization term to handle the model mismatch and ensure feasibility, as discussed in Markovsky and Dörfler (2021). In the inner optimization problem (7c)-(7f), $J(\tilde{H}, H)$ in (7c) denotes system identification loss function with H being the raw Hankel matrix (4). We have also partitioned the variable $\tilde{H}$ as $\mathrm{col}(\tilde{U}_\mathrm{P}, \tilde{Y}_\mathrm{P}, \tilde{U}_\mathrm{F}, \tilde{Y}_\mathrm{F})$. In (7d), $Y_\mathrm{F}/\mathrm{col}(\tilde{U}_\mathrm{P}, \tilde{Y}_\mathrm{P}, \tilde{U}_\mathrm{F})$ denotes the orthogonal projection of Y_F onto the row space of $\mathrm{col}(\tilde{U}_\mathrm{P}, \tilde{Y}_\mathrm{P}, \tilde{U}_\mathrm{F})$. This row space constraint is derived from SPC (Favoreel et al., 1999) which will be discussed in detail in Section 3.2. The rank constraint (7e) and Hankel structure (7f) (where $\mathcal{H}$ is the set of all matrices with Hankel structure; cf. Definition 1) come from low-rank approximation in Markovsky (2016). We refer the interested readers to Fiedler and Lucia (2021) and Willems et al. (2005) for further details on row space and rank number respectively.

We will derive a series of convex approximations for this bi-level formulation (7) in Section 3 and discuss their equivalence (if possible) and relationship in Section 4.

3. Convex Approximations

While the bi-level formulation (7) is not solvable immediately, it provides useful guidance to derive new formulations/variants of `DeePC`. In this section, we present four convex approximations by adapting existing methods; see Figure 1 for an overview. These strategies relax the inner constraints (7d) to (7f) using suitable regularizers to the outer problem.

3.1. `DeePC` with regularization and dimension reduction

We first discuss two existing convex approximations of (7): `DeePC-Hybrid` from Dörfler et al. (2022) and `DeePC-SVD` from Zhang et al. (2023). Both of them use two different regularization terms to relax the rank constraint (7d) and the row space constraint (7e) while `DeePC-Hybrid` keeps the Hankel constraint (7f) and `DeePC-SVD` drops it.

Compared with the basic `DeePC` in (6), besides the regularizer $\|\sigma_y\|_2^2$, we introduce two extra regularizers $\|g\|_1$ and $\|(I-\Pi_1)g\|$ in `DeePC-Hybrid`, which reads as

$$
\begin{aligned}
\min_{g,\sigma_y,u\in\mathcal{U},y\in\mathcal{Y}} \quad & \|u\|_R^2 + \|y\|_Q^2 + \lambda_1\|g\|_1 + \lambda_2\|(I-\Pi_1)g\|_2^2 + \lambda_y\|\sigma_y\|_2^2 \\
\text{subject to} \quad & \begin{bmatrix} U_\mathrm{P} \\ Y_\mathrm{P} \\ U_\mathrm{F} \\ Y_\mathrm{F} \end{bmatrix} g = \begin{bmatrix} u_\mathrm{ini} \\ y_\mathrm{ini} + \sigma_y \\ u \\ y \end{bmatrix}
\end{aligned} \tag{8}
$$

where $\Pi_1 = H_1^\dagger H_1$ with $H_1 = \mathrm{col}(U_\mathrm{P}, Y_\mathrm{P}, U_\mathrm{F})$. Throughout the rest of the discussion, we denote $\|u\|_R^2 = \sum_{k=t}^{t+N-1}\|u(k)\|_R^2$ (similarly for $\|y\|_Q^2$). We note that the l_1 regularization $\|g\|_1$ can be viewed as a convex relaxation of the rank constraint (7e) (Dörfler et al., 2022, Theo. IV.8), while the regularization $\|(I-\Pi_1)g\|_2^2$ relaxes row space constraint (7d) (Dörfler et al., 2022, Theo. IV.6).

Since the column number of H is usually larger than its row number in practice (i.e., H is typically a fat matrix), `DeePC-SVD` in Zhang et al. (2023) utilizes singular value decomposition (SVD) to pre-process H and reduce its column dimension. Denoting the SVD of H as $H = W\Sigma V^\mathsf{T}$, where Σ contains its non-zero singular values, we construct a new data matrix $\bar{H} = W\Sigma$, and partition its rows as $\bar{H} = \mathrm{col}(\bar{U}_\mathrm{P}, \bar{Y}_\mathrm{P}, \bar{U}_\mathrm{F}, \bar{Y}_\mathrm{F})$. Then, the formulation `DeePC-SVD` reads as

$$\begin{aligned} \min_{\bar{g},\sigma_y,u\in\mathcal{U},y\in\mathcal{Y}} \quad & \|u\|_R^2 + \|y\|_Q^2 + \lambda_1\|\bar{g}\|_1 + \lambda_2\|(I-\bar{\Pi}_1)\bar{g}\|_2^2 + \lambda_y\|\sigma_y\|_2^2 \\ \text{subject to} \quad & \begin{bmatrix} \bar{U}_\mathrm{P} \\ \bar{Y}_\mathrm{P} \\ \bar{U}_\mathrm{F} \\ \bar{Y}_\mathrm{F} \end{bmatrix} \bar{g} = \begin{bmatrix} u_\mathrm{ini} \\ y_\mathrm{ini}+\sigma_y \\ u \\ y \end{bmatrix} \end{aligned} \tag{9}$$

where $\bar{\Pi}_1 = \bar{H}_1^\dagger \bar{H}_1$ and $\bar{H}_1 = \mathrm{col}(\bar{U}_\mathrm{P}, \bar{Y}_\mathrm{P}, \bar{U}_\mathrm{F})$. The dimension of $\bar{g}$ in (9) can be much smaller than that in (8), and this simple fact can improve numerical efficiency.

3.2. Data-Driven Subspace Predictive Control (SPC)

We here introduce a new `Data-Driven-SPC` to approximate (7) and establish its equivalence with the classical SPC in Favoreel et al. (1999). Similar to `DeePC-Hybrid` (8), we drop the Hankel structure constraint (7f) and use a l_1 regularization to relax the rank constraint (7e). However, we will directly handle the row space constraint (7d) without using any relaxation. Let us consider

$$\begin{aligned} \min_{\tilde{H}} \quad & \|\mathrm{col}(\tilde{U}_\mathrm{P}, \tilde{Y}_\mathrm{P}, \tilde{U}_\mathrm{F}) - \mathrm{col}(U_\mathrm{P}, Y_\mathrm{P}, U_\mathrm{F})\| \\ \text{subject to} \quad & \tilde{Y}_\mathrm{F} = Y_\mathrm{F}/\mathrm{col}(\tilde{U}_\mathrm{P}, \tilde{Y}_\mathrm{P}, \tilde{U}_\mathrm{F}). \end{aligned}$$

This inner problem has an analytical solution as $\tilde{H}^* = \mathrm{col}(U_\mathrm{P}, Y_\mathrm{P}, U_\mathrm{F}, M)$, with $M = Y_\mathrm{F}\Pi_1$ and Π_1 defined in Section 3.1. Then, we formulate `Data-Driven-SPC` as a problem in the form of (10). For self-completeness, we also present the classical SPC which is in the form of (11).

$$\begin{aligned} \min_{\sigma_y,g,u\in\mathcal{U},y\in\mathcal{Y}} \quad & \|u\|_R^2 + \|y\|_Q^2 + \lambda_1\|g\|_1 + \lambda_y\|\sigma_y\|_2^2 \\ \text{subject to} \quad & \begin{bmatrix} U_\mathrm{P} \\ Y_\mathrm{P} \\ U_\mathrm{F} \\ M \end{bmatrix} g = \begin{bmatrix} u_\mathrm{ini} \\ y_\mathrm{ini}+\sigma_y \\ u \\ y \end{bmatrix}. \end{aligned} \tag{10}$$

$$\begin{aligned} \min_{\sigma_y,u\in\mathcal{U},y\in\mathcal{Y}} \quad & \|u\|_R^2 + \|y\|_Q^2 + \lambda_y\|\sigma_y\|_2^2 \\ \text{subject to} \quad & y = Y_\mathrm{F} \begin{bmatrix} U_\mathrm{P} \\ Y_p \\ U_\mathrm{F} \end{bmatrix}^\dagger \begin{bmatrix} u_\mathrm{ini} \\ y_\mathrm{ini}+\sigma_y \\ u \end{bmatrix}. \end{aligned} \tag{11}$$

We show that (10) is indeed a direct data-driven version of (11) in the sense that they produce the same solution under a very mild condition. The proof is postponed to Appendix A.2.

Theorem 1 *If $Q \succ 0, R \succ 0, \lambda_1 = 0$ and $H_1 = \mathrm{col}(U_\mathrm{P}, Y_\mathrm{P}, U_\mathrm{F})$ has full row rank, then (10) and (11) have the same optimal solution u^*, y^*, σ_y^*, $\forall \lambda_y > 0$.*

Note that our new `Data-Driven-SPC` (10) is more flexible than the classical SPC (11) thanks to the parameter λ_1, which was motivated from the relaxation of the rank constraint (7e).

3.3. `DeePC` with dominant range space and Hankel structure

All the convex approximations (8), (9) and (10) use different regularizations to relax the difficult constraints (7d) and (7e), but they all directly drop the Hankel constraint (7f). In this subsection, we derive another new convex approximation for (7) that also approximates the Hankel structure with a dominate range space from the SVD. We call it as `DeePC-SVD-Iter`.

In particular, we consider (12) as the inner problem in the bi-level formulation (7), where the row space constraint (7d) will be relaxed using regularization. Note that (12) is also difficult to solve due to the interplay between (12b) and (12c). There are extensive results in the field of structured low-rank approximation (SLRA) (Markovsky, 2008). One idea is to use an alternative optimization strategy by considering (12b) and (12c) sequentially. Specifically, we here adapt an iterative SLRA algorithm in Yin and Smith (2021) to get an approximation solution to (12). We first note that problem (12) without (12c) admits an analytical solution $\tilde{H}^* = W_{\mathrm{r}}\Sigma_{\mathrm{r}}V_{\mathrm{r}}^{\mathsf{T}}$ where W_{r}, Σ_{r} and V_{r} represent the leading $mL+n$ singular vectors and singular values of H, i.e., the dominant range space.

$$
\begin{aligned}
\min_{\tilde{H}} \quad & \|\tilde{H} - H\|_2 && \text{(12a)}\\
\text{subject to} \quad & \operatorname{rank}(\tilde{H}) = mL + n && \text{(12b)}\\
& \tilde{H} \in \mathcal{H} && \text{(12c)}
\end{aligned}
$$

The key idea of the iterative SLRA is to utilize SVD for low-rank approximation of noisy data and then project the low-rank matrix to the set of Hankel matrices. This process is summarized in Algorithm 1. For notational simplicity, We define $H_u = \mathcal{H}_L(u_{\mathrm{d}})$, $H_y = \mathcal{H}_L(y_{\mathrm{d}})$ to denote the Hankel matrices in (4). Thanks to the persistent excitation on u_{d} with no noise, we have $\operatorname{rank}(H_u) = mL$. However, the measurement y_{d} usually contain "variance" noise and "bias" error, thus the data matrix satisfies $\operatorname{rank}(\operatorname{col}(H_u, H_y)) > mL + n$. We use an iterative procedure to denoise H_y while maintaining its Hankel structure. Let $\Pi_2 = H_u^{\dagger}H_u$ be the orthogonal projector onto the row space of H_u, and we first compute $H_y(I - \Pi_2)$ which is the component of H_y in the null space of H_u. In each iteration of Algorithm 1, we perform an SVD of $H_y(I - \Pi_2)$, estimate its rank-n approximation and combine it with the component of H_y in the row space of H_u as follows

Algorithm 1: Iterative SLRA

Input: H_y, Π_2, n, ϵ

$H_{y_1} \leftarrow H_y$ **repeat**

  $H_{y_2} \leftarrow \hat{H}(H_{y_1})$

  $H_{y_1} = \Pi_{\mathrm{H}}(H_{y_2})$

until $\|H_{y_1} - H_{y_2}\| \leq \epsilon \|H_{y_1}\|$;

Output: $H_y^* = H_{y_2}$

$$
H_y(I - \Pi_2) = \sum_{i=1}^{pL} \sigma_i u_i v_i^{\mathsf{T}}, \quad \hat{H}(H_y) = H_y\Pi_2 + \sum_{i=1}^{n} \sigma_i u_i v_i^{\mathsf{T}}.
$$

We then project $\hat{H}(H_y)$ onto the set of Hankel matrices by averaging skew-diagonal elements and define Π_{H} as the corresponding operator. The resulting matrix H_y^* from Algorithm 1 is partitioned as $\operatorname{col}(Y_{\mathrm{P}}^*, Y_{\mathrm{F}}^*)$, and we form a new Hankel matrix $\tilde{H}^* = \operatorname{col}(U_{\mathrm{P}}, Y_{\mathrm{P}}^*, U_{\mathrm{F}}, Y_{\mathrm{F}}^*)$. Finally, we perform an SVD of $\tilde{H}^* = W\Sigma V^{\mathsf{T}}$ to reduce its column dimension, and set $\tilde{W}_r\tilde{\Sigma}_r = \operatorname{col}(\hat{U}_{\mathrm{P}}, \hat{Y}_{\mathrm{P}}, \hat{U}_{\mathrm{F}}, \hat{Y}_{\mathrm{F}})$ with $r = mL + n$. This new matrix $\operatorname{col}(\hat{U}_{\mathrm{P}}, \hat{Y}_{\mathrm{P}}, \hat{U}_{\mathrm{F}}, \hat{Y}_{\mathrm{F}})$ is used as the predictor in `DeePC` as

$$
\begin{aligned}
\min_{\hat{g}, \sigma_y, u \in \mathcal{U}, y \in \mathcal{Y}} \quad & \|u\|_R^2 + \|y\|_Q^2 + \lambda_2 \|(I - \hat{\Pi}_1)\hat{g}\|_2^2 + \lambda_y \|\sigma_y\|_2^2 \\
\text{subject to} \quad & \begin{bmatrix} \hat{U}_{\mathrm{P}} \\ \hat{Y}_{\mathrm{P}} \\ \hat{U}_{\mathrm{F}} \\ \hat{Y}_{\mathrm{F}} \end{bmatrix} \hat{g} = \begin{bmatrix} u_{\mathrm{ini}} \\ y_{\mathrm{ini}} + \sigma_y \\ u \\ y \end{bmatrix}
\end{aligned} \tag{13}
$$

where $\hat{\Pi}_1 = \hat{H}_1^{\dagger}\hat{H}_1$, $\hat{H}_1 = \operatorname{col}(\hat{U}_{\mathrm{P}}, \hat{Y}_{\mathrm{P}}, \hat{U}_{\mathrm{F}})$ and $\|(I - \hat{\Pi}_1)\hat{g}\|_2^2$ is the relaxation term derived from the row space constraint. We call this formulation (13) as `DeePC-SVD-Iter`.

4. Relationship among Different Convex Approximations

As motivated above, (8) to (10) and (13) are all tractable convex approximations for the bi-level formulation (7). They all begin with the same data matrix H and apply different relaxation strategies to deal with the identification constraints (7d) to (7f). We here further look into their relationships and establish certain equivalence.

First, it is not difficult to see that all of them are equivalent when the data matrix H comes from an LTI system with no noise. We summarize this simple fact below.

Fact 1 *Suppose that the data matrix H in (4) comes from a controllable LTI system (1) with no noise, and the input $\mathrm{u_d}$ is persistently exciting of order $L+n$. Let $\lambda_1 = 0, \lambda_2 = 0$ and $\sigma_y = 0$. Then, all the* `DeePC` *variants in (8), (9), (10), and (13) have the same unique optimal solution u^*, y^*.*

For a controllable system (1) with no noise, the data matrix H has already satisfied row space (7d), rank number (7e), and Hankel structure constraints (7f). Then the new matrix $\tilde{H}^*$ after preprocessing in `Data-Driven-SPC` (10) and `DeePC-SVD-Iter` (13) will remain the same as that in `DeePC-Hybrid` (8), and their range space are also equal to that of `DeePC-SVD` (9). Thus, all these formulations have the same feasible region and cost functions, and they are equivalent.

We next move away from noise-free LTI systems. The data matrix H may have 'variance" noise and/or "bias" errors; see Dörfler et al. (2022). In this case, we can still show that `DeePC-Hybrid` (8) and `DeePC-SVD` (9) produce the same optimal solution u^*, y^*, σ_y^*.

Theorem 2 *Fix any data matrix H, and suppose $\lambda_1 = 0$, $\mathcal{U}$ and $\mathcal{Y}$ are convex. Then,* `DeePC-Hybrid` *(8) and* `DeePC-SVD` *(9) have the same optimal solution u^*, y^*, σ_y^*, $\forall \lambda_2 > 0, \lambda_y > 0$.*

We establish Theorem 2 by expressing g and $\bar{g}$ in terms of u, y, σ_y. Then, using the SVD properties, we show that (8) and (9) become strictly convex optimization problems with the same objective function, decision variables, and feasible region. The details are presented in Appendix A.3. Note that Theorem 2 includes Zhang et al., 2023, Theorem 1 as a special case, where Zhang et al. (2023) requires $\mathcal{U}$ and $\mathcal{Y}$ are convex polytopes that allow simple KKT conditions in their proof.

Finally, `DeePC-Hybrid`, `DeePC-SVD` and `Data-Driven-SPC` are also equivalent under certain conditions. This is summarized in Theorem 3 below, whose proof is shown in Appendix A.4.

Theorem 3 *Fix any data matrix H, and suppose $\lambda_1 = 0, \lambda_y > 0$ and $\mathcal{U}$ and $\mathcal{Y}$ are convex sets. If λ_2 is sufficiently large,* `DeePC-Hybrid` *(8) and* `DeePC-SVD` *(9) have the same unique optimal solution u^*, y^* and σ_y^* as* `Data-Driven-SPC` *(10).*

The key idea in the proof is to transform the regularizer $\lambda_2\|I - \Pi_1\|_2^2$ in (8) as the constraint $\|I-\Pi_1\|_2 = 0$ when λ_2 is sufficiently large via penalty arguments. Then, (8) and (10) have the same objective function and decision variables. The proof is completed by further establishing that they have the same feasible region. From Theorems 1 to 3, we conclude that `Data-Driven-SPC` (10), `DeePC-Hybrid` (8), `DeePC-SVD` (9) are equivalent to classical SPC (11) with noisy data when $\lambda_1 = 0$, $\lambda_y > 0$, λ_2 is sufficiently large and H_1 has full row rank. This is more general than Fiedler and Lucia (2021); Breschi et al. (2023): the equivalence for `DeePC-Hybrid` with regularizer $\|g\|_2^2$ and classical SPC is discussed in Fiedler and Lucia (2021), while `DeePC-Hybrid` and an approach similar to `Data-Driven-SPC` are proved to be equivalent in Breschi et al. (2023).

Note that the new variant `DeePC-SVD-Iter` involves an iterative algorithm to pre-process the noisy data (Algorithm 1), and thus it is non-trivial to formally establish its relationship with respect to other variants. Yet, our numerical experiments in Section 5 show that `DeePC-SVD-Iter` often has superior performance among all these convex approximations for noisy data.

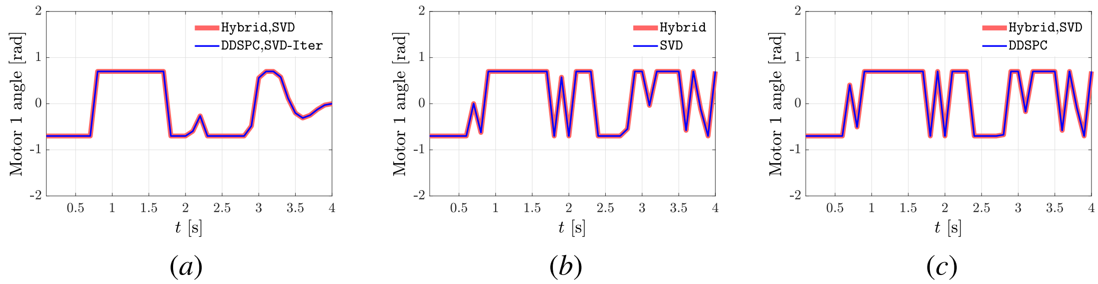


Figure 2: Equivalent optimal solutions of different convex approximations in Theorems 2 and 3. (a) Open-loop control inputs of all methods with noise-free data. (b) Open-loop control inputs of `DeePC-Hybrid` and `DeePC-SVD` with $\lambda_1 = 0, \lambda_2 = 30$ and $\lambda_y = 100$. (c) Open-loop control inputs of `DeePC-SVD-Hybrid`, `DeePC-SVD` and `Data-Driven-SPC` with $\lambda_1 = 0, \lambda_2 = 10000$ and $\lambda_y = 100$.

5. Numerical Experiments

We perform numerical experiments to illustrate Theorems 2 and 3[1]. We also numerically investigate the effects of λ_1, λ_2, and confirm the superior performance of `DeePC-SDV-iter` (13). Some additional numerical results on nonlinear systems are provided in Appendix B.

Experiment setup. We consider an LTI system from Fiedler and Lucia (2021). This is a triple-mass-spring system with $n = 8$ states, $m = 2$ inputs (two stepper motors), and $p = 3$ outputs (disc angles). In our experiments, the length of the pre-collected trajectory is $T = 200$, and the prediction horizon and the initial sequence are chosen as $N = 40$ and $T_{\text{ini}} = 4$, respectively. We choose $Q = I$, $R = 0.1I$ and $\mathcal{U} = [-0.7, 0.7]$.

Equivalence. We here numerically verify that the optimal solutions from different convex approximations are the same under appropriate settings. For noise-free pre-collected data (Fact 1), all methods have the same optimal solution, and one solution instance is given in Figure 2(*a*). We next consider data collection with additive Gaussian measurement noises $\omega \sim \mathcal{N}(0, 0.01I)$. We choose $\lambda_1 = 0, \lambda_2 = 30$ and $\lambda_y = 100$ according to Theorem 2. One solution instance is shown in Figure 2(*b*), which shows that `DeePC-Hybrid` and `DeePC-SVD` provide the same optimal solution. Finally, for the equivalence of `Data-Driven-SPC`, `DeePC-Hybrid` and `DeePC-SVD`, we choose $\lambda_1 = 0$ and $\lambda_2 = 10000$ according to Theorem 3, and the results are shown in Figure 2(*c*).

Influence of λ_1 and λ_2. We then analyze the effect of hyperparameters λ_1 and λ_2. In particular, similar to Dörfler et al. (2022), we consider the realized control cost after applying the optimal control inputs from (8) to (10) and (13), which is computed as $\|u_{\text{opt}}\|_R^2 + \|y_{\text{true}}\|_Q^2$, where u_{opt} is the computed optimal control input and y_{true} is the realized trajectory after applying it. We fixed $\lambda_y = 100$.

Figure 3 shows the realized control performance over λ_1 and λ_2 for different convex approximations. The hyperparameters λ_1 and λ_2 indeed have a significant effect for `DeePC-Hybrid` (8), `DeePC-SVD` (9) and `Data-Driven-SPC` (10)

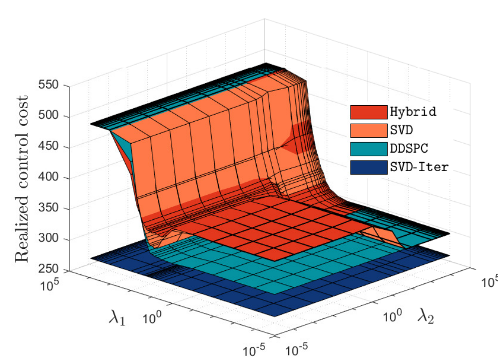
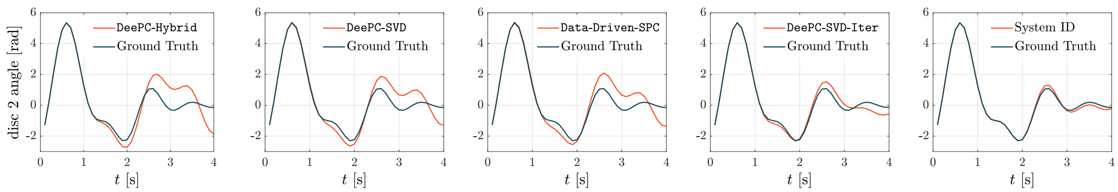


Figure 3: Realized control cost where sampling points are chosen from $\lambda_1, \lambda_2 \in [10^{-5}, 10^4]$ to capture the effect of hyperparameters.

1. Our code is available at `https://github.com/soc-ucsd/Convex-Approximation-for-DeePC`.

Table 1: Realized Control Cost and Computational Time; GT denotes ground truth with noisy-free data.

	GT	`Hybrid`	`SVD`	`Data-Driven SPC`	`SVD-Iter`	System ID
Realized Cost	277.25	388.42	370.20	365.08	288.17	279.86
Increasing Rate	N/A	40.1%	33.5%	31.7%	**3.9**%	0.9%
Compu. time [s]	N/A	0.133	0.131	0.097	0.104	N/A



(*a*) `Hybrid` (*b*) `SVD` (*c*) `DDSPC` (*d*) `SVD-Iter` (*e*) System ID

Figure 4: Open-loop trajectories (the angle of disc 2) of different methods. The blue trajectory and orange trajectories represent ground truth and different approximation methods, respectively.

(denoted as `DDSPC` in Figure 3). For these methods, λ_1 needs to be chosen more carefully, neither too large nor too small and λ_2 should not be chosen too small; similar phenomena also appeared in Dörfler et al. (2022). However, it is notable that `DeePC-SVD-Iter` (13) not only achieves the best performance but also is not very sensitive to λ_2 (note that $\lambda_1 = 0$ in (13)).

Comparison of `DeePC` variants. Finally, we compare the realized control cost and the computational time for different convex approximations. Motivated by Figure 3, we choose $\lambda_1 = \lambda_2 = 30$ and $\lambda_y = 100$. The performance of `DeePC` variants is related to the pre-collected trajectory. Thus, all presented realized control costs and computational time for different convex approximations are averaged over 100 pre-collected trajectories. The numerical results are listed in Table 1. The ground-truth cost is computed from (6) with noise-free data. From Table 1, we see that the realized control cost satisfies `DeePC-Hybrid` $>$ `DeePC-SVD` $>$ `Data-Driven-SPC` $>$ `DeePC-SVD-Iter` $>$ System ID. For the LTI system with noisy data, the inner problem in (7) forces the data-driven representation to be more structured, which enhances noise rejection performance for upper predictive control in (7). The increasing rate of realized cost for our new `DeePC-SVD-Iter` is 3.9%, which is much better than other `DeePC` variants.

Figure 4 shows one typical open-loop trajectory for all methods. In this case, the open-loop trajectories from (8) to (10) and (13) remain close to the ground truth up to 2s. Then the trajectory is better aligned with the ground truth from Fig. 4(*a*) to Fig. 4(*e*) as the corresponding data-driven representation becomes more structured. Our numerical results also suggest that the indirect system ID approach is superior in the case of "variance" noise, consistent with Dörfler et al. (2022). In Appendix B, our numerical results on nonlinear systems further reveal that `DeePC-SVD-Iter` (13) also has enhanced performance in the case of "bias" errors.

6. Conclusion

In this paper, we have proposed a new bi-level formulation incorporating system ID techniques and predictive control. The existing `DeePC` (i.e., `DeePC-Hybrid` (8) and `DeePC-SVD` (9)) and also new variants (i.e., `Data-driven SPC` (10) and `DeePC-SVD-iter` (13)) can be considered as convex approximations of this bi-level formulation. We have further clarified their equivalence under appropriated settings (Theorems 1 to 3). Numerical simulations have validated our theoretical findings, and also revealed the superior performance of `DeePC-SVD-iter` (13) with a more structured predictor. Interesting future directions include analyzing the effect of the length of pre-collected data, and investigating the closed-loop performance of different `DeePC` variants.

Acknowledgments

This work is supported in part by NSF ECCS-2154650 and in part by NSF CMMI-2320697.

References

Mohammad Alsalti, Ivan Markovsky, Victor G Lopez, and Matthias A Müller. Data-based system representations from irregularly measured data. *arXiv preprint arXiv:2307.11589*, 2023.

Karl Johan Åström and Peter Eykhoff. System identification—a survey. *Automatica*, 7(2):123–162, 1971.

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

Valentina Breschi, Alessandro Chiuso, and Simone Formentin. Data-driven predictive control in a stochastic setting: A unified framework. *Automatica*, 152:110961, 2023.

Alessandro Chiuso and Gianluigi Pillonetto. System identification: A machine learning perspective. *Annual Review of Control, Robotics, and Autonomous Systems*, 2:281–304, 2019.

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

Florian Dörfler, Jeremy Coulson, and Ivan Markovsky. Bridging direct and indirect data-driven control formulations via regularizations and relaxations. *IEEE Transactions on Automatic Control*, 68(2):883–897, 2022.

Ezzat Elokda, Jeremy Coulson, Paul N Beuchat, John Lygeros, and Florian Dörfler. Data-enabled predictive control for quadcopters. *International Journal of Robust and Nonlinear Control*, 31 (18):8916–8936, 2021.

Wouter Favoreel, Bart De Moor, and Michel Gevers. Spc: Subspace predictive control. *IFAC Proceedings Volumes*, 32(2):4004–4009, 1999.

Felix Fiedler and Sergio Lucia. On the relationship between data-enabled predictive control and subspace predictive control. In *2021 European Control Conference (ECC)*, pages 222–229. IEEE, 2021.

Gene H Golub and Charles F Van Loan. *Matrix computations*. JHU press, 2013.

Basil Kouvaritakis and Mark Cannon. Model predictive control. *Switzerland: Springer International Publishing*, 38, 2016.

Yingzhao Lian, Jicheng Shi, Manuel Koch, and Colin Neil Jones. Adaptive robust data-driven building control via bilevel reformulation: An experimental result. *IEEE Transactions on Control Systems Technology*, 2023.

Lennart Ljung. System identification. In *Signal analysis and prediction*, pages 163–173. Springer, 1998.

Ivan Markovsky. Structured low-rank approximation and its applications. *Automatica*, 44(4):891–909, 2008.

Ivan Markovsky. A missing data approach to data-driven filtering and control. *IEEE Transactions on Automatic Control*, 62(4):1972–1978, 2016.

Ivan Markovsky and Florian Dörfler. Behavioral systems theory in data-driven analysis, signal processing, and control. *Annual Reviews in Control*, 52:42–64, 2021.

Gianluigi Pillonetto, Francesco Dinuzzo, Tianshi Chen, Giuseppe De Nicolao, and Lennart Ljung. Kernel methods in system identification, machine learning and function estimation: A survey. *Automatica*, 50(3):657–682, 2014.

Xu Shang, Jiawei Wang, and Yang Zheng. Smoothing mixed traffic with robust data-driven predictive control for connected and autonomous vehicles. *arXiv preprint arXiv:2310.00509*, 2023.

Jiawei Wang, Yang Zheng, Keqiang Li, and Qing Xu. DeeP-LCC: Data-enabled predictive leading cruise control in mixed traffic flow. *IEEE Transactions on Control Systems Technology*, 31(6): 2760–2776, 2023.

Jan C. Willems. The behavioral approach to open and interconnected systems. *IEEE Control Systems Magazine*, 27(6):46–99, 2007. doi: 10.1109/MCS.2007.906923.

Jan C Willems, Paolo Rapisarda, Ivan Markovsky, and Bart LM De Moor. A note on persistency of excitation. *Systems & Control Letters*, 54(4):325–329, 2005.

Mingzhou Yin and Roy S Smith. On low-rank hankel matrix denoising. *IFAC-PapersOnLine*, 54 (7):198–203, 2021.

Kaixiang Zhang, Yang Zheng, Chao Shang, and Zhaojian Li. Dimension reduction for efficient data-enabled predictive control. *IEEE Control Systems Letters*, 7:3277–3282, 2023.

Appendix A. Technical proofs

In this section, we provide the technical proofs for Theorems 1 to 3 that are omitted in the main text.

A.1. Linear algebra fundamentals

We briefly review concepts of singular value decomposition (SVD), pseudo inverse and orthorgonal decomposition from linear algebra which are used in proofs later. We consider real matrices $A \in \mathbb{R}^{m_\text{a} \times n_\text{a}}$ and $B \in \mathbb{R}^{m_\text{b} \times n_\text{b}}$ since data collected from the real system are real numbers.

Singular value decomposition (SVD) is a factorization of a matrix which can be used to analyze the property of a matrix. Let $\text{rank}(A) = r$, the compact SVD of A is

$$A = W \Sigma V^\mathsf{T} \tag{14}$$

where $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_r) \in \mathbb{R}^{r \times r}$, $W = [w_1, \dots, w_r] \in \mathbb{R}^{m_\text{a} \times r}$, $V = [v_1, \dots, v_r] \in \mathbb{R}^{n_\text{a} \times r}$ and σ_i, w_i and v_i are singular values, left singular vectors and right singular vectors of A, respectively. Matrices Σ, W, V also satisfies $\sigma_1 \geq \dots \geq \sigma_r > 0$ and $W^\mathsf{T} W = I$, $V^\mathsf{T} V = I$ (i.e., w_i and v_i are othornormal vectors). We note that the range space and row space of A is the same as the range space of W and V, respectively.

The pseudo inverse of a matrix A, represented as $A^\dagger$, is a generalization of the inverse matrix. It is commonly used to solve a system of linear equations $Ax = b$. The pseudo inverse provides a least-squares approximate solution $x = A^\dagger b$ if there is no exact solution for the original problem. On the other hand, if the solution exists, $x_p = A^\dagger b$ is a least-norm solution and the general solution can be represented as $x = x_p + \hat{x}$ where $\hat{x}$ is in the null space of A. The pseudo inverse matrix $A^\dagger$ needs to satisfy the following four criteria

$$AA^\dagger A = A, \qquad A^\dagger A A^\dagger = A^\dagger, \qquad (AA^\dagger)^\mathsf{T} = AA^\dagger, \qquad (A^\dagger A)^\mathsf{T} = A^\dagger A,$$

which is unique and can be computed by compact SVD of A in (14) as $A^\dagger = V \Sigma^{-1} W^\mathsf{T}$. We list the following properties of pseudo inverse that we utilize in our proof:

1. If A has full row rank, $A^\dagger = A^\mathsf{T}(AA^\mathsf{T})^{-1}$ and $AA^\dagger = I$.
2. If A has full column rank, $A^\dagger = (A^\mathsf{T} A)^{-1} A^\mathsf{T}$ and $A^\dagger A = I$.
3. $(AA^\dagger)$ and $(A^\dagger A)$ are symmetric matrices.
4. The range space of $A^\dagger$ is the same as the row space of A.
5. If column vectors of A are orthonormal, $(A^\mathsf{T})^\dagger = A$.
6. If B has orthonormal rows, $(AB)^\dagger = B^\dagger A^\dagger$.

The orthogonal decomposition of a vector $u \in \mathbb{R}^n$ is the sum of a vector v in a subspace F of $\mathbb{R}^n$ and a vector $v^\perp$ in its orthogonal complement $F^\perp$ (i.e., $v^\mathsf{T} v^\perp = 0$) Golub and Van Loan (2013). The vector v is the orthogonal projection of u onto the subspace F. For the matrix $A = [a_1, \dots, a_{m_\text{a}}]^\mathsf{T}$, its row space orthogonal projection $M = [m_1, \dots, m_{m_\text{a}}]^\mathsf{T}$ onto the subspace F is constructed by the orthogonal projection m_i of vectors a_i such that $a_i = m_i + m_i^\perp, i = 1, \dots, m_\text{a}$ (the column space orthogonal projection is similar). The orthogonal projector onto the range space of A^T is

$A^\dagger A$ and the orthogonal decomposition of $u = v + v^\perp$ associated with range space of A^T can be computed as

$$v = A^\dagger A u, \quad v^\perp = (I - A^\dagger A)u,$$

and the row space orthogonal projection M of matrix B onto the row space of A is denoted as

$$M = B/A = ((A^\dagger A)B^\mathsf{T})^\mathsf{T} = B(A^\dagger A).$$

A.2. Proof of Theorem 1

Our proof is divided into two main parts:

1. When $H_1 = \text{col}(U_\text{P}, Y_\text{P}, U_\text{F})$ has full row rank, we show that (10) and (11) have the same feasible region: if g, u, y, σ_y is feasible to (10), then the same u, y, σ_y is also feasible for (11). Conversely, given any feasible solution u, y, σ_y to (11), we can construct a vector g such that g, u, y, σ_y is feasible to (10).

2. When $\lambda_1 = 0$, (10) and (11) have the same cost function in terms of u, y, σ_y.

Combining the two properties above with the fact that the cost function in (11) is strongly convex, we conclude that (10) and (11) have the same unique optimal solution u^*, y^*, σ_y^*.

The property 2 above is obvious. We prove the property 1 below. Let us first decompose

$$Y_\text{F} = M + M^\perp \tag{15}$$

where M is the orthogonal projection of Y_F on the row space of H_1 and $M^\perp$ is the remaining part of Y_F in the null space of H_1. Since H_1 has full row rank, we have $H_1 H_1^\dagger = I$. Also, the range space of $H^\dagger$ is the same as the row space of H, which means $M^\perp H_1^\dagger = 0$.

We assume $u_1, y_1, \sigma_{y_1}, g_1$ is a feasible solution for (10). Then, without loss of generality, the vector g_1 can be represented as

$$g_1 = H_1^\dagger \text{col}(u_\text{ini}, y_\text{ini} + \sigma_{y_1}, u_1) + \hat{g}$$

where $\hat{g}$ is in the null space of H_1. We have $M\hat{g} = 0$ and $Y_\text{F} H_1^\dagger = (M + M^\perp)H_1^\dagger = M H_1^\dagger$ because $M^\perp H_1^\dagger = 0$. Thus, from the equality constrain in (10), the vector y_1 satisfies

$$y_1 = M g_1 = M H_1^\dagger \begin{bmatrix} u_\text{ini} \\ y_\text{ini} + \sigma_{y_1} \\ u_1 \end{bmatrix} + M\hat{g} = M H_1^\dagger \begin{bmatrix} u_\text{ini} \\ y_\text{ini} + \sigma_{y_1} \\ u_1 \end{bmatrix} = Y_\text{F} H_1^\dagger \begin{bmatrix} u_\text{ini} \\ y_\text{ini} + \sigma_{y_1} \\ u_1 \end{bmatrix}, \tag{16}$$

which means u_1, y_1, σ_{y_1} is also a feasible solution of (11).

We next assume u_1, y_1, σ_{y_1} is a feasible solution for (11). Substituting the orthonormal decomposition (15) into the equality constraint of (11), we have

$$y_1 = Y_\text{F} H_1^\dagger \begin{bmatrix} u_\text{ini} \\ y_\text{ini} + \sigma_{y_1} \\ u_1 \end{bmatrix} = (M + M^\perp) H_1^\dagger \begin{bmatrix} u_\text{ini} \\ y_\text{ini} + \sigma_{y_1} \\ u_1 \end{bmatrix} = M H_1^\dagger \begin{bmatrix} u_\text{ini} \\ y_\text{ini} + \sigma_{y_1} \\ u_1 \end{bmatrix}. \tag{17}$$

Upon defining $g_1 = H_1^{\dagger}\text{col}(u_{\text{ini}}, y_{\text{ini}} + \sigma_{y_1}, u_1)$, we have $y_1 = Mg_1$ from (17). We then substitute g_1 into the equality constraint of (10), leading to

$$\begin{bmatrix} H_1 \\ M \end{bmatrix} g_1 = \begin{bmatrix} H_1 H_1^{\dagger} \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} + \sigma_{y_1} \\ u_1 \end{bmatrix} \\ Mg_1 \end{bmatrix} = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} + \sigma_{y_1} \\ u_1 \\ y_1 \end{bmatrix},$$

which means $g_1, u_1, y_1, \sigma_{y_1}$ is a feasible solution for (10). This completes our proof.

A.3. Proof of Theorem 2

Since $\lambda_1 = 0$, we only consider a two-norm $\|\cdot\|_2$ regularizer. Here, we consider `DeePC-Hybrid` (8) and `DeePC-SVD` (9) with a general two-norm regularization. For convenience, we rewrite their forms below:

$$\begin{aligned} \min_{g,\sigma_y,u\in\mathcal{U},y\in\mathcal{Y}} \quad & \|u\|_R^2 + \|y\|_Q^2 + \lambda_2\|Gg\|_2^2 + \lambda_y\|\sigma_y\|_2^2 \\ \text{subject to} \quad & \begin{bmatrix} U_{\text{P}} \\ Y_{\text{P}} \\ U_{\text{F}} \\ Y_{\text{F}} \end{bmatrix} g = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} + \sigma_y \\ u \\ y \end{bmatrix}, \end{aligned} \tag{18}$$

$$\begin{aligned} \min_{\bar{g},\sigma_y,u\in\mathcal{U},y\in\mathcal{Y}} \quad & \|u\|_R^2 + \|y\|_Q^2 + \lambda_2\|\bar{G}\bar{g}\|_2^2 + \lambda_y\|\sigma_y\|_2^2 \\ \text{subject to} \quad & \begin{bmatrix} \bar{U}_{\text{P}} \\ \bar{Y}_{\text{P}} \\ \bar{U}_{\text{F}} \\ \bar{Y}_{\text{F}} \end{bmatrix} \bar{g} = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} + \sigma_y \\ u \\ y \end{bmatrix}. \end{aligned} \tag{19}$$

We recall that the SVD decomposition:

$$\begin{aligned} H &= \text{col}(U_{\text{P}}, Y_{\text{P}}, U_{\text{F}}, Y_{\text{F}}) = W\Sigma V^{\mathsf{T}}, \\ \bar{H} &= \text{col}(\bar{U}_{\text{P}}, \bar{Y}_{\text{P}}, \bar{U}_{\text{F}}, \bar{Y}_{\text{F}}) = W\Sigma, \end{aligned}$$

where Σ contains r nonzero singular values, the columns of $W\Sigma$ are linearly independent and the columns of V are orthonormal vectors. The orthonormal columns lead to the property that $V^{\mathsf{T}}V = I$ and $(V^{\mathsf{T}})^{\dagger} = V$.

We next provide a set of sufficient conditions for the equivalence of (18) and (19). Its proof is provided in Appendix A.5.

Proposition 1 *If the matrices $G \in \mathbb{R}^{m_g \times T - T_{\text{ini}} - N+1}$, $\bar{G} \in \mathbb{R}^{m_{\bar{g}} \times T - T_{\text{ini}} - N+1}$, $V \in \mathbb{R}^{n\times r}$ and $H \in \mathbb{R}^{(m+p)(T_{\text{ini}}+N)\times T - T_{\text{ini}} - N+1}$ in (18) and (19) satisfy the following two properties:*

1. $\text{rowsp}(HG^{\mathsf{T}}G) \subseteq \text{rowsp}(H)$,
2. $V^{\mathsf{T}}G^{\mathsf{T}}GV = \bar{G}^{\mathsf{T}}\bar{G}$,

then the optimal solution for u, y, σ_y of (18) and (19) are the same and unique.

Now, Theorem 2 can be considered as a corollary of Proposition 1.

Proof of Theorem 2: It suffices to show that $G = I - \Pi_1$ and $\bar{G} = I - \bar{\Pi}_1$ satisfies conditions in Proposition 1.

For simplicity, let us denote $\mathrm{col}(u_{\text{ini}}, y_{\text{ini}} + \sigma_y, u, y)$ as v. We also define $\bar{H}_1 = \mathrm{col}(\bar{U}_{\text{P}}, \bar{Y}_{\text{P}}, \bar{U}_{\text{F}})$ and recall that $H_1 = \mathrm{col}(U_{\text{P}}, Y_{\text{P}}, U_{\text{F}})$. Then, the matrices G and $\bar{G}$ can be represented as

$$
\begin{aligned}
&G = I - H_1^\dagger H_1 = I - (\bar{H}_1 V^\top)^\dagger \bar{H}_1 V^\top = I - V\bar{H}_1^\dagger \bar{H}_1 V^\top, \\
&\bar{G} = I - \bar{H}_1^\dagger \bar{H}_1.
\end{aligned}
$$

We note that $(\bar{H}_1 V^\top)^\dagger = V\bar{H}_1^\dagger$ because $V^\top$ has orthonormal row vectors. For condition 1, we have

$$
HG^\top G = H(I - H_1^\dagger H_1)^\top (I - H_1^\dagger H_1) = H - HH_1^\top (H_1^\dagger)^\top - (H - HH_1^\top (H_1^\dagger)^\top) H_1^\dagger H_1
$$

and it is sufficient to prove row spaces of $(H_1^\dagger)^\top$ and H_1 are the subspace of the row space of H. Since H_1 is constructed by the top three matrices block of H and utilizing the property of the pseudo inverse, we have $\mathrm{rowsp}((H_1^\dagger)^\top) = \mathrm{rowsp}(H_1) \subseteq \mathrm{rowsp}(H)$. Thus, condition 1 is satisfied. For condition 2, we have

$$
\begin{aligned}
V^\top G^\top G V &= V^\top (I - V\bar{H}_1^\top (\bar{H}_1^\dagger)^\top V^\top)(I - V\bar{H}_1^\dagger \bar{H}_1 V^\top) V \\
&= (I - \bar{H}_1^\top (\bar{H}_1^\dagger)^\top) V^\top V (I - \bar{H}_1^\dagger \bar{H}_1 V) = \bar{G}^\top \bar{G},
\end{aligned}
$$

which means condition 2 is also satisfied.

A.4. Proof of Theorem 3

We present the proof of Theorem 3 for `DeePC-Hybrid` since `DeePC-Hybrid` and `DeePC-SVD` are equivalent when $\lambda_1 = 0$ (see Theorem 2). The key idea is we can derive `DeePC-Hybrid` as

$$
\begin{aligned}
\min_{g, \sigma_y, u \in \mathcal{U}, y \in \mathcal{Y}} \quad & \|u\|_R^2 + \|y\|_Q^2 + \lambda_y \|\sigma_y\|_2^2 \\
\text{subject to} \quad & \begin{bmatrix} U_{\text{P}} \\ Y_{\text{P}} \\ U_{\text{F}} \\ Y_{\text{F}} \end{bmatrix} g = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} + \sigma_y \\ u \\ y \end{bmatrix}, \qquad (20a) \\
& \|(I - \Pi_1) g\|_2 = 0, \qquad (20b)
\end{aligned}
$$

when λ_2 is sufficiently large. It is obvious that (20) and (10) have the same objective function when $\lambda_1 = 0$ and it only contains u, y and σ_y. Thus, we show they provide the same unique optimal solution u^*, y^* and σ_y^* by proving:

1. Feasible regions of u, y and σ_y are the same for (20) and (10): if g, u, y, σ_y is feasible to (20), then the same g, u, y, σ_y is also feasible for (10). Conversely, given any feasible solution g, u, y, σ_y to (10), we can construct a vector $\tilde{g}$ such that $\tilde{g}, u, y, \sigma_y$ is feasible to (20).

2. The optimial solution of u, y and σ_y is unique for (10).

We assume u_1, y_1, σ_{y_1} and g_1 is a feasible solution for (20). Substituting the orthogonal decomposition (15) of Y_F into (20a), we have

$$\begin{bmatrix} U_\mathrm{P} \\ Y_\mathrm{P} \\ U_\mathrm{F} \\ Y_\mathrm{F} \end{bmatrix} g_1 = \begin{bmatrix} U_\mathrm{P} \\ Y_\mathrm{P} \\ U_\mathrm{F} \\ M + Y_\mathrm{F}(I - \Pi_1) \end{bmatrix} g_1 = \begin{bmatrix} U_\mathrm{P} \\ Y_\mathrm{P} \\ U_\mathrm{F} \\ M \end{bmatrix} g_1 = \begin{bmatrix} u_\mathrm{ini} \\ y_\mathrm{ini} + \sigma_y \\ u \\ y \end{bmatrix}$$

by utilizing (20b) that $(I - \Pi_1)g_1 = 0$. Thus, u_1, y_1 and σ_{y_1} is also a feasible solution for (10).

We next assume u_1, y_1, σ_{y_1} and g_1 is a feasible solution for (10). We define $\tilde{g}_1 = H_1^\dagger \begin{bmatrix} u_\mathrm{ini} \\ y_\mathrm{ini} + \sigma_{y_1} \\ u_1 \end{bmatrix}$ and have $y_1 = Y_\mathrm{F}\tilde{g}_1$ from (16). We then substitute $\tilde{g}_1$ into(20a) and have

$$\begin{bmatrix} U_\mathrm{P} \\ Y_\mathrm{P} \\ U_\mathrm{F} \\ Y_\mathrm{F} \end{bmatrix} \tilde{g}_1 = \begin{bmatrix} H_1 \\ Y_\mathrm{F} \end{bmatrix} H_1^\dagger \begin{bmatrix} u_\mathrm{ini} \\ y_\mathrm{ini} + \sigma_{y_1} \\ u_1 \end{bmatrix} = \begin{bmatrix} u_\mathrm{ini} \\ y_\mathrm{ini} + \sigma_{y_1} \\ u_1 \\ y_1 \end{bmatrix}.$$

For the satisfaction of (20b), since $\tilde{g}_1$ is in the range space of $H_1^\dagger$, we have $\Pi_1\tilde{g}_1 = \tilde{g}_1$ which implies $\|(I - \Pi_1)\tilde{g}_1\|_2 = \|\tilde{g}_1 - \tilde{g}_1\|_2 = 0$. Thus, u_1, y_1, σ_{y_1} and $\tilde{g}_1$ is a feasible solution for (20).

We then prove the uniqueness of the optimal solution u^*, y^* and σ_y^* for (10). We define $x = \mathrm{col}(u, y, \sigma_y, g)$ and $f(x) = \|u\|_R^2 + \|y\|_Q^2 + \lambda_y\|\sigma_y\|_2^2$ as the decision variable and objective function of (10), respectively. We assume x_1 and x_2 are optimal solutions with different u, y and σ_y and the optimal value is f^*. We can then construct $x_3 = \alpha x_1 + (1 - \alpha)x_2$ where $0 < \alpha < 1$. Substituting x_3 into the constraint of (10), we have

$$\begin{bmatrix} U_\mathrm{P} \\ Y_\mathrm{P} \\ U_\mathrm{F} \\ M \end{bmatrix} g_3 = \begin{bmatrix} U_\mathrm{P} \\ Y_\mathrm{P} \\ U_\mathrm{F} \\ M \end{bmatrix} (\alpha g_1 + (1 - \alpha)g_2) = \begin{bmatrix} u_\mathrm{ini} \\ y_\mathrm{ini} + \alpha\sigma_{y_1} + (1 - \alpha)\sigma_{y_2} \\ \alpha u_1 + (1 - \alpha)u_2 \\ \alpha y_1 + (1 - \alpha)y_2 \end{bmatrix} = \begin{bmatrix} u_\mathrm{ini} \\ y_\mathrm{ini} + \sigma_{y_3} \\ u_3 \\ y_3 \end{bmatrix},$$

which means x_3 is a feasible solution. It is obvious that $f(x)$ is a strongly convex function with respect to u, y and σ_y and its value does not affected by g. Thus, by substituting x_3 into the objective function $f(x)$, we have

$$f(x_3) = f(\alpha x_1 + (1 - \alpha)x_2) < \alpha f(x_1) + (1 - \alpha)f(x_2) = f^*,$$

which conflicts with our assumption that f^* is the optimal value. Thus, the optimal solution for (10) is unique. This completes our proof.

A.5. Proof of Proposition 1

We prove Proposition 1 by showing

1. Given u, y, and σ_y, g and $\bar{g}$ have analytical optimal solutions that minimize the objective function when condition 1 in Proposition 1 holds.

2. The objective function of (18) and (19) are the same after replacing g and $\bar{g}$ with u, y and σ_y when condition 2 in Proposition 1 holds.

With the above two properties and the fact that (18) and (19) have the same strongly convex objection function, decision varibales and feasible region after replacing g, we conclude that they have the same unique optimal solution u^*, y^* and σ_y^*.

We first prove the property 1 and we recall that $\text{col}(u_{\text{ini}}, y_{\text{ini}}+\sigma_y, u, y)$ is denoted as v previously. Given u, y, σ_y, without loss of generality, we can represent g as

$$g = g_{\text{p}} + \hat{g}, \tag{21}$$

where $g_{\text{p}} = H^\dagger v$ and $\hat{g} = \text{null}(H)$. We then substitute (21) into the regularizer $\|Gg\|_2^2$ and have

$$\begin{aligned}\|Gg\|_2^2 = \|G(g_{\text{p}} + \hat{g})\|_2^2 &= \|Gg_{\text{p}}\|_2^2 + \|G\hat{g}\|_2^2 + 2(g_{\text{p}}^\top G^\top G\hat{g}) \\ &= \|Gg_{\text{p}}\|_2^2 + \|G\hat{g}\|_2^2 + 2(v^\top (H^\dagger)^\top G^\top G\hat{g}).\end{aligned}$$

Since $(H^\dagger)^\top$ has the same row space as H, using condition 1 that the $\text{rowsp}(HG^\top G)$ is the subspace of the $\text{rowsp}(H)$, we have $\text{rowsp}((H^\dagger)^\top G^\top G) \subseteq \text{rowsp}(H)$. Then we get $v^\top (H^\dagger)^\top G^\top G\hat{g} = 0$ because of $\hat{g} \in \text{null}(H)$. Thus, we can further derive $\|Gg\|_2^2 = \|Gg_{\text{p}}\|_2^2 + \|G\hat{g}\|_2^2$ which means, given u, y and σ_y, g_{p} is an analytical solution to minimize the objective function under condition 1. For (19), since $\bar{H}$ has linearly independent columns, $\bar{g}$ has a unique solution given u, y and σ_y which is $\bar{g}_{\text{p}} = \bar{H}^\dagger v$.

Upon getting the analytical optimal solution g_p and $\bar{g}_p$, we substitute them into the objective function f_1 and f_2 of (18) and (19), leading to

$$\begin{aligned}f_1(u, y, \sigma_y) &= \|u\|_R^2 + \|y\|_Q^2 + \lambda_g \|GH^\dagger v\|_2^2 + \lambda_y \|\sigma_y\|_2^2 \\ &= \|u\|_R^2 + \|y\|_Q^2 + \lambda_g \|GV(W\Sigma)^\dagger v\|_2^2 + \lambda_y \|\sigma_y\|_2^2, \\ f_2(u, y, \sigma_y) &= \|u\|_R^2 + \|y\|_Q^2 + \lambda_g \|\bar{G}\bar{H}^\dagger v\|_2^2 + \lambda_y \|\sigma_y\|_2^2 \\ &= \|u\|_R^2 + \|y\|_Q^2 + \lambda_g \|\bar{G}(W\Sigma)^\dagger v\|_2^2 + \lambda_y \|\sigma_y\|_2^2.\end{aligned}$$

When the condition 2 hold, we have

$$\begin{aligned}\|GV(W\Sigma)^\dagger v\|_2^2 &= v^\top ((W\Sigma)^\dagger)^\top V^\top G^\top GV(W\Sigma)^\dagger v \\ &= v^\top ((W\Sigma)^\dagger)^\top \bar{G}^\top \bar{G}(W\Sigma)^\dagger v = \|\bar{G}(W\Sigma)^\dagger v\|_2^2,\end{aligned}$$

which means (18) and (19) have the same objective function. This completes our proof.

Appendix B. Additional numerical results

In this section, we present the results for nonlinear systems that exhibit "bias" errors. We conduct a numerical comparison between the indirect data-driven method and various variants of `DeePC` across systems characterized by varying degrees of nonlinearity.

Experiment setup. We consider the nonlinear Lotka-Volterra dynamics used in Dörfler et al. (2022)

$$\dot{x} = \begin{bmatrix}\dot{x}_1 \\ \dot{x}_2\end{bmatrix} = \begin{bmatrix} ax_1 - bx_1x_2 \\ dx_1x_2 - cx_2 + u\end{bmatrix}, \tag{22}$$

where $a = c = 0.5, b = 0.025, d = 0.005, x_1, x_2$ denote prey and predator populations and u is the input. We first linearize the system (22) about the equilibrium $(\bar{u}, \bar{x}_1, \bar{x_2}) = (0, \frac{c}{d}, \frac{a}{b})$ which yields the linear system in the error state space and discretize it

$$\hat{x}(k+1) = f_{\text{linear}}(\hat{x}(k), \hat{u}(k)) = \begin{bmatrix} \hat{x}_1(k) + \Delta t(-b\bar{x}_1\hat{x}_2(k)) \\ \hat{x}_2(k) + \Delta t(d\bar{x}_2\hat{x}_1(k) + \hat{u}(k)) \end{bmatrix}$$

where $\Delta t = 0.1$ is the time step for discretization. We then discretize the nonlinear system in the error state space

$$\begin{aligned} \hat{x}(k+1) &= f_{\text{nonlinear}}(\hat{x}(k), \hat{u}(k)) \\ &= \begin{bmatrix} \hat{x}_1(k) + \Delta t(a(\hat{x}_1(k) + \bar{x_1}) - b(\hat{x}_1(k) + \bar{x_1})(\hat{x}_2(k) + \bar{x_2})) \\ \hat{x}_2(k) + \Delta t(d(\hat{x}_1(k) + \bar{x_1})(\hat{x}_2(k) + \bar{x_2}) - c(\hat{x}_2(k) + \bar{x_2}) + \hat{u}(k)) \end{bmatrix}. \end{aligned}$$

We construct systems with various nonlinearity by interpolating between f_{linear} and $f_{\text{nonlinear}}$ that is

$$\hat{x}(k+1) = \epsilon \cdot f_{\text{linear}}(\hat{x}(k), \hat{u}(k)) + (1-\epsilon) \cdot f_{\text{nonlinear}}(\hat{x}(k), \hat{u}(k)).$$

The length of the pre-collected trajectory is $T = 300$ and the prediction horizon, initial sequence are set to $N = 60$ and $T_{\text{ini}} = 4$, respectively. We choose $Q = I, R = 0.5I$ and $\mathcal{U} = [-20, 20]$. We further fix $\lambda_1 = 300, \lambda_2 = 100$ and $\lambda_y = 10000$ for all simulations.

Comparison of direct/indirect methods. We compare the realized control costs for system ID and different convex approximations on systems with varying degrees of nonlinearity. Similar to the previous section, the presented realized control costs are averaged over 100 pre-collected trajectories. As shown in Fig. 5, both direct (convex approximations) and indirect (system ID) approaches perform well when the nonlinearity is low ($\epsilon \in [0.9, 1]$). However, the cost for the indirect method significantly increases with higher nonlinearity, while the performance of direct methods remains relatively consistent. The superior performance of direct data-driven methods aligns with results from Dörfler et al. (2022).

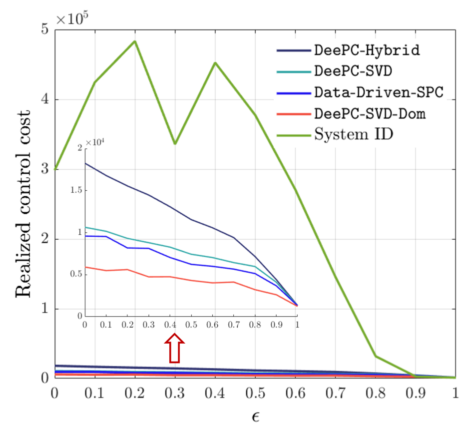


Figure 5: Comparison of realized cost for system ID and `DeePC` based Methods

Comparison of `DeePC` variants. We then compare the performance of different convex approximations as displayed in Fig. 6. The `DeePC-SVD-Iter` and `Data-Driven-SPC` with structured data-driven representation outperform `DeePC-Hybrid` and `DeePC-SVD` which lift constraints as regularizers in objective functions. Furthermore, `DeePC-SVD-Iter` degrades much less than other methods. This numerical analysis illustrates the additional benefit of employing techniques from system ID to pre-process the trajectory library of the nonlinear system.

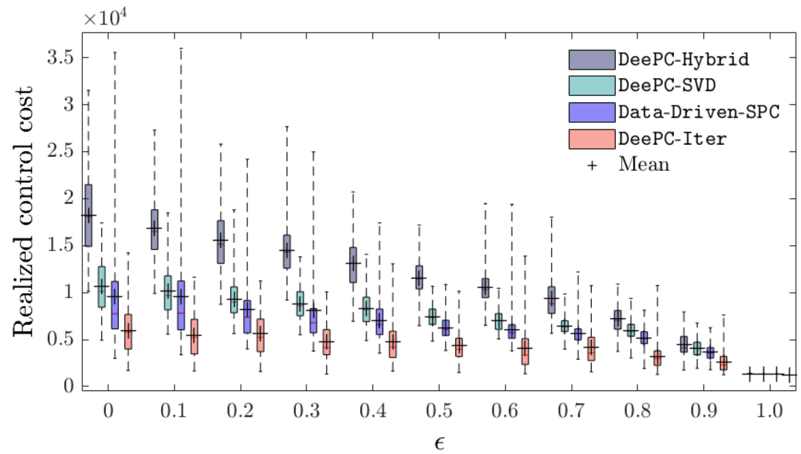


Figure 6: Comparison of realized control cost of convex approximations with varying nonlinearity. The navy block and cyan block represent `DeePC-Hybrid` and `DeePC-SVD`, respectively. The blue block and red block correspond to `Data-Driven-SPC` and `DeePC-SVD-Iter`.