A Modified Adaptive Data-Enabled Policy Optimization Control to Resolve State Perturbations

Mojtaba Kaheni, Niklas Persson, Vittorio De Iuliis, Costanzo Manes, and Alessandro V. Papadopoulos

***Abstract*— This paper proposes modifications to the data-enabled policy optimization (DeePO) algorithm to mitigate state perturbations. DeePO is an adaptive, data-driven approach designed to iteratively compute a feedback gain equivalent to the certainty-equivalence LQR gain. Like other data-driven approaches based on Willems' fundamental lemma, DeePO requires persistently exciting input signals. However, linear state-feedback gains from LQR designs cannot inherently produce such inputs. To address this, probing noise is conventionally added to the control signal to ensure persistent excitation. However, the added noise may induce undesirable state perturbations. We first identify two key issues that jeopardize the desired performance of DeePO when probing noise is not added: the convergence of states to the equilibrium point, and the convergence of the controller to its optimal value. To address these challenges without relying on probing noise, we propose Perturbation-Free DeePO (PFDeePO) built on two fundamental principles. First, the algorithm pauses the control gain updating in DeePO process when system states are near the equilibrium point. Second, it applies a multiplicative noise, scaled by a mean value of 1 as a gain for the control signal, when the controller converges. This approach minimizes the impact of noise as the system approaches equilibrium while preserving stability. We demonstrate the effectiveness of PFDeePO through simulations, showcasing its ability to eliminate state perturbations while maintaining system performance and stability.**

I. Introduction

Traditionally, the design of controllers has relied on two primary approaches: first-principle modeling which is based on established physical laws and principles specific to the system's domain, or system identification that uses available data to construct a mathematical representation. These mathematical models, are then employed to analyze the system's behavior under various inputs and to derive a control law that satisfies specified performance criteria.

However, the seminal work by Willems *et al.* [1] introduced a groundbreaking concept with far-reaching implications for control system design. This study revealed that a finite set of system trajectories, generated using persistently exciting inputs, is sufficient to describe the complete behavior of a controllable linear time-invariant (LTI) system. It implies that controllable LTI systems can be fully characterized using only finite historical data, eliminating the need for traditional model-based representations. This result has sparked widespread interest within the control systems community, as it presents a promising alternative for simplifying controller design. By potentially circumventing the expensive and time-consuming processes of system identification or first-principle modeling, this approach offers a new framework for streamlining and accelerating the development of control systems.

Linear quadratic regulators (LQR) are a widely adopted control design method due to their ability to balance the trade-off between convergence rate and control effort, which can be adjusted based on the designer's preferences. In LTI systems, when the mathematical state-space representation of the system is available, solving a straightforward optimization problem provides the state-feedback gain that minimizes the corresponding cost function [2]. In cases where, instead of the state-space representation, sufficiently rich measurements are available, it is still possible to design an LQR controller by conventional approaches, but an initial identification step is required to first find the system matrices. This approach is typically referred to in the literature as *indirect data-driven* LQR design [3]–[6].

In contrast, several methods have recently been proposed to design LQR controllers for LTI systems directly from data, without the need to identify the system matrices [7]–[14]. These approaches are commonly known as *direct data-driven* LQR design. Most of the direct data-driven LQR design methods rely on pre-measured datasets to compute the optimal feedback gain. A natural consideration is to allow the control system to leverage the data it measures during operation to improve or fine-tune the LQR performance. Toward this goal, the Data-enabled Policy Optimization (DeePO) algorithm was recently proposed [15], [16]. DeePO incorporates an adaptation feature on top of direct data-driven LQR design. At each time step, newly measured input and state data are added to the previously stored dataset, and the control feedback gain is updated iteratively using a learning rate, steering the design towards reducing the objective function's cost based on the updated data. Furthermore, it has been proven that DeePO computes a feedback gain equivalent to the certainty-equivalence LQR gain typically derived using indirect data-driven techniques [16]. DeePO has been evaluated in simulations on a power converter

This work was supported in part by the Swedish Research Council (VR) with grant "Pervasive Self-Optimizing Computing Infrastructures (PSI)" n. 2020-05094, and by the Knowledge Foundation (KKS) with grant "Mälardalen University Automation Research Center (MARC)", n. 20240011. The work was also supported in part by the Italian Government under CIPE resolution n. 70/2017 (Centre of excellence EX-EMERGE), and by University of L'Aquila (Progetti di Avvio alla Ricerca 2025).

M. Kaheni, N. Persson, and A.V. Papadopoulos are with the Division of Intelligent Future Technologies, Mälardalen University, 721 23 Västerås, Sweden. (e-mails: mojtaba.kaheni@mdu.se, niklas.persson@mdu.se, alessandro.papadopoulos@mdu.se).

V. De Iuliis and C. Manes are with the Department of Information Engineering, Computer Science and Mathematics, University of L'Aquila, Italy. (e-mails: vittorio.deiuliis@univaq.it, costanzo.manes@univaq.it).

system [17], and was used for balancing an autonomous bicycle in real experiments [18]. Similar to other data-driven methods based on Willems' fundamental lemma, DeePO requires persistently exciting input signals. On the other hand, linear state-feedback gains from LQR designs cannot inherently generate such inputs. To overcome this limitation, probing noise is conventionally added to the control signal to ensure persistent excitation. This added noise often induces undesirable state perturbations. In this article, we introduce PFDeePO to resolve state perturbations in DeePO.

A. Notation

Throughout this paper, unless clearly stated otherwise, the symbols $\mathbb{N}$, $\mathbb{Z}$, and $\mathbb{R}$ denote the sets of, natural, integer, and real numbers, respectively. Scalars are represented by lowercase letters such as x, while $\mathbf{x}$ and $\mathbf{X}$ denote a (column) vector and a matrix, respectively. The notation $\mathbf{X} \prec 0$ ($\mathbf{X} \preceq 0$) and $\mathbf{X} \succ 0$ ($\mathbf{X} \succeq 0$) indicates that $\mathbf{X}$ is negative (semi-) definite and positive (semi-) definite, respectively. Additionally, $\mathbf{I}_i$ denotes the $i \times i$ identity matrix. The symbols $\mathbf{0}_{i\times j}$ and $\mathbf{1}_{i\times j}$ represent all-zeros and all-ones matrices of size $i \times j$, respectively. The 2-norm of matrix $\mathbf{X}$ is denoted by $\|\mathbf{X}\|$. $\mathbf{X}^\top$, $\mathrm{Tr}(\mathbf{X})$, and $\mathbf{X}^\dagger$ represent the transpose, trace, and pseudoinverse of matrix $\mathbf{X}$, respectively. The notation $\mathbf{x}^i$ and $\mathbf{X}^i$ refer to the i-th coordinate of $\mathbf{x}$ and the i-th row of $\mathbf{X}$, respectively. Meanwhile, $\mathbf{X}^{i,j}$ denotes the element located in the i-th row and j-th column of $\mathbf{X}$. $\mathscr{N}(\mathbf{x},\mathbf{X})$ represents a multivariate Gaussian distribution with a mean vector $\mathbf{x}$ and a covariance matrix $\mathbf{X}$. $\Pi_{\mathbf{X}}$ stands for projection operator on $\mathbf{X}$. Finally, $\underline{\sigma}(\mathbf{X})$ denotes the smallest singular value of $\mathbf{X}$.

II. Data-Driven Policy Optimization for LQR Learning

A. Background

Consider an LTI discrete-time system, represented in state space form as:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \boldsymbol{\omega}_k, \\ \mathbf{z}_k &= \begin{bmatrix} \mathbf{Q}^{1/2} & \mathbf{0}_{n\times m} \\ \mathbf{0}_{m\times n} & \mathbf{R}^{1/2} \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix}, \end{aligned} \tag{1}$$

where $k \in \mathbb{N}$ is the index for counting samples, $\mathbf{x} \in \mathbb{R}^n$ is the state, $\mathbf{u} \in \mathbb{R}^m$ represents the input, and $\boldsymbol{\omega}_k$ is noise. Furthermore, let $\mathbf{z}_k \in \mathbb{R}^{n+m}$ represent the performance signal. We assume that the pair $(\mathbf{A},\mathbf{B})$ is controllable, and that $(\mathbf{Q},\mathbf{R})$ are positive definite square matrices with compatible dimensions. The objective of the LQR design is to determine a state feedback controller, $\mathbf{K} \in \mathbb{R}^{m\times n}$, that minimizes the $\mathscr{H}_2$-norm of the transfer function $\mathscr{T}(\mathbf{K}) : \boldsymbol{\omega} \mapsto \mathbf{z}$ of the following closed-loop system:

$$\begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{z}_k \end{bmatrix} = \left[\begin{array}{c|c} \mathbf{A}+\mathbf{B}\mathbf{K} & \mathbf{I}_n \\ \hline \begin{bmatrix} \mathbf{Q}^{1/2} \\ \mathbf{R}^{1/2}\mathbf{K} \end{bmatrix} & \mathbf{0}_{(m+n)\times n} \end{array}\right] \begin{bmatrix} \mathbf{x}_k \\ \boldsymbol{\omega}_k \end{bmatrix}. \tag{2}$$

As discussed in [19], the $\mathscr{H}_2$-norm of the transfer function $\mathscr{T}(\mathbf{K})$ obtained from (2) can be expressed as:

$$C(\mathbf{K}) \triangleq \|\mathscr{T}(\mathbf{K})\|^2 = \mathrm{Tr}\left(\left(\mathbf{Q}+\mathbf{K}^\top\mathbf{R}\mathbf{K}\right)\boldsymbol{\Sigma}_{\mathbf{K}}\right). \tag{3}$$

where $C(\mathbf{K})$ represents the cost function, and $\boldsymbol{\Sigma}_{\mathbf{K}}$ is typically referred to as the closed-loop state covariance matrix, which is the solution to the following Lyapunov equation:

$$\boldsymbol{\Sigma}_{\mathbf{K}} = \mathbf{I}_n + (\mathbf{A}+\mathbf{B}\mathbf{K})\,\boldsymbol{\Sigma}_{\mathbf{K}}\,(\mathbf{A}+\mathbf{B}\mathbf{K})^\top. \tag{4}$$

Therefore, the LQR design can be summarized as:

$$\begin{aligned} \min_{\boldsymbol{\Sigma}_{\mathbf{K}}\succeq 0,\mathbf{K}} \quad & C(\mathbf{K}) = \mathrm{Tr}\left(\left(\mathbf{Q}+\mathbf{K}^\top\mathbf{R}\mathbf{K}\right)\boldsymbol{\Sigma}_{\mathbf{K}}\right) \\ \text{subject to} \quad & \boldsymbol{\Sigma}_{\mathbf{K}} = \mathbf{I}_n + (\mathbf{A}+\mathbf{B}\mathbf{K})\,\boldsymbol{\Sigma}_{\mathbf{K}}\,(\mathbf{A}+\mathbf{B}\mathbf{K})^\top. \end{aligned} \tag{5}$$

To directly compute the optimal feedback gain matrix $\mathbf{K}$ from (5), the system matrices $(\mathbf{A},\mathbf{B})$ must be known. However, if $(\mathbf{A},\mathbf{B})$ are unknown, it may still be possible to incorporate an identification step to estimate them. Suppose signals of length t of states, inputs, noises, and successor states, which do not necessarily need to be consecutive. These signals are defined as follows:

$$\begin{aligned} \mathbf{X}_0 &\triangleq \begin{bmatrix} \mathbf{x}_0 & \mathbf{x}_1 & \cdots & \mathbf{x}_{t-1} \end{bmatrix}, \\ \mathbf{X}_1 &\triangleq \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_{t} \end{bmatrix}, \\ \mathbf{U}_0 &\triangleq \begin{bmatrix} \mathbf{u}_0 & \mathbf{u}_1 & \cdots & \mathbf{u}_{t-1} \end{bmatrix}, \\ \mathbf{W}_0 &\triangleq \begin{bmatrix} \boldsymbol{\omega}_0 & \boldsymbol{\omega}_1 & \cdots & \boldsymbol{\omega}_{t-1} \end{bmatrix}. \end{aligned} \tag{6}$$

The input signal $\mathbf{U}_0$ must be *sufficiently rich* to effectively represent the dynamical system described by (1). This property is commonly referred to as *persistently exciting*, and is formally defined as follows:

Definition 1 ([1]): A signal $\mathbf{U}_0$ is said to be persistently exciting of order l when

$$\mathscr{U}_0 = \begin{bmatrix} \mathbf{u}_0 & \mathbf{u}_1 & \cdots & \mathbf{u}_{t-l} \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_{t-l+1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{u}_{l-1} & \mathbf{u}_l & \cdots & \mathbf{u}_{t-1} \end{bmatrix}, \tag{7}$$

has full rank ml. ■

The following lemma is also useful for determining the persistent excitation of a system.

Lemma 1 ([1]): If the system (1) is controllable and $\mathbf{U}_0$ is persistently exciting of order $n+1$, then $\mathrm{rank}(\mathscr{D}) = n+m$, where

$$\mathscr{D} \triangleq \begin{bmatrix} \mathbf{U}_0 \\ \mathbf{X}_0 \end{bmatrix}. \tag{8}$$

■

If $\mathbf{U}_0$ is persistently exciting, the estimates of the system matrices $(\mathbf{A},\mathbf{B})$ can be obtained by solving the following optimization problem:

$$(\hat{\mathbf{A}},\hat{\mathbf{B}}) = \min_{\mathbf{A},\mathbf{B}} \quad \left\|\mathbf{X}_1 - \begin{bmatrix} \mathbf{B} & \mathbf{A} \end{bmatrix}\begin{bmatrix} \mathbf{U}_0{}^\top & \mathbf{X}_0{}^\top \end{bmatrix}^\top\right\|. \tag{9}$$

The LQR controller can then be designed by substituting $(\hat{\mathbf{A}},\hat{\mathbf{B}})$, obtained from (9), in place of the true system matrices $(\mathbf{A},\mathbf{B})$ in (5). This approach is commonly referred to as *certainty-equivalence* and is a typical strategy in *indirect data-driven* LQR design [4], [20]–[22].

Recently, several methods have been proposed in the literature to bypass the identification step in (9) by directly leveraging the data introduced in (6). These methods are

commonly referred to as *direct data-driven* LQR design [7]–[13]. From Lemma 1, we know that if $\mathbf{U}_0$ is persistently exciting of order $n+1$, then $\text{rank}(\mathscr{D}) = n+m$. Consequently, by the Rouché–Capelli theorem [23], there exists a matrix $\mathbf{G} \in \mathbb{R}^{t\times n}$ such that:

$$\begin{bmatrix}\mathbf{K}\\ \mathbf{I}_n\end{bmatrix} = \mathscr{D}\mathbf{G}. \tag{10}$$

Substituting into the state-space representation, we have:

$$\mathbf{A}+\mathbf{B}\mathbf{K} = \begin{bmatrix}\mathbf{B} & \mathbf{A}\end{bmatrix}\begin{bmatrix}\mathbf{K}\\ \mathbf{I}_n\end{bmatrix} = \begin{bmatrix}\mathbf{B} & \mathbf{A}\end{bmatrix}\mathscr{D}\mathbf{G}. \tag{11}$$

On the other hand, the measured data in (6) must satisfy the system dynamics:

$$\mathbf{X}_1 = \mathbf{A}\mathbf{X}_0 + \mathbf{B}\mathbf{U}_0 + \mathbf{W}_0. \tag{12}$$

By substituting the definition of $\mathscr{D}$ from Lemma 1 into (11) and incorporating (12), we obtain:

$$\mathbf{A}+\mathbf{B}\mathbf{K} = (\mathbf{X}_1 - \mathbf{W}_0)\mathbf{G}. \tag{13}$$

Since $\mathbf{W}_0$ is unknown and cannot be directly accounted for, we approximate $\mathbf{A}+\mathbf{B}\mathbf{K}$ with $\mathbf{X}_1\mathbf{G}$ and $\mathbf{K}$ with $\mathbf{U}_0\mathbf{G}$ in the LQR optimization problem. This leads to the following direct data-driven LQR optimization formulation:

$$\begin{aligned}\min_{\mathbf{\Sigma_K}\succeq 0,\mathbf{G}} \quad & C(\mathbf{G}) = \text{Tr}\left(\left(\mathbf{Q}+\mathbf{G}^\top\mathbf{U}_0{}^\top\mathbf{R}\mathbf{U}_0\mathbf{G}\right)\mathbf{\Sigma_K}\right)\\ \text{subject to} \quad & \mathbf{\Sigma_K} = \mathbf{I}_n + \mathbf{X}_1\mathbf{G}\mathbf{\Sigma_K}\mathbf{G}^\top\mathbf{X}_1{}^\top,\\ & \mathbf{X}_0\mathbf{G} = \mathbf{I}_n.\end{aligned} \tag{14}$$

The optimal feedback gain is then given by $\mathbf{K} = \mathbf{U}_0\mathbf{G}$.

In [16], the authors introduce an alternative policy parametrization based on the sample covariance of the data, defined as:

$$\mathbf{\Phi} \triangleq \frac{1}{t}\mathscr{D}\mathscr{D}^\top = \begin{bmatrix}\mathbf{U}_0\mathscr{D}^\top/t\\ \mathbf{X}_0\mathscr{D}^\top/t\end{bmatrix} = \begin{bmatrix}\overline{\mathbf{U}}_0\\ \overline{\mathbf{X}}_0\end{bmatrix}. \tag{15}$$

Defining $\mathbf{V} \in \mathbb{R}^{(n+m)\times n}$ as the solution to:

$$\begin{bmatrix}\mathbf{K}\\ \mathbf{I}_n\end{bmatrix} = \mathbf{\Phi}\mathbf{V}, \tag{16}$$

and following steps similar to (11)–(13), the data-driven LQR optimization problem can be reformulated as:

$$\begin{aligned}\min_{\mathbf{\Sigma_V}\succeq 0,\mathbf{V}} \quad & C(\mathbf{V}) = \text{Tr}\left(\left(\mathbf{Q}+\mathbf{V}^\top\overline{\mathbf{U}}_0^\top\mathbf{R}\overline{\mathbf{U}}_0\mathbf{V}\right)\mathbf{\Sigma_V}\right)\\ \text{subject to} \quad & \mathbf{\Sigma_V} = \mathbf{I}_n + \overline{\mathbf{X}}_1\mathbf{V}\mathbf{\Sigma_V}\mathbf{V}^\top\overline{\mathbf{X}}_1^\top,\\ & \overline{\mathbf{X}}_0\mathbf{V} = \mathbf{I}_n,\end{aligned} \tag{17}$$

where $\overline{\mathbf{X}}_1 = \mathbf{X}_1\mathscr{D}^\top/t$. Since the dimension of $\mathbf{V}$ is independent of the number of samples, t, this formulation is particularly advantageous in adaptive design strategies where the sample size grows linearly.

In the DeePO algorithm, starting from an initial feasible solution $\mathbf{K}_t$, the feedback gain evolves iteratively via a gradient descent approach to reach the optimal solution $\mathbf{K}^*$. The following lemma provides a methodology for computing $\widehat{\nabla_{\mathbf{V}}C}$ solely based on data.

Lemma 2 ([16]): Let $\mathbf{P_V}$ be the unique solution for the Lyapunov equation

$$\mathbf{P_V} = \mathbf{Q} + \mathbf{V}^\top\overline{\mathbf{U}}_0^\top\mathbf{R}\overline{\mathbf{U}}_0\mathbf{V} + \mathbf{V}^\top\overline{\mathbf{X}}_1^\top\mathbf{P_V}\overline{\mathbf{X}}_1\mathbf{V},$$

then

$$\widehat{\nabla_{\mathbf{V}}C} = 2\left(\overline{\mathbf{U}}_0^\top\mathbf{R}\overline{\mathbf{U}}_0 + \overline{\mathbf{X}}_1^\top\mathbf{P_V}\overline{\mathbf{X}}_1\right)\mathbf{V}\mathbf{\Sigma_V}.$$

■

Algorithm 1 outlines the steps required to execute the DeePO algorithm.

Algorithm 1 Data-Enabled Policy Optimization (DeePO) [16]

Require: $\mathbf{U}_0, \mathbf{X}_0, \mathbf{X}_1, \mathbf{K}_t$, a probing noise signal $\{\mathbf{e}\}$, and a learning rate η.

Start

$i = t$.

while the stop criterion is not satisfied, do:

Apply $\mathbf{u}_i = \mathbf{K}_i\mathbf{x}_i + \mathbf{e}_i$ and observe $\mathbf{x}_{i+1}$.

Update $\mathbf{X}_0$ by $\mathbf{X}_0 = [\mathbf{X}_0, \mathbf{x}_i]$.

Update $\mathbf{X}_1$ by $\mathbf{X}_1 = [\mathbf{X}_1, \mathbf{x}_{i+1}]$.

Update $\mathbf{U}_0$ by $\mathbf{U}_0 = [\mathbf{U}_0, \mathbf{u}_i]$.

$\mathbf{V}_{i+1} = \mathbf{\Phi}_{i+1}^{-1}\begin{bmatrix}\mathbf{K}_i\\ \mathbf{I}_n\end{bmatrix}$.

$\mathbf{V}'_{i+1} = \mathbf{V}_{i+1} - \eta\Pi_{\overline{\mathbf{X}}_0}\widehat{\nabla_{\mathbf{V}}C}$.

Update the control gain by $\mathbf{K}_{i+1} = \overline{\mathbf{U}}_0\mathbf{V}'_{i+1}$.

$i = i+1$.

End while

End

In summary, to ensure the convergence of Algorithm 1, the optimization problem (17) must be feasible, $\mathbf{U}_0$ must be persistently exciting, and both $\boldsymbol{\omega}_k$ and $\mathbf{x}_k$ must remain bounded to maintain stability.

B. Why is probing noise added to $\mathbf{u}_i$ in Algorithm 1?

To begin, we present a lemma demonstrating that state feedback control of the form $\mathbf{u}_k = \mathbf{K}\mathbf{x}_k$ is incapable of generating a persistently exciting sequence $\mathbf{U}_0$.

Lemma 3: State feedback control of the form $\mathbf{u}_k = \mathbf{K}\mathbf{x}_k$, where $\mathbf{K}$ is a gain matrix and $\mathbf{x}_k$ is the state vector at time step k, cannot generate a persistently exciting sequence $\mathbf{U}_0$.

Proof: From Lemma 1, we construct the matrix $\mathscr{D}$ as follows:

$$\mathscr{D} = \begin{bmatrix}\sum_{j=1}^n \mathbf{k}^{1,j}\mathbf{x}_0^j & \sum_{j=1}^n \mathbf{k}^{1,j}\mathbf{x}_1^j & \cdots & \sum_{j=1}^n \mathbf{k}^{1,j}\mathbf{x}_{t-1}^j\\ \vdots & \vdots & \ddots & \vdots\\ \sum_{j=1}^n \mathbf{k}^{m,j}\mathbf{x}_0^j & \sum_{j=1}^n \mathbf{k}^{m,j}\mathbf{x}_1^j & \cdots & \sum_{j=1}^n \mathbf{k}^{m,j}\mathbf{x}_{t-1}^j\\ \mathbf{x}_0^1 & \mathbf{x}_1^1 & \cdots & \mathbf{x}_{t-1}^1\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{x}_0^n & \mathbf{x}_1^n & \cdots & \mathbf{x}_{t-1}^n\end{bmatrix}, \tag{18}$$

where first m rows of $\mathscr{D}$ can be expressed as linear combinations of the last n rows. Consequently, $\text{rank}(\mathscr{D}) = n$. From Lemma 1, we observe that the condition $\text{rank}(\mathscr{D}) \neq n+m$

implies that the sequence $\mathbf{U}_0$ is not persistently exciting. This completes the proof. ∎

In addition, the following lemma provides a more transparent perspective on the subject.

Lemma 4: Let $\mathscr{D}$ be defined as the horizontal concatenation of two matrices, $\mathscr{D} = [\mathscr{D}_1, \mathscr{D}_2]$,

where:

- $\mathscr{D}_1 \in \mathbb{R}^{(n+m)\times t_1}$ with $\text{rank}(\mathscr{D}_1) = n+m$, and
- $\mathscr{D}_2 \in \mathbb{R}^{(n+m)\times t_2}$ with $\text{rank}(\mathscr{D}_2) < n+m$.

If $t_2 \to \infty$, then $\lim_{t_2\to\infty} \underline{\sigma}(\Phi) \to 0$.

Proof: First, notice that Φ defined in (15) is only a function of t_2, since t_1 is fixed:

$$\Phi(t_2) = \frac{1}{t_1+t_2}(\mathscr{D}_1\mathscr{D}_1^\top + \mathscr{D}_2\mathscr{D}_2^\top). \tag{19}$$

Since $\mathscr{D}_2$ is not full row rank for all t_2, there exists a non-zero vector $\mathbf{a} \in \mathbb{R}^{n+m}$ such that $\mathbf{a}^\top\mathscr{D}_2 = 0$. Thus

$$\begin{aligned}\mathbf{a}^\top\Phi(t_2)\mathbf{a} &= \frac{1}{t_1+t_2}(\mathbf{a}^\top\mathscr{D}_1\mathscr{D}_1^\top\mathbf{a} + \mathbf{a}^\top\mathscr{D}_2\mathscr{D}_2^\top\mathbf{a}) \\ &= \frac{1}{t_1+t_2}\mathbf{a}^\top\mathscr{D}_1\mathscr{D}_1^\top\mathbf{a}.\end{aligned} \tag{20}$$

From this, it is clear that

$$\lim_{t_2\to\infty} \mathbf{a}^\top\Phi(t_2)\mathbf{a} = 0. \tag{21}$$

Since $\underline{\sigma}(\Phi(t_2))$ is such that

$$\underline{\sigma}(\Phi(t_2))\|\mathbf{a}\|^2 \le \mathbf{a}^\top\Phi(t_2)\mathbf{a}, \quad \forall t_2 \in \mathbb{N}, \tag{22}$$

then the limit (21) implies

$$\lim_{t_2\to\infty} \underline{\sigma}\big(\Phi(t_2)\big) \to 0.$$

This concludes the proof. ∎

The results from Lemmas 3 and 4 reveal critical limitations for applying conventional control laws, such as $\mathbf{u}_i = \mathbf{K}_i\mathbf{x}_i$, to the system:

- As $\mathbf{x}_i \to 0$, the columns of $\mathscr{D}$ also approach zero, and $\mathscr{D}$ can be partitioned such that the second matrix is zero. By Lemma 4, $\lim_{t_2\to\infty} \underline{\sigma}(\Phi) \to 0$, jeopardizing the full rank of Φ needed to ensure (16).
- As $\mathbf{K}_i \to \mathbf{K}$, Lemma 3 indicates that adding columns to $\mathscr{D}$ is equivalent to appending a singular matrix to $\mathscr{D}_1$, pushing Φ toward singularity.

To address this undesired outcome, which compromises DeePO's performance, the authors in [16] propose adding a probing noise to the input. This noise ensures that the input remains sufficiently rich, thereby preserving the full rank of Φ and maintaining persistent excitation.

C. Undesired State Perturbations by Adding Probing Noise to $\mathbf{u}_i$

In asymptotically stable LTI systems, the system naturally drives the state $\mathbf{x}_i$ towards equilibrium as time progresses. However, when probing noise is added to the input signal to maintain persistent excitation, the noise introduces high-frequency components into the control input. These high-frequency components can interact with the feedback dynamics, causing rapid oscillations or fluctuations in the control signal and, consequently, the system state. States perturbations is particularly problematic in practical implementations, as it can lead to actuator wear, increased energy consumption, and degraded overall system performance.

III. Perturbations-free DeePO (PFDeePO)

As discussed in Section II, the convergence of $\mathbf{x}_i \to 0$ and $\mathbf{K}_i \to \mathbf{K}$ can compromise the full rank of Φ. In Algorithm 2, we present our proposed method, PFDeePO, which serves as an alternative to DeePO, ensuring the full rank of Φ is preserved while avoiding performance degradation caused by state perturbations.

Algorithm 2 Perturbations-free DeePO (PFDeePO)

Require: $\mathbf{U}_0$, $\mathbf{X}_0$, $\mathbf{X}_1$, $\mathbf{K}_t$, $\gamma > 0$, $\delta > 0$, and $\eta > 0$.
Start
$i = t$.
$\Delta\mathbf{K} = (\delta+1)\cdot\mathbf{1}_{m\times n}$.
while the stop criterion is not satisfied, do:
if $\|\Delta\mathbf{K}\| > \delta$ **or** $\|\mathbf{x}_i\| \le \gamma$
Apply $\mathbf{u}_i = \mathbf{K}_i\mathbf{x}_i$ and observe $\mathbf{x}_{i+1}$.
else
Find $\bar{v}$ and $\underline{v}$ as in Theorem 2 for $\mathbf{K}_i$.
Randomly select $\underline{v} \le v_i \le \bar{v}$.
Apply $\mathbf{u}_i = v_i\mathbf{K}_i\mathbf{x}_i$ and observe $\mathbf{x}_{i+1}$.
End if
if $\|\mathbf{x}_i\| > \gamma$
Update $\mathbf{X}_0$ by $\mathbf{X}_0 = [\mathbf{X}_0, \mathbf{x}_i]$.
Update $\mathbf{X}_1$ by $\mathbf{X}_1 = [\mathbf{X}_1, \mathbf{x}_{i+1}]$.
Update $\mathbf{U}_0$ by $\mathbf{U}_0 = [\mathbf{U}_0, \mathbf{u}_i]$.
$\mathbf{V}_{i+1} = \Phi_{i+1}^{-1}\begin{bmatrix}\mathbf{K}_i \\ \mathbf{I}_n\end{bmatrix}$.
$\mathbf{V}'_{i+1} = \mathbf{V}_{i+1} - \eta\Pi_{\overline{\mathbf{X}}_0}\widehat{\nabla_{\mathbf{V}}C}$.
Update the control gain by $\mathbf{K}_{i+1} = \overline{\mathbf{U}}_0\mathbf{V}'_{i+1}$.
else
Update the control gain by $\mathbf{K}_{i+1} = \mathbf{K}_i$.
End if
$\Delta\mathbf{K} = \mathbf{K}_{i+1} - \mathbf{K}_i$.
$i = i+1$.
End while
End

The main idea behind PFDeePO is to prevent conditions that may compromise the full rank of Φ. To achieve this, we introduce two additional positive scalars, $\gamma > 0$ and $\delta > 0$.

At each iteration, the first "if-else" block ensures that arbitrarily scaling the control signal by a random gain, v_i, where $\underline{v} \le v_i \le \bar{v}$, occurs only when necessary. Specifically, this scaling is applied only if the control gain, $\mathbf{K}_i$, has already converged to its optimal value, yet the system states remain relatively far from the equilibrium point.

Next, in the second "if-else" block, we evaluate the convergence of the states. The condition $\|\mathbf{x}_i\| < \gamma$ represents a scenario where the states are close to the equilibrium point, causing the control signal to approach zero. In this situation,

the data gathered is not sufficiently informative to update the controller. Conversely, if the states have already reached near-equilibrium, further efforts to improve the control law are unnecessary, as the states have effectively converged, and any additional effort is unlikely to yield meaningful improvements in classical control performance metrics.

To formalize PFDeePO, we first need to ensure that the matrix Φ_i retains full rank when Algorithm 2 is applied.

Theorem 1: Let Φ_i be the matrix constructed at time step i during the execution of PFDeePO. By implementing Algorithm 2, the matrix Φ_i attains full rank, i.e., $\text{rank}(\Phi_i) = n+m$, and as a result, $\underline{\sigma}(\Phi_i) > 0$.

Proof: From (15) we see that $\text{rank}(\Phi_i) = \text{rank}(\mathscr{D})$. We proceed by contradiction. Assume that $\text{rank}(\mathscr{D}) < n+m$. This implies the existence of scalars $\lambda_i, \mu_j \in \mathbb{R}$ such that:

$$\sum_{i=1}^{n} \lambda_i \mathbf{X}_0{}^i = \sum_{j=1}^{m} \mu_j \mathbf{U}_0{}^j, \tag{23}$$

where $\mathbf{X}_0{}^i$ denotes the i-th row of $\mathbf{X}_0$, and $\mathbf{U}_0{}^j$ denotes the j-th row of $\mathbf{U}_0$. Note that the second if-else block, avoids presence of all zeros columns in $\mathbf{X}_0$ and $\mathbf{U}_0$ and recall that at each time step k, $\mathbf{U}_0{}^{j,k} = \sum_{i=1}^{n} \mathbf{k}_k^{j,i} \mathbf{x}_k^i$, which holds for all $k = 0, \ldots, t-1$, we substitute into (23) to obtain:

$$\sum_{i=1}^{n} \lambda_i \mathbf{x}_k^i = \sum_{j=1}^{m} \sum_{i=1}^{n} \mu_j \mathbf{k}_k^{j,i} \mathbf{x}_k^i. \tag{24}$$

For (24) to hold for all $k = 0, \ldots, t-1$ and $i = 1, \ldots, n$, we must have:

$$\lambda_i = \sum_{j=1}^{m} \mu_j \mathbf{k}_k^{j,i}. \tag{25}$$

However, this condition (25) is violated under both cases defined by Algorithm 2:

- When $\mathbf{K}_k \neq \mathbf{K}_{k-1}$, the matrices $\mathbf{K}_k$ are distinct, and it is impossible for λ_i to be expressed as a consistent linear combination of $\mu_j \mathbf{k}_k^{j,i}$. In other words, since the i-th column of the controller changes at each time step, the value of λ_i obtained at time step k as a linear combination of $\mu_j \mathbf{k}_k^{j,i}$ will differ from the values obtained at other time steps and it is not feasible to find a consistent λ_i for all $k = 0, \ldots, t-1$.
- When $\mathbf{K}_k = v_k \mathbf{K}$, where v_i is randomly sampled from an arbitrary distribution, the randomness in v_k ensures that λ_i will be different value at each time step in $\lambda_i = \sum_{j=1}^{m} v_k \mu_j \mathbf{k}^{j,i}$.

Thus, the assumption that $\text{rank}(\mathscr{D}) < n+m$ leads to a contradiction. Therefore, we conclude that:

$$\text{rank}(\mathscr{D}) = n+m.$$

■

Another critical aspect to address is ensuring that randomly scaling the control signal by v_i does not compromise the closed-loop stability. First recall that from Lemma 1 in [16], the converged control gain $\mathbf{K}$ obtained by DeePO is equivalent to indirect LQR solution. Therefore,

$$\mathbf{K} = \left(\hat{\mathbf{B}}^\top \mathbf{H} \hat{\mathbf{B}} + \mathbf{R}\right)^{-1} \hat{\mathbf{B}}^\top \mathbf{H} \hat{\mathbf{A}}, \tag{26}$$

where $\mathbf{H}$ is the solution of the following Discrete Algebraic Riccati Equation (DARE)

$$\mathbf{H} = \hat{\mathbf{A}}^\top \mathbf{H} \hat{\mathbf{A}} + \mathbf{Q} - \hat{\mathbf{A}}^\top \mathbf{H} \hat{\mathbf{B}} \left(\hat{\mathbf{B}}^\top \mathbf{H} \hat{\mathbf{B}} + \mathbf{R}\right)^{-1} \hat{\mathbf{B}}^\top \mathbf{H} \hat{\mathbf{A}}. \tag{27}$$

and $\mathbf{Q} \in \mathbb{R}^{n \times n}$ and $\mathbf{R} \in \mathbb{R}^{m \times m}$ are given symmetric positive definite matrices. In the following, we establish the conditions that $\bar{v}$ and $\underline{v}$ must satisfy to guarantee stability in Algorithm 2. We begin by the following lemma.

Lemma 5: Consider matrices $\hat{\mathbf{B}} \in \mathbb{R}^{n \times m}$ and $\mathbf{K} \in \mathbb{R}^{m \times n}$, and symmetric positive definite matrices $\mathbf{Q} \in \mathbb{R}^{n \times n}$ and $\mathbf{R} \in \mathbb{R}^{m \times m}$. Then, there exists an interval $\mathscr{V} = [\underline{v}, \bar{v}]$, where $0 < \underline{v} < 1 < \bar{v}$, such that

$$\mathbf{Q} - \mathbf{K}^\top \left((v-1)^2 \hat{\mathbf{B}}^\top \mathbf{H} \hat{\mathbf{B}} + (1-2v)\mathbf{R}\right) \mathbf{K} \succeq 0, \quad \forall v \in [\underline{v}, \bar{v}]. \tag{28}$$

Proof: Note first that for $v = 1$, we have

$$\begin{aligned} \left(\mathbf{Q} - \mathbf{K}^\top \left((v-1)^2 \hat{\mathbf{B}}^\top \mathbf{H} \hat{\mathbf{B}} + (1-2v)\mathbf{R}\right) \mathbf{K}\right)_{v=1} \\ = \mathbf{Q} + \mathbf{K}^\top \mathbf{R} \mathbf{K} \succ 0. \end{aligned} \tag{29}$$

Since $\mathbf{Q} \succ 0$ by assumption and $\mathbf{K}^\top \mathbf{R} \mathbf{K} \succeq 0$, the matrix in the left-hand side of (28) is positive definite at $v = 1$. Recall that the eigenvalues of a square matrix, whose components depend continuously on a parameter, are also continuous functions of that parameter. As a result, there exists an open neighborhood $(\underline{v}, \bar{v})$ around $v = 1$ where all eigenvalues of the matrix remain strictly positive, ensuring that the matrix is positive definite.

At the endpoints $\underline{v}$ and $\bar{v}$, the matrix becomes positive semidefinite, meaning some of its eigenvalues reach zero. Consequently, the inequality (28) holds for all $v \in [\underline{v}, \bar{v}]$, completing the proof. ■

Remark 1: In Lemma 5, we show that there exists an interval $\mathscr{V} = [\underline{v}, \bar{v}]$ such that (28) holds for all $v \in \mathscr{V}$. A practical approach to determine $\mathscr{V}$ is to start with a small interval around 1 and iteratively narrow it, checking Lemma 5 at each step, until the inequality is satisfied. □

Now, we are ready to present our main theorem.

Theorem 2: Consider a system controlled by Algorithm 2. Let $\mathbf{Q} \in \mathbb{R}^{n \times n}$ and $\mathbf{R} \in \mathbb{R}^{m \times m}$ be given symmetric and positive definite matrices, and let $\beta \in (0, 1)$.

Let's represent the converged control gain by the matrix $\mathbf{K} \in \mathbb{R}^{m \times n}$, since $\mathbf{K}$ is equal to a certainly equivalent LQR gain, we have

$$\mathbf{K} = (\mathbf{R} + \hat{\mathbf{B}}^\top \mathbf{H} \hat{\mathbf{B}})^{-1} \hat{\mathbf{B}}^\top \mathbf{H} \hat{\mathbf{A}}, \tag{30}$$

where $\mathbf{H} \in \mathbb{R}^{n \times n}$ is the unique solution of the modified DARE:

$$\hat{\mathbf{A}}^\top \mathbf{H} \hat{\mathbf{A}} - \beta^2 \mathbf{H} - \hat{\mathbf{A}}^\top \mathbf{H} \hat{\mathbf{B}} (\mathbf{R} + \hat{\mathbf{B}}^\top \mathbf{H} \hat{\mathbf{B}})^{-1} \hat{\mathbf{B}}^\top \mathbf{H} \hat{\mathbf{A}} + \mathbf{Q} = \mathbf{0}_{n \times n}. \tag{31}$$

Consider the interval $[\underline{v}, \bar{v}]$, as defined in Lemma 5, such that the inequality (28) holds. Then, the time-varying state feedback control law, $\mathbf{u}_k = -v_k \mathbf{K} \mathbf{x}_k$, where v_k is any sequence

taking values in the interval $[\underline{v},\overline{v}]$, ensures that the origin of the closed-loop system

$$\mathbf{x}_{k+1} = (\hat{\mathbf{A}} - v_k\hat{\mathbf{B}}\mathbf{K})\mathbf{x}_k, \quad k = 0,1,\dots \tag{32}$$

is exponentially stable. Specifically, the state satisfies the bound

$$\|\mathbf{x}_k\| \le \beta^k \sqrt{\frac{h_{\max}}{h_{\min}}}\|\mathbf{x}_0\|, \quad k = 0,1,\dots, \tag{33}$$

where $h_{\max} = \lambda_{\max}(\mathbf{H})$, and $h_{\min} = \lambda_{\min}(\mathbf{H})$.

Proof: For a compact notation, the gain $\mathbf{K}$ defined in (30) will be written as

$$\mathbf{K} = \mathbf{S}^{-1}\hat{\mathbf{B}}^\top\mathbf{H}\hat{\mathbf{A}}, \quad \text{where} \quad \mathbf{S} = \mathbf{R} + \hat{\mathbf{B}}^\top\mathbf{H}\hat{\mathbf{B}}, \tag{34}$$

so that the term $\hat{\mathbf{A}}^\top\mathbf{H}\hat{\mathbf{B}}(\mathbf{R}+\hat{\mathbf{B}}^\top\mathbf{H}\hat{\mathbf{B}})^{-1}\hat{\mathbf{B}}^\top\mathbf{H}\hat{\mathbf{A}}$ in the modified DARE (31) can be equivalently written as

$$\hat{\mathbf{A}}^\top\mathbf{H}\hat{\mathbf{B}}\mathbf{S}^{-1}\hat{\mathbf{B}}^\top\mathbf{H}\hat{\mathbf{A}} = \hat{\mathbf{A}}^\top\mathbf{H}\hat{\mathbf{B}}\mathbf{K} = \mathbf{K}^\top\hat{\mathbf{B}}^\top\mathbf{H}\hat{\mathbf{A}} = \mathbf{K}^\top\mathbf{S}\mathbf{K}, \tag{35}$$

and the modified DARE (31) can be rewritten as

$$\hat{\mathbf{A}}^\top\mathbf{H}\hat{\mathbf{A}} - \beta^2\mathbf{H} - \mathbf{K}^\top\mathbf{S}\mathbf{K} + \mathbf{Q} = \mathbf{0}_{n\times n}. \tag{36}$$

From this, we get the identity

$$\hat{\mathbf{A}}^\top\mathbf{H}\hat{\mathbf{A}} - \beta^2\mathbf{H} = -\mathbf{Q} + \mathbf{K}^\top\mathbf{S}\mathbf{K}. \tag{37}$$

Now, consider the Lyapunov function candidate $V(\mathbf{x}) = \mathbf{x}^\top\mathbf{H}\mathbf{x}$. We will prove that

$$V(\mathbf{x}_{k+1}) \le \beta^2 V(\mathbf{x}_k), \quad k = 0,1,\dots, \tag{38}$$

independently of the sequence v_k, provided that $v_k \in \mathscr{V}$, so that

$$V(\mathbf{x}_k) \le \beta^{2k} V(\mathbf{x}_0), \quad k = 0,1,\dots. \tag{39}$$

From this, recalling that $h_{\min}\|\mathbf{x}\|^2 \le \mathbf{x}^\top\mathbf{H}\mathbf{x} \le h_{\max}\|\mathbf{x}\|^2$, we easily get

$$h_{\min}\|\mathbf{x}_k\|^2 \le \beta^{2k} h_{\max}\|\mathbf{x}_0\|^2, \tag{40}$$

from which, taking the square roots of both terms of the inequality, the inequality (33) follows.

Thus, to prove the Theorem, i.e., the inequalities (33), it is sufficient to prove the inequalities (38). These can be rewritten as

$$V(\mathbf{x}_{k+1}) - \beta^2 V(\mathbf{x}_k) \le 0, \quad k = 0,1,\dots, \tag{41}$$

or

$$\mathbf{x}_{k+1}^\top\mathbf{H}\mathbf{x}_{k+1} - \beta^2\mathbf{x}_k^\top\mathbf{H}\mathbf{x}_k \le 0, \quad k = 0,1,\dots. \tag{42}$$

Exploiting the system equation (32), these inequalities change to the following

$$\mathbf{x}_k^\top\big((\hat{\mathbf{A}} - v_k\hat{\mathbf{B}}\mathbf{K})^\top\mathbf{H}(\hat{\mathbf{A}} - v_k\hat{\mathbf{B}}\mathbf{K}) - \beta^2\mathbf{H}\big)\mathbf{x}_k \le 0, \tag{43}$$

where $v_k \in [\underline{v},\overline{v}]$. It is clear that all inequalities (43) are verified if the following inequality is true:

$$(\hat{\mathbf{A}} - v\hat{\mathbf{B}}\mathbf{K})^\top\mathbf{H}(\hat{\mathbf{A}} - v\hat{\mathbf{B}}\mathbf{K}) - \beta^2\mathbf{H} \le 0, \quad \forall v \in [\underline{v},\overline{v}]. \tag{44}$$

Upon carrying out the multiplications in the left-hand side term of the inequality and considering the identities (35) and (37), we have

$$\begin{aligned}
&(\hat{\mathbf{A}} - v\hat{\mathbf{B}}\mathbf{K})^\top\mathbf{H}(\hat{\mathbf{A}} - v\hat{\mathbf{B}}\mathbf{K}) - \beta^2\mathbf{H} \\
&= \hat{\mathbf{A}}^\top\mathbf{H}\hat{\mathbf{A}} - \beta^2\mathbf{H} - v\mathbf{K}^\top\hat{\mathbf{B}}^\top\mathbf{H}\hat{\mathbf{A}} - v\hat{\mathbf{A}}^\top\mathbf{H}\hat{\mathbf{B}}\mathbf{K} \\
&\quad + v^2\mathbf{K}^\top\hat{\mathbf{B}}^\top\mathbf{H}\hat{\mathbf{B}}\mathbf{K} \\
&= \hat{\mathbf{A}}^\top\mathbf{H}\hat{\mathbf{A}} - \beta^2\mathbf{H} - 2v\mathbf{K}^\top\mathbf{S}\mathbf{K} + v^2\mathbf{K}^\top\hat{\mathbf{B}}^\top\mathbf{H}\hat{\mathbf{B}}\mathbf{K} \\
&= -\mathbf{Q} + \mathbf{K}^\top\mathbf{S}\mathbf{K} - 2v\mathbf{K}^\top\mathbf{S}\mathbf{K} + v^2\mathbf{K}^\top\hat{\mathbf{B}}^\top\mathbf{H}\hat{\mathbf{B}}\mathbf{K} \\
&= -\mathbf{Q} + \mathbf{K}^\top\big((1-2v)\mathbf{S} + v^2\hat{\mathbf{B}}^\top\mathbf{H}\hat{\mathbf{B}}\big)\mathbf{K}.
\end{aligned} \tag{45}$$

Recalling that $\mathbf{S} = \mathbf{R} + \hat{\mathbf{B}}^\top\mathbf{H}\hat{\mathbf{B}}$, we get (44):

$$\begin{aligned}
&(\hat{\mathbf{A}} - v\hat{\mathbf{B}}\mathbf{K})^\top\mathbf{H}(\mathbf{A} - v\hat{\mathbf{B}}\mathbf{K}) - \beta^2\mathbf{H} \\
&\quad = -\mathbf{Q} + \mathbf{K}^\top\big((v-1)^2\hat{\mathbf{B}}^\top\mathbf{H}\hat{\mathbf{B}} + (1-2v)\mathbf{R}\big)\mathbf{K}.
\end{aligned} \tag{46}$$

Thus, the inequality (44) can be written equivalently in the form (28) of Lemma 5, from which we know that there exists a nonempty interval $[\underline{v},\overline{v}]$ such that the matrix (46) is negative semidefinite. It follows that all inequalities (43) are satisfied as long as $v_k \in [\underline{v},\overline{v}]$, and this implies inequality (33), and the Theorem is proved. ■

Remark 2: As discussed in Theorem 2, $\hat{\mathbf{B}}$ is essential for verifying (28) and determining $v \in [\underline{v},\overline{v}]$ to ensure stability. If $\hat{\mathbf{B}}$ is unknown, it can be estimated from collected data using

$$\begin{bmatrix}\hat{\mathbf{B}} & \hat{\mathbf{A}}\end{bmatrix} = \mathbf{X}_1\begin{bmatrix}\mathbf{U}_0\\ \mathbf{X}_0\end{bmatrix}^\dagger. \tag{47}$$

Since any interval $(v_m, v_M) \subseteq [\underline{v},\overline{v}]$ with $v_m < 1 < v_M$ can replace $\mathscr{V}$ in Algorithm 2, choosing a narrower interval is preferable, as it reduces control deviations from the optimum and improves stability margins. □

IV. Simulations

To evaluate Algorithm 2, the open loop and controllable LTI system in [16] is utilized in numerical simulations, with

$$A = \begin{bmatrix} -0.13 & 0.14 & -0.29 & 0.28 \\ 0.48 & 0.09 & 0.41 & 0.30 \\ -0.01 & 0.04 & 0.17 & 0.43 \\ 0.14 & 0.31 & -0.29 & -0.10 \end{bmatrix}, \; B = \begin{bmatrix} 1.63 & 0.93 \\ 0.26 & 1.79 \\ 1.46 & 1.18 \\ 0.77 & 0.11 \end{bmatrix}. \tag{48}$$

First, a persistently exciting input, $\mathbf{U}_0$, is used to generate the offline state data $\mathbf{X}_0$ and $\mathbf{X}_1$ from the system dynamics in (1). The noise is sampled from a normal distribution, $\boldsymbol{\omega}_k = \mathscr{N}(0,\sigma_{\boldsymbol{\omega}})$ and the offline input is sampled from $\mathbf{u}_k = \mathscr{N}(0,\sigma_u)$ with $\sigma_{\boldsymbol{\omega}} = \sigma_{\mathbf{u}} = 0.01$. The offline data consists of 8 time samples. Since we consider an open-loop stable system, the initial policy is kept at zero, i.e, $\mathbf{K}_t = \mathbf{0}_{m\times n}$. The parameters of PFDeePO are chosen as $\gamma = 0.1$, $\delta = 0.1$,$\eta = 10^{-4}$, and the interval of v is chosen as $\underline{v} = 0.5$ and $\overline{v} = 1.5$, which satisfies (28). DeePO, introduced in Algorithm 1, is used as a benchmark, with $\eta = 10^{-4}$ and a probing noise signal $\mathbf{e}$ sampled from a zero mean normal distribution with $\sigma_{\mathbf{e}} = 0.1$. At sample $k = 15$, a disturbance is induced in the states using a uniform random value.

In Fig. 1, we present the results of running both algorithms on the system. As observed, once the states reach equilibrium, PFDeePO does not introduce further perturbations. In

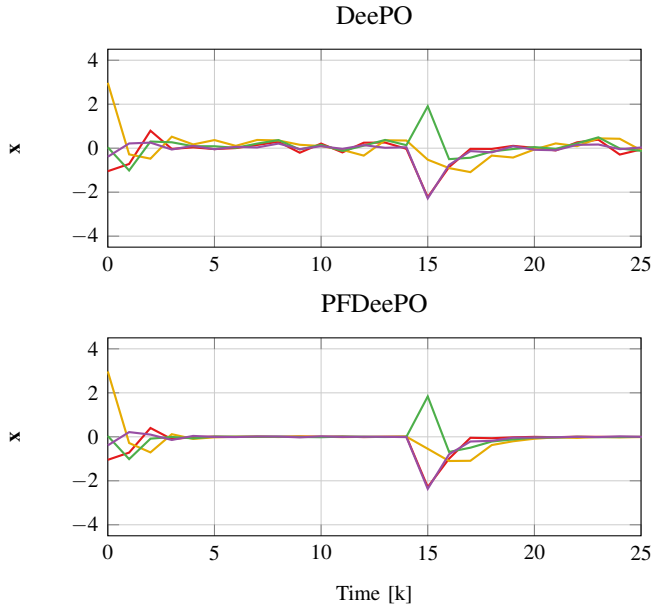

Fig. 1. Evolution of the states $\mathbf{x}$, using DeePO (top) and PFDeePO (bottom).

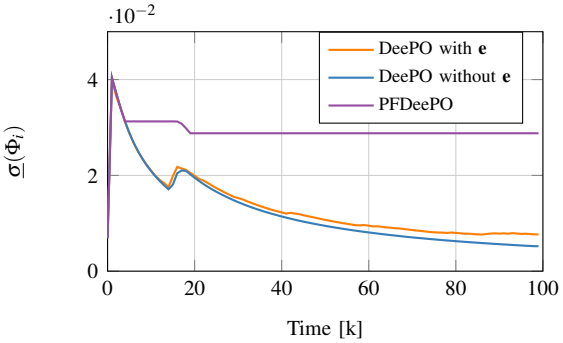

Fig. 2. $\underline{\sigma}(\Phi_i)$ for PFDeePO and DeePO, with and without probing noise.

contrast, the probing noise in DeePO continuously disturbs the states, inducing oscillations that increase control effort.

Fig. 2 illustrates evolution of $\underline{\sigma}(\Phi_i)$. It is evident that in DeePO, probing noise is essential to maintaining the full rank of Φ, as its minimum singular value otherwise approaches zero. However, with PFDeePO, $\underline{\sigma}(\Phi_i)$ remains consistently above zero throughout the control period, ensuring the rank condition is met without perturbing the system states.

V. Conclusion

Data-driven control design has gained significant attention, but the field remains in its early stages, with several open challenges. Among recent approaches for LTI systems, DeePO is a promising method that refines control gains online through adaptive updates. However, its reliance on probing noise to ensure persistency of excitation can degrade performance and induce state perturbations. This article introduces PFDeePO, which alleviates these issues. Simulations show that PFDeePO outperforms conventional DeePO in terms of control performance. Future work includes extending PFDeePO to LTV and nonlinear systems and optimizing its design parameter selection.

References

[1] J. C. Willems, P. Rapisarda, I. Markovsky, and B. L. De Moor, "A note on persistency of excitation," *Systems & Control Letters*, vol. 54, no. 4, pp. 325–329, 2005.

[2] C.-T. Chen, *Linear System Theory and Design*, 3rd ed. Oxford University Press, 1999.

[3] A. Cohen, T. Koren, and Y. Mansour, "Learning linear-quadratic regulators efficiently with only $\sqrt{T}$ regret," in *Proceedings of the 36th International Conference on Machine Learning*, vol. 97, June 2019, pp. 1300–1309.

[4] H. Mania, S. Tu, and B. Recht, "Certainty equivalence is efficient for linear quadratic control," in *33rd International Conference on Neural Information Processing Systems (NIPS)*, December 2019, pp. 10 154 – 10 164.

[5] M. Ferizbegovic, J. Umenberger, H. Hjalmarsson, and T. B. Schön, "Learning robust LQ-controllers using application oriented exploration," *IEEE Control Systems Letters*, vol. 4, no. 1, pp. 19–24, 2020.

[6] L. Sforni, G. Carnevale, I. Notarnicola, and G. Notarstefano, "On-policy data-driven linear quadratic regulator via combined policy iteration and recursive least squares," in *62nd IEEE Conference on Decision and Control (CDC)*, 2023, pp. 5047–5052.

[7] C. De Persis and P. Tesi, "Formulas for data-driven control: Stabilization, optimality, and robustness," *IEEE Transactions on Automatic Control*, vol. 65, no. 3, pp. 909–924, 2020.

[8] H. Mohammadi, M. Soltanolkotabi, and M. R. Jovanović, "On the linear convergence of random search for discrete-time LQR," *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 989–994, 2021.

[9] C. De Persis and P. Tesi, "Low-complexity learning of linear quadratic regulators from noisy data," *Automatica*, vol. 128, p. 109548, 2021.

[10] F. Dörfler, P. Tesi, and C. De Persis, "On the role of regularization in direct data-driven LQR control," in *IEEE 61st Conference on Decision and Control (CDC)*, 2022, pp. 1091–1098.

[11] V. G. Lopez, M. Alsalti, and M. A. Müller, "Efficient off-policy Q-learning for data-based discrete-time LQR problems," *IEEE Transactions on Automatic Control*, vol. 68, no. 5, pp. 2922–2933, 2023.

[12] F. Dörfler, P. Tesi, and C. De Persis, "On the certainty-equivalence approach to direct data-driven LQR design," *IEEE Transactions on Automatic Control*, vol. 68, no. 12, pp. 7989–7996, 2023.

[13] W. Fan and J. Xiong, "Q-Learning methods for LQR control of completely unknown discrete-time linear systems," *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 5933–5943, 2025.

[14] N. Persson, M. Kaheni, and A. V. Papadopoulos, "A direct data-driven control design for autonomous bicycles," in *IEEE 20th International Conference on Automation Science and Engineering (CASE)*, 2024, pp. 114–120.

[15] F. Zhao, F. Dörfler, and K. You, "Data-enabled policy optimization for the linear quadratic regulator," in *62nd IEEE Conference on Decision and Control (CDC)*, 2023, pp. 6160–6165.

[16] F. Zhao, F. Dörfler, A. Chiuso, and K. You, "Data-enabled policy optimization for direct adaptive learning of the LQR," *IEEE Transactions on Automatic Control*, pp. 1–16, 2025, Early Access.

[17] F. Zhao, R. Leng, L. Huang, H. Xin, K. You, and F. Dörfler, "Direct adaptive control of grid-connected power converters via output-feedback data-enabled policy optimization," *arXiv preprint arXiv:2411.03909*, 2024.

[18] N. Persson, F. Zhao, M. Kaheni, F. Dörfler, and A. V. Papadopoulos, "An adaptive data-enabled policy optimization approach for autonomous bicycle control," *arXiv preprint arXiv:2502.13676*, 2025.

[19] B. D. O. Anderson and J. B. Moore, *Optimal Control: Linear Quadratic Methods*. Courier Corporation, 2007.

[20] Y. Sattar, Z. Du, D. A. Tarzanagh, S. Oymak, L. Balzano, and N. Ozay, "Certainty equivalent quadratic control for markov jump systems," in *American Control Conference (ACC)*, 2022, pp. 2871–2878.

[21] J. Pilipovsky and P. Tsiotras, "Data-driven covariance steering control design," in *62nd IEEE Conference on Decision and Control (CDC)*, 2023, pp. 2610–2615.

[22] W. Liu, G. Wang, J. Sun, F. Bullo, and J. Chen, "Learning robust data-based LQG controllers from noisy data," *IEEE Transactions on Automatic Control*, vol. 69, no. 12, pp. 8526–8538, 2024.

[23] I. R. Shafarevich and A. O. Remizov, *Linear Algebra and Geometry*, 1st ed. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2013.