IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 5 January 2023, accepted 27 January 2023, date of publication 3 February 2023, date of current version 15 February 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3241926

RESEARCH ARTICLE

Data-Driven Control Design With LMIs and Dynamic Programming

DONGHWAN LEE[1], (Member, IEEE), AND DO WAN KIM[2]
[1]Department of Electrical Engineering, KAIST, Daejeon 34141, South Korea
[2]Department of Electrical Engineering, Hanbat National University, Daejeon 34158, South Korea

Corresponding author: Donghwan Lee (donghwan@kaist.ac.kr)

This work was supported in part by the National Research Foundation under Grant NRF-2021R1I1A3058581; in part by the Institute of Information communications Technology Planning Evaluation (IITP) Grant through the Korea Government, Ministry of Science and ICT (MIST), under Grant 2022-0-00469; and in part by the brain korea 21 (BK21) FOUR from the Ministry of Education, Republic of Korea.

ABSTRACT The goal of this paper is to study model-free data-driven control evaluation and design strategies for discrete-time linear time-invariant systems, where the system model is unknown. In particular, our main contribution is twofold: 1) new state-input exploration and data collection schemes from experiences; 2) new data-driven linear matrix inequalities and dynamic programming methods for stabilization and optimal control problems. The proposed exploration and data collection schemes theoretically guarantee to acquire sufficient information from the system's state-input trajectories that can solve the underlying control design problems. We prove that under mild assumptions, as more and more data is accumulated, the collected data can solve the problems with higher probability along with the proposed algorithms.

INDEX TERMS Data-driven design, reinforcement learning, optimal control, linear matrix inequality (LMI), dynamic programming, linear time-invariant (LTI) system.

I. INTRODUCTION

Recently, reinforcement learning (RL) [1] and data-driven control design have captured significant attentions due to its successful demonstrations that outperform humans in several challenging tasks [2], [3]. The goal of this paper is to study 1) data-driven control design methods for discrete-time linear-time invariant (LTI) systems and 2) state-input exploration and data collection strategies that theoretically guarantee solvability of the underlying data-driven problems with high probability. For the data-driven control design methods, two different lines are addressed: linear matrix inequalities (LMIs) [4] and dynamic programming (DP) [5], [6]. Both data-driven approaches can solve stabilization and linear quadratic regulator (LQR) problems without the knowledge of models. The exploration and data collection algorithms are tailored to the proposed data-driven designs, and once collected, then no more data is required to solve the problems completely.

The associate editor coordinating the review of this manuscript and approving it for publication was Zhiguang Feng.

A. RELATED WORKS

The previous works can be roughly categorized into two parts: RL (or data-driven dynamic programming) and data-based LMIs. As for RL, the early work [7] proposed a Q-learning algorithm [8] for discrete-time LTI systems, where the approximate Bellman equation is solved using the least-square method and state-input trajectories. More comprehensive least-square RL approaches were reported in [9]. A model-based RL has been studied in [10] recently for discrete-time LTI systems with sample complexity analysis. A policy gradient algorithm for LTI systems and its global convergence were studied in [11]. An efficient online RL with guaranteed finite-time regret bounds has been proposed in [12] based on a novel semidefinite programming relaxation. The paper [13] proposed several model-based and model-free RLs. Reference [14] proposed a policy iteration RL based on the Lagrangian duality perspectives of the Bellman equation.

As for the data-based LMIs [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], several advances have been made recently in deriving numerically tractable data-based

LMIs that enable direct data-driven control designs. Data-dependent LMIs were developed in [15] for stabilization of switched systems. The paper [16] introduced a data-dependent controller parameterization, and proposed data-based LMIs for stabilization and optimal control problems. The concept of informative data was introduced in [17], from which necessary and sufficient data-based conditions have been developed for various control problems. The paper [18] proposed LMI conditions for control with guaranteed stability and performance by introducing a notion of noise bounds. Recently, [19] and [20] introduced data-driven LMI conditions for stabilization problems based on a matrix version of the classical Finsler's lemma [26].

Most of the previous works usually focus on 1) developing data-dependent LMI formulations, and 2) conditions on the data structures that guarantee the solvability of the underlying data-based control design problems. However, most of the previous works did not fully address how to generate data structures to meet the solvability conditions. Compared to the previous works, in this paper, we mainly focus on stochastic explorations schemes that theoretically guarantee solvability of the underlying data-driven problems with high probability, which are lacking in the literature to the authors' knowledge. Moreover, we also study the development of new data-driven control design methods in the forms of the dynamic programming and LMIs. The proposed new data-based LMI method can be computationally efficient alternatives in some cases as discussed in the main parts of this paper.

In particular, unique features of the proposed data-driven methods are summarized in more details as follows:

1) One of the main contributions is the proposition of exploration and data collection schemes, which theoretically guarantee solvability of the underlying data-driven problems with high probability. We prove that the new data collection approaches is guaranteed to solve the problem with probability one as more and more experiences are accumulated. In particular, the exploration and data collection methods that best match with the proposed methods are those in [10] and [11], which studied probabilistic finite-sample analysis. Compared to the existing methods, the proposed approach has unique aspects that are summarized in the main results.
2) Based on the data matrices collected from the proposed methods, we investigate new forms of data-driven DPs and LMIs that are tailored to the proposed methods. The proposed data-driven LMIs are more memory efficient than existing methods in the sense that the size of LMIs is independent of the length of the collected trajectory. Moreover, depending on the LMI solver used and the structure of the underlying problem, the proposed LMIs may be computationally more efficient in some cases. In this respect, we regard the proposed methods as complements rather than replacement of existing methods.

B. NOTATION

The adopted notation is as follows: $\mathbb{R}$: set of real numbers; $\mathbb{R}^n$: n-dimensional Euclidean space; $\mathbb{R}^{n\times m}$: set of all $n \times m$ real matrices; A^T: transpose of matrix A; A^{-T}: transpose of matrix A^{-1}; $A \succ 0$ ($A \prec 0$, $A \succeq 0$, and $A \preceq 0$, respectively): symmetric positive definite (negative definite, positive semi-definite, and negative semi-definite, respectively) matrix A; I: identity matrix with appropriate dimensions; $\mathbb{S}^n$: symmetric $n \times n$ matrices; $\mathbb{S}^n_+$: cone of symmetric $n \times n$ positive semi-definite matrices; $\mathbb{S}^n_{++}$: symmetric $n \times n$ positive definite matrices; $Tr(A)$: trace of matrix A; $\rho(\cdot)$: spectral radius; $\text{diag}(A_1, \ldots, A_n)$: block diagonal matrix with diagonal elements $A_1, \ldots, A_n$..

II. PROBLEM FORMULATIONS AND PRELIMINARIES

Throughout the paper, we consider the discrete-time linear time-invariant (LTI) system

$$x(k+1) = Ax(k) + Bu(k), \quad x(0) = z \in \mathbb{R}^n, \tag{1}$$

where $k \in \mathbb{N}$ is the time, $x(k) \in \mathbb{R}^n$ is the state vector, $u(k) \in \mathbb{R}^m$ is the input vector, and $z \in \mathbb{R}^n$ is the initial state.

Remark 1: Data-driven control design methods for such linear systems have been actively studied until recently [16], [17]. Although the system in (1) is simple, it covers a wide range of industrial applications, such as circuits and mechanical systems [27]. For instance, in many industrial applications, practical nonlinear systems are often approximated by linear systems, and simple PID controllers are the most widely used control strategy today for such complex industrial applications [28]. Moreover, rigorous study of such simple linear systems gives more insights on fundamental mechanisms and help us develop more advanced methodologies for complex and general systems.

Assuming the control $u(k)$ is given by a state-feedback control policy $u(k) = Fx(k)$, we denote by $x(k; F, z)$ the solution of (1) starting from $x(0) = z$. Under the state-feedback control policy, the cost function for the classical LQR problem is denoted by

$$J(F, z) := \sum_{k=0}^{\infty} \begin{bmatrix} x(k;F,z) \\ Fx(k;F,z) \end{bmatrix}^T \Lambda \begin{bmatrix} x(k;F,z) \\ Fx(k;F,z) \end{bmatrix}, \tag{2}$$

where $\Lambda := \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} \succeq 0$ is the weight matrix.

By introducing the augmented state vector $v(k) := \begin{bmatrix} x(k) \\ u(k) \end{bmatrix}$, we will consider the augmented system

$$v(k+1) = A_F v(k), \quad v(0) = v_0 \in \mathbb{R}^{n+m}, \tag{3}$$

where $A_F := \begin{bmatrix} A & B \\ FA & FB \end{bmatrix} \in \mathbb{R}^{(n+m)\times(n+m)}$, which plays an important role throughout the paper. A useful property of A_F is that its spectral radius $\rho(A_F)$ is identical to that of $A + BF$.

Lemma 1 [14]: $\rho(A + BF) = \rho(A_F)$ holds.

In this paper, we study both LQR and stabilization problems. The stabilization problem is to find a stabilizing feedback gain F such that $\rho(A + BF) < 1$. The

LQR problem is to solve $F^* = \arg\min_{F\in\mathbb{R}^{m\times n}} J(F,z)$ if the optimal value of $\inf_{F\in\mathbb{R}^{m\times n}} J(F,z)$ exists and is attained. From the standard LQR theory, although $J^*(F,z)$ has different values for different $z \in \mathbb{R}^n$, the minimizer $F^* = \arg\min_{F\in\mathbb{R}^{m\times n}} J(F,z)$ is not dependent on z. Therefore, it follows that $\arg\min_{F\in\mathbb{R}^{m\times n}} J(F,z) = \arg\min_{F\in\mathbb{R}^{m\times n}} \sum_{i=1}^{r} J(F,z_i)$ for any $z, z_i \in \mathbb{R}^n$, $i \in \{1,2,\ldots,r\}$. For technical reasons that will become clear later, we solve

$$F^* := \arg\min_{F\in\mathbb{R}^{m\times n}} \sum_{i=1}^{n} J(F,e_i)$$

instead of $\arg\min_{F\in\mathbb{R}^{m\times n}} J(F,z)$, where $e_i \in \mathbb{R}^n$ is the ith standard basis vector. Therefore, it will be useful to define a standard measure of the cost. In this paper, we will use the following cost index:

$$J(F) := \sum_{i=1}^{n} J(F,e_i)$$

For a given $z \in \mathbb{R}^n$, if the optimal value of $\inf_{F\in\mathbb{R}^{m\times n}} J(F,z)$ exists and is attained, then the optimal cost is denoted by $J^*(z) = J(F^*)$. Assumptions that will be used throughout the paper are summarized below.

Assumption 1: Throughout the paper, we assume that

- $Q \succeq 0, R \succ 0$;
- (A,B) *is stabilizable, and* Q *can be written as* $Q = C^TC$, *where* (A,C) *is detectable.*

Under Assumption 1, the optimal value of $\inf_{F\in\mathbb{R}^{m\times n}} J(F)$ exists, is attained, and $J^*(z)$ is a quadratic function, i.e., $J^*(z) = z^TX^*z$, where X^* is the unique solution of the algebraic Riccati equation (ARE) [5, Proposition 4.4.1] for $X \succeq 0$:

$$X = A^TXA - A^TXB(R+B^TXB)^{-1}B^TXA + Q.$$

In this case, $J^*(z)$ as a function of $z \in \mathbb{R}^n$ is called the optimal value function. The reader can refer to [5] and [29] for more details of the classical LQR results. The corresponding optimal control policy is $u^*(z) = F^*z$, where

$$F^* := -(R+B^TX^*B)^{-1}B^TX^*A \in \mathcal{F} \tag{4}$$

is the unique optimal gain. Alternatively, the Q-function [5] is defined as

$$Q^*(z,u) := z^TQz + u^TRu + J^*(Az+Bu) = \begin{bmatrix} z \\ u \end{bmatrix}^T P^* \begin{bmatrix} z \\ u \end{bmatrix}, \tag{5}$$

where

$$P^* := \begin{bmatrix} Q + A^TX^*A & A^TX^*B \\ B^TX^*A & R + B^TX^*B \end{bmatrix}. \tag{6}$$

The optimal policy in terms of the Q-function is then given by

$$u^*(z) = F^*z = \arg\min_{u\in\mathbb{R}^m} Q^*(z,u).$$

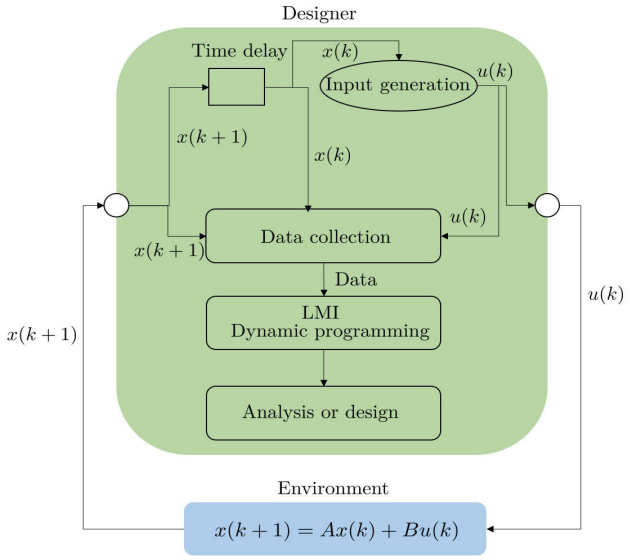

FIGURE 1. Overall diagram of the data-driven analysis and design schemes.

Before closing this section, some useful lemma is summarized.

Lemma 2: Given matrices U, V *of appropriate dimensions, the following holds for any* $\varepsilon > 0$:

$$-\varepsilon^{-1}U^TU - \varepsilon V^TV \preceq U^TV + V^TU \preceq \varepsilon^{-1}U^TU + \varepsilon V^TV.$$

Proof: The first inequality comes from $(\varepsilon^{-1/2}U + \varepsilon^{1/2}V)^T(\varepsilon^{-1/2}U + \varepsilon^{1/2}V) = \varepsilon^{-1}U^TU + U^TV + V^TU + \varepsilon V^TV \succeq 0$ and the reversed inequality is obtained from $(\varepsilon^{-1/2}U - \varepsilon^{1/2}V)^T(\varepsilon^{-1/2}U - \varepsilon^{1/2}V) = \varepsilon^{-1}U^TU - U^TV - V^TU + \varepsilon V^TV \succeq 0$. This completes the proof. □

The overall diagram of the data-driven analysis and design schemes considered in this paper is given in Figure 1. In general, the data-driven methods consist of the following steps:

1) Data collection: From the environment (system), we (designer or analyzer) receive the state $x(k+1)$, and then generates the input $u(k)$. Using the time delay unit, the complete transition set $(x(k), x(k+1), u(k))$ can be collected. Over a certain period of time, we can collect a series of the transition set, which is called the data collection process.
2) Analysis or design: From the data collection part, we can obtain a data set, which is then transmitted to the analysis or design part, where we can solve data-dependent LMIs or perform data-dependent dynamic programming methods based on the collected data to analyze the system or design a controller for the unknown system.

In this paper, we will address both the data collection and analysis or design parts with more emphasis on the data collection schemes. In Section IV, the data collection methods are introduced, the data-driven LMIs are discussed in Section IV, Section V, and the data-driven dynamic programming is addressed in Section VI. Finally, more

sophisticated data collection methods are discussed in Section VII with theoretical proofs of their effectiveness.

III. DATA COLLECTION

In this section, we introduce two data acquisition schemes. We note that these two data collection algorithms are conceptual, and introduced as an intermediate step. New exploration and data collection schemes, which are more practical, will be proposed in Section VII. In particular, Algorithm 1 will be called an on-policy data collection algorithm with exploring starts, $(S(F), H(F)) = \texttt{On}-\texttt{Collect}(F)$, where on-policy means that the generated data depends on a particular state-feedback gain F. Note that S_i in Algorithm 1 is nothing but a sample covariance matrix

$$S_i = \frac{1}{N}\sum_{k=1}^{N-1}\begin{bmatrix} x(k;F,e_i) \\ u(k)\end{bmatrix}\begin{bmatrix} x(k;F,e_i) \\ u(k)\end{bmatrix}^T.$$

The data generated by Algorithm 1 can be useful when we want to evaluate a state-feedback gain F, i.e., its stabilizability or the LQR performance. The exploring starts [1] imply that for sufficient exploration of the state-space, Algorithm 1 needs trajectories starting from different initial states $x(0) = e_i, i = 1, 2, \ldots, n$, where $(e_1, e_2, \ldots, e_n)$ is the standard basis that spans the state-space, $\mathbb{R}^n$.

The following lemma offers a useful property of the data matrices.

Lemma 3 (Data Matrix Transformation): $S(F)A_F^T = H(F)$ *holds.*

Proof: We have

$$\begin{aligned} S(F)A_F^T &= \frac{1}{nN}\sum_{i=1}^{n}\sum_{k=0}^{N-1}\begin{bmatrix} x(k;F,e_i) \\ u(k)\end{bmatrix}\begin{bmatrix} x(k;F,e_i) \\ u(k)\end{bmatrix}^T A_F^T \\ &= \frac{1}{nN}\sum_{i=1}^{n}\sum_{k=0}^{N-1}\begin{bmatrix} x(k;F,e_i) \\ u(k)\end{bmatrix}\begin{bmatrix} x(k+1;F,e_i) \\ u(k+1)\end{bmatrix}^T \\ &= H(F) \end{aligned}$$

□

Another method, Algorithm 2, is an off-policy data collection algorithm with exploration. Here, the off-policy indicates that the data generated by Algorithm 2 does not depend on a specific state-feedback gain, and it is particularly useful for design algorithms. Roughly speaking, the exploration means that it uses some exploration signals in control inputs to sufficiently explore the state-space so as to collect sufficient information on the model. Note that the data matrices, (S, H), in Algorithm 2 can be expressed as

$$\begin{aligned} S &:= \frac{1}{N}\sum_{k=0}^{N-1}\begin{bmatrix} x(k) \\ u(k)\end{bmatrix}\begin{bmatrix} x(k) \\ u(k)\end{bmatrix}^T \\ H &:= \frac{1}{N}\sum_{k=0}^{N-1}\begin{bmatrix} x(k) \\ u(k)\end{bmatrix}x(k+1)^T \end{aligned}$$

Throughout the paper, we call the data generated by the data collection algorithms is valid if the S-matrix ($S(F)$ or S)

Algorithm 1 On-Policy Data Collection $(S(F), H(F)) = \texttt{On}-\texttt{Collect}(F)$ With Exploring Starts

for $i \in \{1, 2, \ldots, n\}$ **do**
  Initialize $S_i = 0, H_i = 0$.
  Initialize $x(0) = e_i$.
  for $k \in \{0, 1, \ldots, N-1\}$ **do**
    Apply control input $u(k) = Fx(k)$
    Observe $x(k+1)$
    Update

$$\begin{aligned} S_i &\leftarrow \frac{k}{k+1}S_i + \frac{1}{k+1}\begin{bmatrix} x(k) \\ u(k)\end{bmatrix}\begin{bmatrix} x(k) \\ u(k)\end{bmatrix}^T \\ H_i &\leftarrow \frac{k}{k+1}H_i + \frac{1}{k+1}\begin{bmatrix} x(k) \\ u(k)\end{bmatrix}\begin{bmatrix} x(k+1) \\ u(k+1)\end{bmatrix}^T \end{aligned}$$

  end for
end for
Return $(S(F), H(F)) = \left(\frac{1}{n}\sum_{i=1}^{n} S_i, \frac{1}{n}\sum_{i=1}^{n} H_i\right)$

Algorithm 2 Off-Policy Data Collection $(S, H) = \texttt{Off}-\texttt{Collect}(z)$ With Exploration

Initialize $S_0 = 0, H_0 = 0$.
Initialize $x(0) = z$.
Initialize $\varepsilon > 0$.
for $k \in \{0, 1, \ldots\}$ **do**
  Apply control input $u(k)$ with some excitation input
  Observe $x(k+1)$
  Update

$$\begin{aligned} S_{k+1} &\leftarrow \frac{k}{k+1}S_k + \frac{1}{k+1}\begin{bmatrix} x(k) \\ u(k)\end{bmatrix}\begin{bmatrix} x(k) \\ u(k)\end{bmatrix}^T \\ H_{k+1} &\leftarrow \frac{k}{k+1}H_k + \frac{1}{k+1}\begin{bmatrix} x(k) \\ u(k)\end{bmatrix}x(k+1)^T \end{aligned}$$

  if $\lambda_{\min}(S_{k+1}) > \varepsilon$ **then**
    Stop and return $(S, H) = (S_{k+1}, H_{k+1})$
  end if
end for

is strictly positive definite. For completeness, the definition is formally stated below.

Definition 1 (Data Validity): The data (S, H) *and* $(S(F), H(F))$ *generated by Algorithm 1 and Algorithm 2, respectively, is said to be valid if* $S \succ 0$ *and* $S(F) \succ 0$, *respectively.*

The validity of the data ensures that all the proposed methods perform well, and completely solve the desired problems. Due to the exploring starts in Algorithm 1, we can prove that the data from Algorithm 1 is always valid for any $N > 0$.

Lemma 4: [Data validity of Algorithm 1] With a positive integer $N > 0$, $S(F) \succ 0$ *holds.*

Proof: We have

$$S(F) = \frac{1}{nN}\sum_{i=1}^{n}\sum_{k=0}^{N-1}\begin{bmatrix} x(k;F,e_i) \\ u(k)\end{bmatrix}\begin{bmatrix} x(k;F,e_i) \\ u(k)\end{bmatrix}^T$$

$$
\begin{aligned}
&= \frac{1}{nN}\sum_{i=1}^{n}\sum_{k=0}^{N-1}(A_F)^k e_i e_i^T (A_F^T)^k \\
&= \frac{1}{nN}\sum_{k=0}^{N-1}(A_F)^k (A_F^T)^k \\
&\succeq \frac{1}{nN} I \\
&\succ 0,
\end{aligned}
$$

which completes the proof. □

On the other hand, Algorithm 2 cannot theoretically guarantee the validity. Therefore, we adopt the so-called persistent excitation assumption for Algorithm 2, given below.

Assumption 2 (Persistent Excitation (PE)): There exists a positive integer $N > 0$ such that $S \succ 0$ from Algorithm 2.

We notice that it is typical to apply Assumption 2 in adaptive control and RL community [7], [9], [30]. Moreover, in the last section, more sophisticated exploration data collection algorithms will be developed, which theoretically guarantee the data validity with different scenarios.

Remark 2: Some remarks are in order.

1) *Suppose that one input-state trajectory of (1) is stacked into the matrices*

$$
\begin{aligned}
X &= \begin{bmatrix} x(0) & \cdots & x(N-1) \end{bmatrix}, \\
U &= \begin{bmatrix} u(0) & \cdots & u(N-1) \end{bmatrix}.
\end{aligned}
$$

Then, a standard condition for the PE in the literature [16], [31] corresponds to the condition that $\begin{bmatrix} X \\ U \end{bmatrix}$ has full row rank. The PE condition is equivalent to $\begin{bmatrix} X \\ U \end{bmatrix}\begin{bmatrix} X \\ U \end{bmatrix}^T \succ 0$, which is the data validity in Definition 1.

2) *There are hyper-parameters N and ε to be tuned in Algorithm 1 and Algorithm 2, respectively. In Algorithm 1, any $N > 0$ can generate valid $(S(F), H(G))$. Therefore, we can simply set $N = 1$. In Algorithm 2, any sufficiently small $\varepsilon > 0$ is a good choice. To avoid ill-conditioning problems, we recommend* 0.001 *or* 0.01.

Finally, the following lemma will be useful throughout the paper.

Lemma 5 (Data Matrix Transformation): The following identity holds:

$$
S\begin{bmatrix} A^T \\ B^T \end{bmatrix} = H
$$

Proof:

$$
\begin{aligned}
S\begin{bmatrix} A^T \\ B^T \end{bmatrix} &= \frac{1}{N}\sum_{k=0}^{N-1}\begin{bmatrix} x(k) \\ u(k) \end{bmatrix}(Ax(k)+Bu(k))^T \\
&= \frac{1}{N}\sum_{k=0}^{N-1}\begin{bmatrix} x(k) \\ u(k) \end{bmatrix}x(k+1)^T \\
&= H
\end{aligned}
$$

□

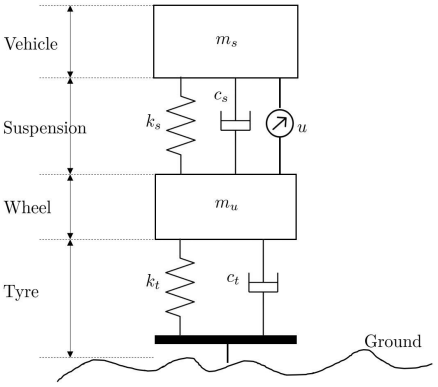

FIGURE 2. Example 1: Diagram of the quarter-vehicle suspension model.

Example 1: As a running example, we will consider the generalized quarter-vehicle suspension model [32] whose continuous-time linear model is given by $\dot{x}(t) = A_c x(t) + B_c u(t)$, $t \geq 0$ with

$$
A_c = \begin{bmatrix} 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \\ -\frac{k_s}{m_s} & 0 & -\frac{c_s}{m_s} & \frac{c_s}{m_s} \\ \frac{k_s}{m_u} & -\frac{k_t}{m_u} & \frac{c_s}{m_u} & -\frac{c_s+c_t}{m_u} \end{bmatrix},
$$

$$
B_c = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{m_s} \\ -\frac{1}{m_u} \end{bmatrix},
$$

where m_s is the sprung mass, which represents the vehicle chassis; m_u is the unsprung mass, which represents mass of the wheel assembly; c_s and k_s are damping and stiffness of the suspension system, respectively; k_t and c_t stand for compressibility and damping of the pneumatic tyre, respectively. The overall diagram of the system is given in Figure 2.

The model parameters are $m_s = 973kg$, $m_u = 114kg$, $k_s = 42720N/m$, $k_t = 101115N/m$, $c_s = 1095Ns/m$, and $c_t = 14.6Ns/m$. Using the Euler discretization with the sampling time $T = 0.01$ s, we can obtain the discrete-time LTI system (1) with

$$
A = \begin{bmatrix} 1 & 0 & 0.01 & -0.0100 \\ 0 & 1 & 0 & 0.0100 \\ -0.4391 & 0 & 0.9887 & 0.0113 \\ 3.7474 & -8.8697 & 0.0961 & 0.9027 \end{bmatrix}
$$

and

$$
B = 10^{-4}\begin{bmatrix} 0 & 0 & 0.1028 & -0.8772 \end{bmatrix}^T.
$$

We run Algorithm 1 with $N = 100$ and $\varepsilon = 0.1$, resulting in

$$
S = \begin{bmatrix} 1.0960 & 0.2110 & -3.6529 & -1.5332 & -0.0585 \\ 0.2110 & 0.2833 & -0.6616 & -1.7359 & -0.1209 \\ -3.6529 & -0.6616 & 18.8238 & -24.0709 & 0.5410 \\ -1.5332 & -1.7359 & -24.0709 & 186.8419 & -1.5513 \\ -0.0585 & -0.1209 & 0.5410 & -1.5513 & 1.0506 \end{bmatrix}
$$

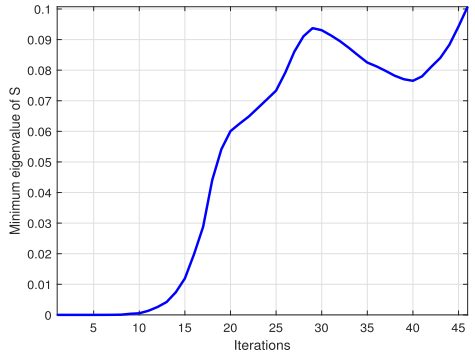

FIGURE 3. Example 1: Evolution of $\lambda_{\min}(S)$ for different iterations.

and

$$H = \begin{bmatrix} 1.0748 & 0.1957 & -4.1103 & 0.5009 \\ 0.2217 & 0.2659 & -0.7663 & -3.3523 \\ -3.2240 & -0.9023 & 19.9449 & -27.7407 \\ -3.6423 & 0.1325 & -21.0241 & 175.9953 \\ -0.0376 & -0.1364 & 0.5432 & -0.4956 \end{bmatrix}.$$

Note that the data (S, H) is valid because $S \succ 0$ (i.e., $\lambda_{\min}(S) > 0$). The evolution of $\lambda_{\min}(S)$ for different iterations is shown in Figure 3.

IV. DATA-DRIVEN LMIs FOR STABILIZATION

In this section, the main focus is on data-driven LMIs for stabilization, where the model (A, B) is unknown. The main breakthrough in this approach lies in augmenting the state and input into a single augmented state as in (3). Then, the model data (A, B) can be eliminated using the data matrices (S, H) or $(S(F), H(F))$ together with Lemma 3 and Lemma 5. We first consider a policy evaluation problem. In this setting, given a potentially unknown state-feedback gain F, we only have an access to the state-input trajectories. Under this situation, the problem is to determine whether or not the unknown feedback gain F stabilizes the system.

Proposition 1 (Stability evaluation): The system (1) under $u(k) = Fx(k)$ is stable if and only if there exist $P \in \mathbb{S}^{(n+m)}$ such that the following LMI holds:

$$H(F)^T P H(F) \prec S(F) P S(F), \quad P \succ 0$$

Proof: Lemma 1 tells us that the original system (1) is stabble if and only if the augmented system (3) is stable, or equivalently A_F is Schur. From the Lyapunov theory, A_F^T is stable if and only if there exist $P \in \mathbb{S}_{++}^{(n+m)}$ such that $A_F P A_F^T \prec P$. Replacing P with $S(F)PS(F)$ leads to the equivalent condition $A_F S(F) P S(F) A_F^T \prec S(F) P S(F)$. Using the relation in Lemma 3 yields the conclusion. □

Proposition 1 will be useful when we want to check if the unknown system with a given and fixed gain F is (asymptotically) stable. Its main feature is that it requires the on-policy data from Algorithm 1. Note however that it does not require the knowledge of F as well as (A, B). Next, a stabilizing state-feedback control design algorithm is proposed using LMIs and data from Algorithm 2.

Proposition 2 (Stabilization): The system (1) is stabilizable if there exist $G \in \mathbb{R}^{n\times n}$, $P \in \mathbb{S}^{n+m}$, and $X \in \mathbb{R}^{n\times m}$, such that the following LMI holds:

$$\begin{bmatrix} -SPS & * \\ \begin{bmatrix} G & X \end{bmatrix} & H^T P H - G - G^T \end{bmatrix} \prec 0 \tag{7}$$

If a solution, $(\bar{P}, \bar{G}, \bar{X})$, exists, then a stabilizing state-feedback gain is given by $F = \bar{X}^T (\bar{G}^T)^{-1}$, and $V(x) = x^T S \bar{P} S x$ is the corresponding Lyapunov function of A_F.

Proof: Suppose that (7) holds. We use the identity in Lemma 5, and replace SPS with $\hat{P}$ and to obtain

$$\begin{bmatrix} -\hat{P} & * \\ G \begin{bmatrix} I & G^{-1}X \end{bmatrix} & \begin{bmatrix} A^T \\ B^T \end{bmatrix}^T \hat{P} \begin{bmatrix} A^T \\ B^T \end{bmatrix} - G - G^T \end{bmatrix} \prec 0 \tag{8}$$

The first block diagonal matrix of (8) ensures $\hat{P} \succ 0$, and the second block diagonal matrix implies $G + G^T \succ 0$. This guarantees that G is nonsingular. With the change of variables, $G^{-1}X = F^T$, the last matrix inequality becomes

$$\begin{bmatrix} -\hat{P} & * \\ G \begin{bmatrix} I & F^T \end{bmatrix} & \begin{bmatrix} A^T \\ B^T \end{bmatrix}^T \hat{P} \begin{bmatrix} A^T \\ B^T \end{bmatrix} - G - G^T \end{bmatrix} \prec 0 \tag{9}$$

Clearly, (9) holds if and only if (8) holds from the bijective mapping $F^T = G^{-1}X$. Now, multiplying $\begin{bmatrix} I \\ \begin{bmatrix} I & F^T \end{bmatrix} \end{bmatrix}$ from the right, and its transpose from the left, and after simple algebraic manipulations, we have $A_F \hat{P} A_F^T \prec \hat{P}$, $\hat{P} \succ 0$. From the Lyapunov theory, the dual system A_F^T is Schur. Moreover, from a standard result of the linear system theory, we know that A_F is Schur if and only if the corresponding dual system A_F^T is Schur. Lemma 1 tells us that the original system (1) is stable if and only if the augmented system (3) is stable, or equivalently A_F is Schur. This completes the proof. □

Using the LMI condition in Proposition 2, a stabilizing state-feedback controller can be found only using the trajectories. Note that the data used in Proposition 2 is generated from the off-policy method Algorithm 1.

Example 2: Let us consider Example 1 again. Since (S, H) is valid, we can apply the LMI condition in Proposition 2 to obtain the stabilizing state-feedback control gain

$$F = 10^4 \begin{bmatrix} 3.6991 & 2.6712 & -0.8137 & 1.0457 \end{bmatrix}.$$

One can check $\rho(A+BF) = 0.9932 < 1$ so that the obtained gain stabilizes the system.

Remark 3: The proposed data-based LMI can be efficient alternatives in some cases. Consider the LMI condition in Proposition 2, where the number of variables is $n^2 + nm + (n+m)(n+m-1)/2$, and the size of LMIs is $2n+m$. On the other hand, for [16, Thm. 3], the number of variables is $n^2 T$, and the size of LMIs is n, where T is the length of the collected trajectory. If an LMI solver based on interior point methods is used, as for instance the LMI Control Toolbox [33], the complexity of the LMI optimization problem can be estimated as

being proportional to $O(N_d^3 N_s)$, where N_d is the total number of scalar decision variables and N_s the total row size of the LMIs. For SeDuMi [34] on the other hand, the complexity of the LMI optimization problem can be estimated as being proportional to $O(N_d^2 N_s^{2.5} + N_s^{3.5})$. SeDuMi is more efficient for problems with a large number of variables. Therefore, when T is large, the proposed method can be more efficient than [16, Thm. 3]. On the other hand, for [20, Thm. 14], the number of variables is $nm + n(n-1)/2 + 2$, and the size of LMIs is $3n+m$. However, the memory size to store the data is $O((2n+m)T)$, while the memory size is fixed to $O((n+m)(n+m-1)/2)$ for the proposed method. Moreover, the condition in [20, Thm. 14] requires additional structural information on the noise signals. A disadvantage is that Proposition 2 is only a sufficient condition, and hence, can introduce some conservatism.

Example 3: For a comparative analysis, we randomly generate 1000 systems, characterized by (A, B), with size $(n, m) = (10, 2)$ and elements uniformly distributed within $[-1, 1]$. Using Proposition 2, 799 systems are identified as stabilizable, while 990 systems are identified as stabilizable using [16, Thm. 3]. The results show the conservatism of Proposition 2. Still, the main benefit of Proposition 2 is its efficiency in terms of memory and computation.

V. DATA-DRIVEN LMIs FOR LQR DESIGN

Beyond the stabilization problem, the idea in the previous section can be also applied to LQR design problems. We first consider a policy evaluation problem again. Given a potentially unknown state-feedback gain F, suppose that we only have an access to the state-input trajectories. Under this situation, the problem is to determine the LQR performance of the unknown F.

Proposition 3 (Performance Evaluation): Consider the optimization problem

$$\begin{aligned} \min_{P\in\mathbb{S}^{n+m}} \quad & Tr\left(\Lambda S(F)PS(F)\right) \\ \text{subject to} \quad & H(F)^T PH(F) + I \preceq S(F)PS(F) \end{aligned} \tag{10}$$

and $\bar{P} \in \mathbb{S}^{n+m}$ is the corresponding optimal point. Then, the optimal objective function value (10) is the cost corresponding to F, i.e., $Tr\left(\Lambda S(F)\bar{P}S(F)\right) = J(F)$.

Proof: The optimal solution $\bar{P}$ satisfies $H(F)^T\bar{P}H(F) + I \preceq S(F)\bar{P}S(F), \bar{P} \succ 0$. Using Lemma 3 and letting $\tilde{P} = S(F)\bar{P}S(F)$, it follows that

$$A_F\tilde{P}A_F^T + I \preceq \tilde{P}, \quad \tilde{P} \succ 0. \tag{11}$$

Here, note that $\tilde{P} \succ 0$ holds because the LMI constraint in (10) implies $S(F)\tilde{P}S(F) \succ 0$. Since the above inequality is a Lyapunov inequality, A_F is Schur. Therefore, there exists $\hat{P} \in \mathbb{S}_{++}$ such that $A_F\hat{P}A_F^T + I = \hat{P}$, where $\hat{P} := \sum_{k=0}^{\infty} A_F^k (A_F^T)^k$. Replacing $\hat{P}$ with $S(F)MS(F)$, where $M = S(F)^{-1}\hat{P}S(F)^{-1}$, we can see that M satisfies $H(F)^T MH(F) + I = S(F)MS(F)$. This implies that M is a feasible point for (10). Therefore, $Tr\left(\Lambda S(F)\bar{P}S(F)\right) \le Tr(\Lambda S(F)MS(F)) = Tr(\Lambda\hat{P})$. On the other hand, repeatedly applying the inequality (11) yields $\sum_{k=0}^{\infty} A_{\tilde{K}}^k (A_F^T)^k \preceq \tilde{P} = S(F)\bar{P}S(F)$, by which we have $Tr(\Lambda S(F)\bar{P}S(F)) \ge Tr(\Lambda\hat{P})$, implying $Tr(\Lambda S(F)\bar{P}S(F)) = Tr(\Lambda\hat{P})$. Then, we can conclude $Tr\left(\Lambda S(F)\bar{P}S(F)\right) = Tr(\Lambda\hat{P}) = Tr\left(\sum_{k=0}^{\infty}(A_F^T)^k\Lambda A_F^k\right) = \sum_{i=1}^{n}\sum_{k=0}^{\infty} e_i^T(A_F^T)^k\Lambda A_F^k e_i = J(F)$. This completes the proof. □

Next, the LQR design problem is addressed using a data-driven LMI. The following LMI condition allows us to design an LQR control of unknown system.

Proposition 4 (LQR Design): Consider the optimization problem

$$\begin{aligned} \min_{P\in\mathbb{S}^{n+m},\, G\in\mathbb{R}^{n\times n},\, X\in\mathbb{R}^{n\times m}} \quad & Tr\,(\Lambda SPS) \\ \text{subject to} \quad & \begin{bmatrix} -SPS + I & * \\ \begin{bmatrix} G & X \end{bmatrix} & H^T PH - G - G^T \end{bmatrix} \prec 0 \end{aligned} \tag{12}$$

and $\bar{G} \in \mathbb{R}^{n\times n}$, $\bar{P} \in \mathbb{S}^{n+m}$, and, $\bar{Y} \in \mathbb{R}^{n\times m}$ are the corresponding optimal points. Then, the optimal objective function value upper bounds the optimal cost, $J(F^*)$, and the corresponding state-feedback gain is given by $\bar{F} = \bar{X}^T(\bar{G}^T)^{-1}$,

Proof: The LMI constraint is identical to the stabilization case except for I in the first block diagonal position. Therefore, we can follow the same procedure to arrive that the conclusion that the optimal value of (12) upper bounds that of the following optimization:

$$\begin{aligned} \min_{P\in\mathbb{S}^{n+m},\, F\in\mathbb{R}^{m\times n}} \quad & Tr\,(\Lambda P) \\ \text{subject to} \quad & A_F PA_F^T + I \preceq P. \end{aligned} \tag{13}$$

This is because the inequality in (12) is sufficient for the inequality in (13). Define $(\hat{S}, \hat{F})$ is an optimal solution of (13). Then, $Tr(\Lambda S\bar{P}S) \ge Tr(\Lambda\hat{P})$. Next, applying the inequality in (13) recursively leads to $\sum_{k=0}^{\infty} A_{\hat{F}}^k (A_{\hat{F}}^T)^k \preceq \hat{P}$. Multiplying with Λ and taking the trace on the last inequality, one gets

$$\begin{aligned} Tr(\Lambda\hat{P}) &\ge Tr\left(\Lambda\sum_{k=0}^{\infty} A_{\hat{F}}^k (A_{\hat{F}}^T)^k\right) \\ &= \sum_{i=1}^{n}\sum_{k=0}^{\infty} x(k;\hat{F}, e_i)^T\Lambda x(k;\hat{F}, e_i) \\ &= J(\hat{F}). \end{aligned} \tag{14}$$

Therefore, the optimization in (12) finds an optimal solution $(\hat{S}, \hat{F})$. Finally, combining (14) with $Tr(\Lambda S\bar{P}S) \ge Tr(\Lambda\hat{P})$ leads to the desired conclusion. □

Theorem 4 allows us to design a controller with a guaranteed upper bound on the LQR performance.

Example 4: Let us consider Example 1 again, and consider the LQR problem with $\Lambda = I$. Solving the LMI condition in Theorem 4 leads to the state-feedback gain

$$F^* = 10^3 \begin{bmatrix} 0.9934 & -1.9672 & -0.1342 & 0.4862 \end{bmatrix}$$

with the bound $J(F^*) \le 18.8316$.

VI. DATA-DRIVEN DYNAMIC PROGRAMMING

Although the data-driven LMIs in the previous sections are efficient, it is still meaningful to briefly discuss and summarize DP methods [6], which does not depend on LMI solvers. In particular, DP approaches can be alternatives to LMI methods when the LMI solvers are not available or when one wants to find a solution without additional steps. Moreover, LMI problems are more universal, and finding solutions of LMIs is a more complicated problem compared to DP problems. In general, DPs are in general more efficient than LMIs when dealing with large-scale problems. Finally, the LMI conditions for the design problems are only sufficient, and hence, are conservative compared to the DP approaches. Now, the data-driven DPs are summarized in Algorithm 3 and Algorithm 4.

Algorithm 3 Data-Driven Policy Iteration

1: Initialize $F_0 = 0$.
2: **for** $k \in \{0, 1, \ldots\}$ **do**
3: Collect data $(S(F_k), H(F_k)) = \texttt{On} - \texttt{Collect}(F_k)$
4: Solve for P_{k+1} the linear equation

$$H(F_k)^T P_{k+1} H(F_k) + S(F_k) \Lambda S(F_k) = S(F_k) P_{k+1} S(F_k)$$

5: Update $F_{k+1} = -P_{k+1,22}^{-1} P_{k+1,12}^T$
6: **if** $\|P_k - P_{k+1}\| \leq \varepsilon$ **then**
7: Stop and return P_{k+1} and $F_{k+1} = -P_{k+1,22}^{-1} P_{k+1,12}^T$
8: **end if**
9: **end for**

Algorithm 3 summarizes a policy iteration algorithm proposed in [14] for completeness. Its convergence was also proved in [14].

Proposition 5: [Convergence of Algorithm 3, [14]] The iteration P_k in Algorithm 3 converges to P^* defined in (6).

The main feature of Algorithm 3 is that it uses on-policy data generated by Algorithm 1. Therefore, it needs to collect new data at every iterations, and each data collection should apply the exploring starts scheme. The newly proposed value iteration algorithm presented in Algorithm 4 suggests an off-policy algorithm in the sense that the policy used to generate the data is independent of the policy we want to learn or the intermediate policies while learning. Therefore, it collects data once at the beginning. Moreover, it does not need to stick to the exploring starts scheme because the exploratory inputs can be used during the data collection. In this sense, the new Algorithm 4 is more sample efficient than Algorithm 3. Multiplying both sides of the linear matrix equation in Algorithm 4 by S, it is reduced to

$$P_{k+1} = \Lambda + S^{-1} H (P_{k,11} - P_{k,12} P_{k,22}^{-1} P_{k,12}^T) H^T S^{-1}$$

which can be interpreted as a model-based value iteration because $H^T S^{-1} = \begin{bmatrix} A & B \end{bmatrix}$ from Lemma 5. In both algorithms, the hyper-parameter ε can be any sufficient small positive numbers to obtain a suboptimal solution whose error

Algorithm 4 Data-Driven Value Iteration

1: Initialize $P_0 = 0$.
2: Given fixed initial state $x(0) = z$, collect data $(S, H) = \texttt{Off} - \texttt{Collect}(z)$
3: **for** $k \in \{0, 1, \ldots\}$ **do**
4: Solve for P_{k+1} the linear matrix equation

$$S P_{k+1} S = S \Lambda S + H (P_{k,11} - P_{k,12} P_{k,22}^{-1} P_{k,12}^T) H^T$$

5: **if** $\|P_k - P_{k+1}\| \leq \varepsilon$ **then**
6: Stop and return P_{k+1} and $F_{k+1} = -P_{k+1,22}^{-1} P_{k+1,12}^T$
7: **end if**
8: **end for**

is tolerable. Usually tolerable error levels depend on underlying applications, and from our experiences, $\varepsilon = 10^{-6}$ is enough in general. Lastly, we establish the convergence of Algorithm 4.

Proposition 6: [Convergence of Algorithm 4] The iteration P_k in Algorithm 4 converges to P^*.

Proof: We only need to prove that Algorithm 4 is equivalent to the Q-value iteration, which is known to converge to the optimal P^* [6]. Applying Lemma 5 and multiplying both sides of the P-update equation in Algorithm 4 by S^{-1}, we obtain

$$P_{k+1} = \Lambda + \begin{bmatrix} A & B \\ -P_{k,22}^{-1} P_{k,12} A & -P_{k,22}^{-1} P_{k,12} B \end{bmatrix}^T \times P_k \begin{bmatrix} A & B \\ -P_{k,22}^{-1} P_{k,12} A & -P_{k,22}^{-1} P_{k,12} B \end{bmatrix}$$

Multiplying both sides by $\begin{bmatrix} x \\ u \end{bmatrix}$ from the right and its transpose from the left, we have $Q_{k+1}(x, u) = \begin{bmatrix} x \\ u \end{bmatrix}^T \Lambda \begin{bmatrix} x \\ u \end{bmatrix} + \min_{v \in \mathbb{R}^m} Q_k(x, v)$ with $Q_k(x, u) = \begin{bmatrix} x \\ u \end{bmatrix}^T P_k \begin{bmatrix} x \\ u \end{bmatrix}$. It is equivalent to the Q-value iteration [5], which is known to converge to P^*, where $Q^*(x, u) = \begin{bmatrix} x \\ u \end{bmatrix}^T P^* \begin{bmatrix} x \\ u \end{bmatrix}$. This completes the proof. □

VII. NEW EXPLORATION SCHEMES

For the on-policy data collection, Algorithm 1, the exploring starts always guarantee $S(F) \succ 0$. However, collecting the trajectories with different initial points which span $\mathbb{R}^n$ may not be tractable in practice. The off-policy data collection, Algorithm 2, is relevantly more promising in this respect, because it can use the exploratory inputs while generating the trajectories, and can be used in the case that the initial state is given and fixed. However, it needs PE assumption. We can apply an arbitrary input, $u(k)$, and expect that $S \succ 0$ eventually under PE assumption. A standard exploration strategy is to inject the i.i.d. Gaussian noises, $u(k) \sim \mathcal{N}(0, U)$, where $U \in \mathbb{S}_{++}^m$ is the covariance matrix.

If trajectories starting from the arbitrary but fixed $x(0) = z$ can be collected as many as possible, then we can develop a new version of the off-policy exploration strategy given in Algorithm 5, which offers theoretical guarantees of the data validity under a mild assumption, i.e., the controllability. Note that the fixed initial condition, $x(0) = z$, is not restrictive in practical applicabilities because only a small portion of state-space is accessible for initial states in practice.

Algorithm 5 Off-Policy Data Collection $(S, H) = \texttt{Off-Collect2}(z)$ With Restarting

Initialize $S_0 = 0$, $H_0 = 0$, and arbitrary $z \in \mathbb{R}^n$.
for $i \in \{1, 2, \ldots, N\}$ **do**
  Initialize $x(0; i) = z$.
  Initialize $\tilde{S}_{0;i} = 0$, $\tilde{H}_{0;i} = 0$.
  for $k \in \{0, 1, \ldots, n-1\}$ **do**
    Apply control input $u(k; i) = \zeta(k; i)$, $\zeta(k; i) \sim \mathcal{N}(0, U)$
    Observe $x(k+1; i)$
    Update

$$\tilde{S}_{k+1;i} \leftarrow \tilde{S}_{k;i} + \begin{bmatrix} x(k;i) \\ u(k;i) \end{bmatrix} \begin{bmatrix} x(k;i) \\ u(k;i) \end{bmatrix}^T$$

$$\tilde{H}_{k+1;i} \leftarrow \tilde{H}_{k;i} + \begin{bmatrix} x(k;i) \\ u(k;i) \end{bmatrix} x(k+1;i)^T$$

  end for
  Update

$$S_{i+1} \leftarrow \frac{i}{i+1} S_i + \frac{1}{i+1} \tilde{S}_{n;i}$$

$$H_{i+1} \leftarrow \frac{i}{i+1} H_i + \frac{1}{i+1} \tilde{H}_{n;i}$$

end for
Return $(S, H) = (S_N, H_N)$

In Algorithm 5, N trajectories are collected and then averaged, i.e., $S_N = \frac{1}{N}\sum_{i=1}^{N} \tilde{S}_{n;i}$, $H_N = \frac{1}{N}\sum_{i=1}^{N} \tilde{H}_{n;i}$. Each trajectory starts from $x(0) = z$ which is fixed. We can readily prove that the data matrices from Algorithm 5 also satisfies the data transformation property in Lemma 5. We can also prove that if (A, B) is controllable, then the data collection strategy guarantees that S_N converges to a strictly positive definite matrix with probability one as $N \to \infty$.

Theorem 1: Suppose that (A, B) is controllable, and consider Algorithm 5, whose output is

$$S_N = \frac{1}{N}\sum_{i=1}^{N} \left(\sum_{k=0}^{n} \begin{bmatrix} x(k;i) \\ u(k;i) \end{bmatrix} \begin{bmatrix} x(k;i) \\ u(k;i) \end{bmatrix}^T \right)$$

where $x(k; i)$ and $u(k; i)$ stand for the state and input at time k at the ith outer iteration. Then, we have

$$\mathbb{P}\left[\lim_{N\to\infty} S_N \succ 0 \right] = 1$$

Proof: Define

$$\mathcal{O}_k := \begin{bmatrix} B & AB & \cdots & A^{k-1}B \end{bmatrix}$$

$$\mathcal{U}_k := \operatorname{diag}(\underbrace{U, \ldots, U}_{\text{k}-times}) \tag{15}$$

and

$$u_{k;i} := \begin{bmatrix} \zeta(k-1;i) \\ \vdots \\ \zeta(1;i) \\ \zeta(0;i) \end{bmatrix} \tag{16}$$

Then, $x(k; i)$ is expressed as $x(k; i) = A^k z + \mathcal{O}_k u_{k;i}$, and thus

$$x(k;i)x(k;i)^T = A^k z^T z (A^T)^k + 2A^k z u_{k;i}^T \mathcal{O}_k^T + \mathcal{O}_k u_{k;i} u_{k;i}^T \mathcal{O}_k^T.$$

Taking the expectation leads to

$$\mathbb{E}[x(k;i)x(k;i)^T] = A^k z z^T (A^T)^k + \mathcal{O}_k \mathcal{U}_k \mathcal{O}_k^T$$

At $k = n$, $\mathcal{O}_n$ is the controllability matrix, and it is full row rank due to the controllability in Assumption 1. Since $\mathcal{U}_k \succ 0$, one concludes $\mathbb{E}[x(n)x(n)^T] \succ 0$. Since $u(k; i)$ is i.i.d. and the initial state is reset periodically after n steps, S_N is written as

$$S_N = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=0}^{n} \begin{bmatrix} x(k;i) \\ u(k;i) \end{bmatrix} \begin{bmatrix} x(k;i) \\ u(k;i) \end{bmatrix}^T = \frac{1}{N}\sum_{i=1}^{N} M_i$$

where

$$M_i = \sum_{k=0}^{n} \begin{bmatrix} x(k;i) \\ u(k;i) \end{bmatrix} \begin{bmatrix} x(k;i) \\ u(k;i) \end{bmatrix}^T$$

is an i.i.d. random variables with mean

$$\mathbb{E}[M_i] = M := \sum_{k=0}^{n} \begin{bmatrix} A^k z z^T (A^T)^k + \mathcal{O}_k \mathcal{U}_k \mathcal{O}_k^T & 0 \\ 0 & U \end{bmatrix} \succ 0$$

By the strong law of large numbers, we get

$$\mathbb{P}\left[\lim_{N\to\infty} S_N = M \right] = 1,$$

which leads to the desired conclusion. □

Algorithm 5 provides a data collection scheme with theoretical guarantees of the validity of the data. It is useful especially when the exploring starts scheme (starting with arbitrary initial states) is not available. However, it still requires the ability to generate N trajectories from the given initial state z. In practice, if only a single trajectory starting from a fixed z is available, we can develop another data acquisition method given in Algorithm 6. The benefit comes from some cost to pay. In particular, it initially needs a stabilizing state-feedback gain K or at least, the system matrix, A, itself needs to be stable. In such case, we can approximately mimic the restarting strategy in Algorithm 5 using the stability of the closed-loop system matrix, $A + BK$. Algorithm 6 will be called the off-policy data collection with periodic excitation. The main feature of Algorithm 6 lies in that the process can be interpreted as an alternation of the two phases: the first phase is a settling down period, where the state tends to vanish without the excitation signals in the input $u(k)$. This phase stops when the current state $x(k)$ is sufficiently small in the sense that $\|x(k)\| \leq \varepsilon$ for a sufficiently small $\varepsilon > 0$. The

Algorithm 6 Off-Policy Data Collection $(S, H) = \texttt{Off-Collect2}(z)$ With Periodic Excitation

Initialize $S_0 = 0, H_0 = 0$.
Initialize state $x(0) = z$.
Initialize $\varepsilon > 0$
for $i \in \{1, 2, \ldots, N\}$ **do**
  Initialize time $k = 0$
  while $\|x(k)\| > \varepsilon$ **do**
    Apply control input $\tilde{u}(k) = Kx(k)$
    $k \leftarrow k + 1$
  end while
  Initialize $\tilde{S}_0 = 0, \tilde{H}_0 = 0$.
  Initialize time $k = 0$ and $x(0; i) := x(0)$
  for $k \in \{0, 1, \ldots, n-1\}$ **do**
    Apply control input $u(k; i) = Kx(k; i) + \zeta(k; i), \zeta(k; i) \sim \mathcal{N}(0, U)$
    Observe $x(k+1; i) := x(k+1)$
    Update

$$\tilde{S}_{k+1} \leftarrow \tilde{S}_k + \begin{bmatrix} x(k;i) \\ u(k;i) \end{bmatrix} \begin{bmatrix} x(k;i) \\ u(k;i) \end{bmatrix}^T$$
$$\tilde{H}_{k+1} \leftarrow \tilde{H}_k + \begin{bmatrix} x(k;i) \\ u(k;i) \end{bmatrix} x(k+1;i)^T$$

  end for
  Update

$$S_{i+1} \leftarrow \frac{i}{i+1} S_i + \frac{1}{i+1} \tilde{S}_n$$
$$H_{i+1} \leftarrow \frac{i}{i+1} H_i + \frac{1}{i+1} \tilde{H}_n$$

end for
Return $(S, H) = (S_N, H_N)$

second phase is an excitation or exploration period, where the state is excited by injecting Gaussian noises in the input. As in Theorem 1, we can prove that Algorithm 6 theoretically ensures the validity of the data output provided that (A, B) is controllable.

Theorem 2: Suppose that (A, B) is controllable, and consider Algorithm 5, whose output is

$$S_N = \frac{1}{N} \sum_{i=1}^{N} \left(\sum_{k=0}^{n} \begin{bmatrix} x(k;i) \\ u(k;i) \end{bmatrix} \begin{bmatrix} x(k;i) \\ u(k;i) \end{bmatrix}^T \right)$$

where $x(k; i)$ and $u(k; i)$ stand for the state and input, respectively, at time k at the ith outer iteration. Then, there exists a sufficient small $\varepsilon > 0$ such that

$$\mathbb{P}\left[\lim_{N \to \infty} S_N \succ 0 \right] = 1$$

Proof: Define

$$\mathcal{O}_k := \begin{bmatrix} B & (A+BK)B & \cdots & (A+BK)^{k-1}B \end{bmatrix}.$$

Then, the state at time k is $x(k; i) = (A+BK)^k z_i + \mathcal{O}_k u_{k;i}$, where z_i is the initial state, $x(0; i) = z_i$ at the ith period such that $\|z_i\| \le \varepsilon$, and $u_{k;i}$ is defined in (16). Then, one gets

$$\begin{aligned} x(k;i)x(k;i)^T &= (A+BK)^k z_i z_i^T ((A+BK)^T)^k \\ &\quad + 2(A+BK)^k z_i u_{k;i}^T \mathcal{O}_k^T \\ &\quad + \mathcal{O}_k u_{k;i} u_{k;i}^T \mathcal{O}_k^T \end{aligned} \tag{17}$$

which is lower bounded by

$$\begin{aligned} x(k;i)x(k;i)^T &\succeq (A+BK)^k z_i z_i^T ((A+BK)^T)^k \\ &\quad - (A+BK)^k z_i z_i^T ((A+BK)^T)^k \frac{1}{\varepsilon} \\ &\quad - \varepsilon \mathcal{O}_k u_{k;i} u_{k;i}^T \mathcal{O}_k^T + \mathcal{O}_k u_{k;i} u_{k;i}^T \mathcal{O}_k^T \end{aligned}$$

where Lemma 2 was applied to (17). Again, the last bound is further bounded from below as

$$\begin{aligned} &x(k;i)x(k;i)^T \\ &\succeq (A+BK)^k z_i z_i^T ((A+BK)^T)^k \\ &\quad - (A+BK)^k z_i z_i^T ((A+BK)^T)^k \frac{1}{\varepsilon} \\ &\quad - \varepsilon \mathcal{O}_n u_{k;i} u_{k;i}^T O_k^T + \mathcal{O}_k u_{k;i} u_{k;i}^T \mathcal{O}_k^T \\ &\succeq (A+BK)^k z_i z_i^T ((A+BK)^T)^k \\ &\quad - I \lambda_{\max}((A+BK)^k z_i z_i^T ((A+BK)^T)^k) \frac{1}{\varepsilon} \\ &\quad + (1-\varepsilon) \mathcal{O}_k u_{k;i} u_{k;i}^T \mathcal{O}_k^T \\ &\succeq (A+BK)^k z_i z_i^T ((A+BK)^T)^k \\ &\quad - I \|(A+BK)^k\|^2 \|z_i\|^2 \frac{1}{\varepsilon} \\ &\quad + (1-\varepsilon) \mathcal{O}_k u_{k;i} u_{k;i}^T \mathcal{O}_k^T \\ &\succeq (A+BK)^k z_i z_i^T ((A+BK)^T)^k \\ &\quad - \varepsilon I \|(A+BK)^k\|^2 + (1-\varepsilon) \mathcal{O}_k u_{k;i} u_{k;i}^T \mathcal{O}_k^T \end{aligned}$$

where $\lambda_{\max}(\cdot)$ denotes the maximum eigenvalue of a symmetric matrix, and the last inequality uses the fact that $\|z_i\| \le \varepsilon$.

On the other hand, noting $x(k; i)u(k; i)^T = (A+BK)^k z_i u(k)^T + \mathcal{O}_k u_k u(k)^T$, we have

$$\begin{aligned} &\begin{bmatrix} 0 & x(k;i)u(k;i)^T \\ u(k;i)x(k;i)^T & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & (A+BK)^k z_i u(k;i)^T \\ u(k;i) z_i^T ((A+BK)^T)^k & 0 \end{bmatrix} \\ &\quad + \begin{bmatrix} 0 & \mathcal{O}_k u_k u(k;i)^T \\ u(k;i) u_k^T \mathcal{O}_k^T & 0 \end{bmatrix} \end{aligned}$$

where the first term on the right-hand side is bounded as

$$\begin{aligned} &\begin{bmatrix} 0 & (A+BK)^k z_i u(k;i)^T \\ u(k;i) z_i^T ((A+BK)^T)^k & 0 \end{bmatrix} \\ &= \begin{bmatrix} (A+BK)^k z_i \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ u(k;i) \end{bmatrix}^T \\ &\quad + \begin{bmatrix} 0 \\ u(k;i) \end{bmatrix} \begin{bmatrix} (A+BK)^k z_i \\ 0 \end{bmatrix}^T \\ &\succeq -\varepsilon \begin{bmatrix} 0 \\ u(k;i) \end{bmatrix} \begin{bmatrix} 0 \\ u(k;i) \end{bmatrix}^T \end{aligned}$$

$$
\begin{aligned}
&-\frac{1}{\varepsilon}\begin{bmatrix}(A+BK)^k z_i\\ 0\end{bmatrix}\begin{bmatrix}(A+BK)^k z_i\\ 0\end{bmatrix}^T \qquad (18)\\
&\succeq -\begin{bmatrix}\varepsilon^{-1}\lambda_{\max}((A+BK)^k z_i z_i^T((A+BK)^T)^k) & 0\\ 0 & \varepsilon u(k;i)u(k;i)^T\end{bmatrix}\\
&= -\begin{bmatrix}\varepsilon^{-1}\|(A+BK)^k z_i\|^2 I & 0\\ 0 & \varepsilon u(k;i)u(k;i)^T\end{bmatrix}\\
&\succeq -\begin{bmatrix}\varepsilon\|(A+BK)^k\|^2 I & 0\\ 0 & \varepsilon u(k;i)u(k;i)^T\end{bmatrix}
\end{aligned}
$$

where (18) is due to Lemma 2 and the last inequality is due to $\|z_i\| \le \varepsilon$. Combining the two lower bounds, we have

$$
\begin{aligned}
&\begin{bmatrix}x(k;i)x(k;i)^T & x(k;i)u(k;i)^T\\ u(k;i)x(k;i)^T & u(k;i)u(k;i)^T\end{bmatrix}\\
&\succeq \begin{bmatrix}(A+BK)^k z_i z_i^T((A+BK)^T)^n & 0\\ -2\varepsilon I\|(A+BK)^k\|^2 & 0\\ 0 & 0\end{bmatrix}\\
&+ \underbrace{\begin{bmatrix}(1-\varepsilon)\mathcal{O}_k u_{k;i}u_{k;i}^T\mathcal{O}_k^T & \mathcal{O}_k u_{k;i}u(k;i)^T\\ u(k;i)u_{k;i}^T\mathcal{O}_k^T & (1-\varepsilon)u(k;i)u(k;i)^T\end{bmatrix}}_{=:M_i}
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
S_N &= \frac{1}{N}\sum_{i=1}^{N}\left(\sum_{k=0}^{n}\begin{bmatrix}x(k;i)\\ u(k;i)\end{bmatrix}\begin{bmatrix}x(k;i)\\ u(k;i)\end{bmatrix}^T\right)\\
&\succeq \frac{1}{N}\sum_{i=1}^{N}\begin{bmatrix}x(n;i)\\ u(n;i)\end{bmatrix}\begin{bmatrix}x(n;i)\\ u(n;i)\end{bmatrix}^T\\
&\succeq \begin{bmatrix}-2\varepsilon I\|(A+BK)^k\|^2 & 0\\ 0 & 0\end{bmatrix} + \frac{1}{N}\sum_{i=1}^{N}M_i
\end{aligned}
$$

Since $(M_1, M_2, \ldots, M_N)$ are i.i.d. random variables with mean

$$
\mathbb{E}[M_i] = \begin{bmatrix}(1-\varepsilon)\mathcal{O}_n\mathcal{U}_n\mathcal{O}_n^T & 0\\ 0 & (1-\varepsilon)U\end{bmatrix}
$$

where $\mathcal{U}_k$ is defined in (15). From the strong law of large numbers, with $\varepsilon \in (0, 1)$, we have

$$
\mathbb{P}\left[\lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N}M_i > 0\right] = 1.
$$

Therefore, for a sufficiently small $\varepsilon \in (0, 1)$, S_N converges to a positive definite matrix with probability one. □

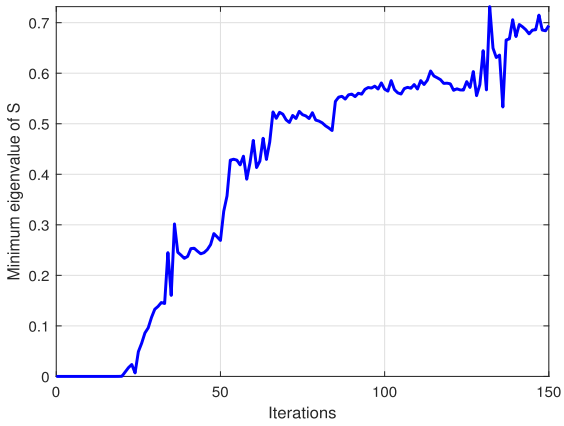

FIGURE 4. Example 5: Evolution of $\lambda_{\min}(S)$ for different iterations and for Algorithm 2.

We close this section by briefly comparing [10], [11] with the proposed exploration methods. Compared to [10] and [11], the exploration schemes in this paper have unique aspects summarized as follows: [10], [11] consider the stochastic system

$$
x(k+1) = Ax(k) + Bu(k) + w(k), \quad x(0) = z \in \mathbb{R}^n
$$

with an additional stochastic noise $w(k) \sim \mathcal{N}(0, \sigma^2 I)$, where $\sigma > 0$ is a standard deviation. Although the above stochastic system is more general than (1), $w(k)$ plays the role of effective state explorations, and hence, facilitate the learning process. In particular, if we do not have $w(k)$, then acquiring full information on the system via the input exploration is more challenging, and successful state explorations depend on the controllability of (A, B). To the authors' knowledge, Algorithm 6 provides the first successful exploration schemes that can learn the system information without the noise $w(k)$.

Example 5: To demonstrate the proposed excitation methods, we randomly generate an LTI system with $n = 30$ and $m = 5$ such that its elements are uniformly distributed in $[-1, 1]$ and (A, B) is controllable. Applying Algorithm 2 without the termination condition, the evolution of $\lambda_{\min}(S)$ for different iterations is shown in Figure 4. As can be seen, $\lambda_{\min}(S)$ increases, and S eventually becomes a positive definite matrix. As for Algorithm 5, the evolution of $\lambda_{\min}(S)$ is shown in Figure 5, where one can also observe that $\lambda_{\min}(S)$ increases as well, and S eventually becomes a positive definite matrix. The graph in Figure 5 is stair case because S is periodically updated with period n in Algorithm 5, while within each period, the iteration number increases n times.

Similar results can be obtained for Algorithm 6 as shown in Figure 6, whose graph is similar to that in Figure 5. The main difference between Algorithm 6 and Algorithm 5 lies in the fact that Algorithm 5 assumes restarting of the system trajectory, which is not practical, while Algorithm 6 does not assume such restarting of the system trajectory. Therefore, this example demonstrates the effectiveness of the proposed data collection schemes in Algorithm 5 and Algorithm 6. For

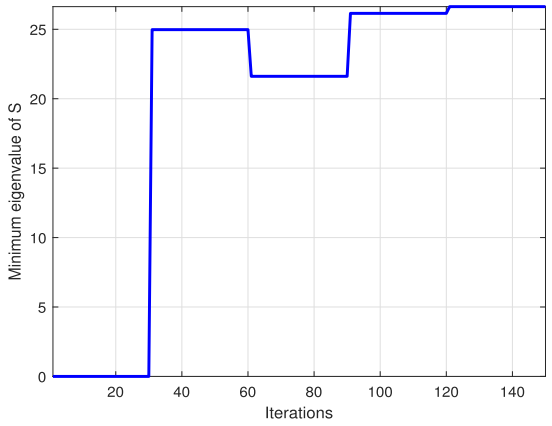

FIGURE 5. Example 5: Evolution of $\lambda_{\min}(S)$ for different iterations and for Algorithm 5.

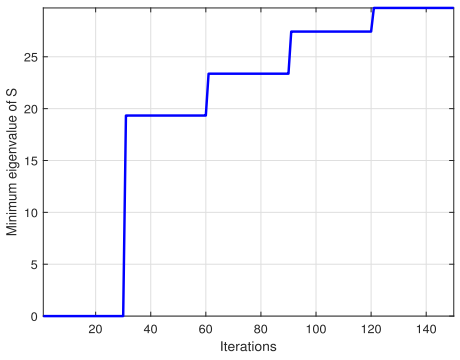

FIGURE 6. Example 5: Evolution of $\lambda_{\min}(S)$ for different iterations and for Algorithm 6.

the native data collection scheme Algorithm 2, the corresponding effectiveness has not been established in this paper (or elsewhere in the literature to the authors' knowledge), while for Algorithm 5 and Algorithm 6, their probabilistic effectiveness has been proved in Theorem 1 and Theorem 2.

VIII. CONCLUSION

In this paper, we have studied data-driven control design strategies based on LMIs and DP. Moreover, exploration and data collection schemes were investigated. The exploration schemes have been proved to asymptotically guarantee solvability of the data-drive design problems from the trajectories. New contributions of the proposed methods compared to existing works have been discussed. In particular, the proposed methods have unique features that can complement existing approaches under different scenarios. In this respect, we regard the proposed methods as useful complements rather than replacement of existing methods.

REFERENCES

[1] R. S. Sutton and A. G. Barto, *Reinforcement learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998.
[2] V. Mnih, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
[3] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*.
[4] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in Systems and Control Theory*. Philadelphia, PA, USA: SIAM, 1994.
[5] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, vol. 1, 3rd ed. Nashua, MA, USA: Athena Scientific, 2005.
[6] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Belmont, MA, USA: Athena Scientific, 1996.
[7] S. J. Bradtke, B. E. Ydstie, and A. G. Barto, "Adaptive linear quadratic control using policy iteration," in *Proc. IEEE Amer. Control Conf.*, vol. 3, Jul. 1994, pp. 3475–3479.
[8] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, 1992.
[9] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits Syst. Mag.*, vol. 9, no. 3, pp. 32–50, Aug. 2009.
[10] S. Dean, H. Mania, N. Matni, B. Recht, and S. Tu, "On the sample complexity of the linear quadratic regulator," *Found. Comput. Math.*, vol. 20, no. 4, pp. 633–679, Aug. 2020.
[11] M. Fazel, R. Ge, S. Kakade, and M. Mesbahi, "Global convergence of policy gradient methods for the linear quadratic regulator," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1467–1476.
[12] A. Cohen, A. Hasidim, T. Koren, N. Lazic, Y. Mansour, and K. Talwar, "Online linear quadratic control," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1029–1038.
[13] S. Tu and B. Recht, "The gap between model-based and model-free methods on the linear quadratic regulator: An asymptotic viewpoint," in *Proc. Conf. Learn. Theory*, 2019, pp. 3036–3083.
[14] D. Lee and J. Hu, "Primal-dual Q-learning framework for LQR design," *IEEE Trans. Autom. Control*, vol. 64, no. 9, pp. 3756–3763, Sep. 2019.
[15] T. Dai and M. Sznaier, "A moments based approach to designing MIMO data driven controllers for switched systems," in *Proc. IEEE Conf. Decis. Control (CDC)*, Dec. 2018, pp. 5652–5657.
[16] C. De Persis and P. Tesi, "Formulas for data-driven control: Stabilization, optimality, and robustness," *IEEE Trans. Autom. Control*, vol. 65, no. 3, pp. 909–924, Mar. 2020.
[17] H. J. van Waarde, J. Eising, H. L. Trentelman, and M. K. Camlibel, "Data informativity: A new perspective on data-driven analysis and control," *IEEE Trans. Autom. Control*, vol. 65, no. 11, pp. 4753–4768, Nov. 2020.
[18] J. Berberich, A. Koch, C. W. Scherer, and F. Allgower, "Robust data-driven state-feedback design," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2020, pp. 1532–1538.
[19] H. J. van Waarde and M. K. Camlibel, "A matrix Finsler's lemma with applications to data-driven control," 2021, *arXiv:2103.13461*.
[20] H. J. van Waarde, M. K. Camlibel, and M. Mesbahi, "From noisy data to feedback controllers: Nonconservative design via a matrix S-Lemma," *IEEE Trans. Autom. Control*, vol. 67, no. 1, pp. 162–175, Jan. 2022.
[21] F. Dorfler, J. Coulson, and I. Markovsky, "Bridging direct and indirect data-driven control formulations via regularizations and relaxations," *IEEE Trans. Autom. Control*, vol. 68, no. 2, pp. 883–897, Feb. 2023.
[22] J. Berberich, C. W. Scherer, and F. Allgower, "Combining prior knowledge and data for robust controller design," *IEEE Trans. Autom. Control*, early access, Sep. 26, 2022, doi: 10.1109/TAC.2022.3209342.
[23] A. Bisoffi, C. D. Persis, and P. Tesi, "Controller design for robust invariance from noisy data," *IEEE Trans. Autom. Control*, vol. 68, no. 1, pp. 636–643, Jan. 2023.
[24] H. J. van Waarde, M. K. Camlibel, and H. L. Trentelman, "Data-driven analysis and design beyond common Lyapunov functions," in *Proc. IEEE 61st Conf. Decis. Control (CDC)*, Dec. 2022, pp. 2783–2788.
[25] B. Huang and U. Vaidya, "A convex approach to data-driven optimal control via Perron–Frobenius and Koopman operators," *IEEE Trans. Autom. Control*, vol. 67, no. 9, pp. 4778–4785, Sep. 2022.
[26] R. E. Skelton, T. Iwasaki, and D. E. Grigoriadis, *A Unified Algebraic Approach to Control Design*. Milton Park, U.K.: Taylor & Francis, 1997.
[27] C.-T. Chen, *Linear System Theory and Design*. London, U.K.: Oxford Univ. Press, 1995.
[28] C. Knospe, "Pid control," *IEEE Control Syst. Mag.*, vol. 26, no. 1, pp. 30–31, Feb. 2006.
[29] H. Kwakernaak and R. Sivan, *Linear Optimal Control Systems*. New York, NY, USA: Wiley-Interscience, 1972.

[30] K. J. Åström and B. Wittenmark, *Adaptive Control*. Chelmsford, MA, USA: Courier Corporation, 2013.
[31] J. C. Willems, P. Rapisarda, I. Markovsky, and B. L. M. De Moor, "A note on persistency of excitation," *Syst. Control Lett.*, vol. 54, no. 4, pp. 325–329, Apr. 2005.
[32] H. Li, X. Jing, and H. R. Karimi, "Output-feedback-based H_∞ control for vehicle suspension systems with control delay," *IEEE Trans. Ind. Electron.*, vol. 61, no. 1, pp. 436–446, Jan. 2014.
[33] P. Gahinet, A. Nemirovski, A. J. Laub, and M. Chilali, *LMI Control Toolbox*. Natick, MA, USA: MathWorks, 1995.
[34] J. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optim. Methods Softw.*, vols. 11–12, nos. 1–4, pp. 625–653, 1999.

DONGHWAN LEE (Member, IEEE) received the B.S. degree in electronic engineering from Konkuk University, Seoul, South Korea, in 2008, the M.S. degree in electrical and electronic engineering from Yonsei University, Seoul, in 2010, and the M.S. degree in mathematics and the Ph.D. degree in electrical and computer engineering from Purdue University, West Lafayette, IN, USA, in 2017. He was a Postdoctoral Research Associate at the Coordinated Science Laboratory, University of Illinois at Urbana–Champaign, IL, USA, from 2017 to 2019. He is currently an Assistant Professor with the School of Electrical Engineering, KAIST, Daejeon, South Korea. His current research interests include stochastic programming, multi-agent systems, and reinforcement learning.

DO WAN KIM received the B.S., M.S., and Ph.D. degrees from the Department of Electrical and Electronic Engineering, Yonsei University, Seoul, South Korea, in 2002, 2004, and 2007, respectively. He was at the Engineering Research Institute, Yonsei University. He also worked as a Visiting Scholar at the Department of Mechanical Engineering, University of California at Berkeley, Berkeley. In 2009, he worked as a Research Professor with the Department of Electrical and Electronic Engineering, Yonsei University. Since 2010, he has been a Professor with the Department of Electrical Engineering, Hanbat National University, Daejeon, South Korea. His current research interests include analysis and synthesis of nonlinear sampled-data control systems and autonomous unmanned vehicles.