Closed-form and robust expressions for data-driven LQ control

Federico Celi\textsuperscript{b}, Giacomo Baggio\textsuperscript{a}, Fabio Pasqualetti\textsuperscript{b,}\textsuperscript{*}

\textsuperscript{a} Department of Information Engineering, University of Padova, Padova, 35131, Italy
\textsuperscript{b} Department of Mechanical Engineering, University of California Riverside, Riverside, CA 92521, USA

\section*{A B S T R A C T}

This article provides an overview of certain direct data-driven control results, where control sequences are computed from (noisy) data collected during offline control experiments without an explicit identification of the system dynamics. For the case of noiseless datasets, we derive several closed-form data-driven expressions that solve a variety of optimal control problems for linear systems with quadratic cost functions of the state and input (including the linear quadratic regulator problem, the minimum energy control problem, and the linear quadratic control problem with terminal constraints), discuss their advantages and drawbacks with respect to alternative data-driven and model-based approaches, and showcase their effectiveness through a number of numerical studies. Interestingly, these results provide an alternative and explicit way of solving classic control problems that, for instance, does not require the solution of an implicit and recursive Riccati equation as in the model-based setting. For the case of noisy datasets, we show how the closed-form expressions derived in the noiseless setting can be modified to compensate for the bias induced by noise, and perform a sensitivity analysis to reveal favorable asymptotic robustness properties of the derived data-driven controls. We conclude the paper with some considerations and a discussion of outstanding questions and directions of future investigation.

\section*{1. Introduction}

Data-driven control refers to the design of algorithms for systems with unknown dynamics using data obtained from a set of control experiments. This approach does not require the modeling of the system dynamics using first principles, which is instead needed by the traditional model-based methods and can be challenging for certain complex systems. The design of data-driven controls can be indirect, where the control sequence is computed using a model of the system dynamics identified from data (i.e., the identification plus control pipeline), or direct, where the control sequence is computed in an end-to-end fashion without explicitly estimating the system dynamics from data. While model-based and indirect data-driven control design have a long and widely accepted history, direct data-driven control has received renewed attention, possibly motivated by the numerous successes of machine learning and artificial intelligence techniques.

This article reviews some of the recent results for direct data-control of linear time-invariant systems. Our main premise is that, especially for systems with linear time-invariant dynamics, system-theoretic properties and control methods are well-understood and have been developed over the years, ranging from tests and algorithms in the frequency domain (Åström and Murray (2010)) to methods using state-space (Kalilath, 1980) and geometric computations (Basilé & Marro, 1991), among others. The analysis, design and synthesis of control methods based on the system behaviors (Willems & Polderman, 1997) is certainly possible and interesting, but perhaps most useful only if complemented with an understanding of when a specific problem should be solved within a specific domain (e.g., frequency, state space, behaviors) and using a specific algorithm. This type of questions, which has been present, for instance in the machine learning research to characterize the tradeoffs between generative and discriminative models (Ng & Jordan, 2001), has instead received only scarce attention in the context of data-driven control. As we shall see later, while data-driven and model-based methods are theoretically equivalent in the absence of noise and assuming perfect computations, in practice, the methods can differ considerably in the way they propagate uncertainties thus leading unexpectedly to different results. Furthermore, even within the data-driven framework, different formalisms can lead to expressions with different complexity, interpretability and performance, thus motivating careful analysis and comparisons. We will focus on a specific (and, in our opinion, simple and insightful) framework to solve Linear Quadratic (LQ) control problems using data, and comment on the advantages and disadvantages of the proposed solutions. We refer the interested reader to Markovsky and Dörfler (2021) for a more comprehensive survey of data-driven control methods and the behavioral framework.

\textsuperscript{*} Corresponding author.

E-mail addresses: fceli001@ucr.edu (F. Celi), baggio@dei.unipd.it (G. Baggio), fabiopas@ucr.edu (F. Pasqualetti).

https://doi.org/10.1016/j.arcontrol.2023.100916

Received 20 July 2023; Accepted 1 November 2023

1367-5788/© 2023 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
The rest of the paper is organized as follows. Section 2 introduces the problem setting, our notation, and some preliminary results on representing linear dynamical systems using data. In particular, Lemma 2.1 contains a reformulation of Willems’ Fundamental Lemma (Willems, Rapisarda, Markovsky, & De Moor, 2005) that allows for the explicit characterization of the free and forced responses of a linear system using data, and extends the state-space data-based representation in van Waarde, De Persis, Camibellé, & Tesi (2020). Section 3 contains explicit data-driven formulas for the solution to a variety of LQ control problems. These results extend and generalize the ones obtained in Baggio, Katewa, & Pasqualetti (2019) and Baggio, Bassett, & Pasqualetti (2021) for the case of a single dataset, and the ones in Baggio & Pasqualetti (2020) and van Waarde, De Persis, et al. (2020). Lemma 3.3, in particular, further extends the data-based representation of the system dynamics in Lemma 2.1 to allow for the use of heterogeneous datasets and for the representation of trajectories longer than the ones available in the dataset. Section 4 contains our formulas for data-driven LQ control with noisy datasets, and contains (i) closed-form expressions that compensate for the effect of noise (when the noise statistics are known) and are asymptotically consistent, and (ii) a sensitivity study that shows that the effect of perturbations vanish asymptotically, when their magnitude is bounded and they affect only a “sublinear” number of datapoints. These results generalize our previous bounds in Baggio et al. (2021) and Celi, Baggio, & Pasqualetti (2023b). The case of closed-loop LQ control is treated in Section 5, where we follow the ideas in Celi, Baggio, & Pasqualetti (2022) and show how the Linear Quadratic Regulator (LQR) gains can be learned in a data-driven setting without identifying the system dynamics nor solving iterative, implicit Riccati equations (Theorem 5.1 contains our data-driven closed-form expressions of the LQR gain). Finally, Section 6 concludes the paper.

Notation. We let $\mathbb{R}$ and $\mathbb{N}$ denote the set of real and integer numbers, respectively. Given a matrix $A \in \mathbb{R}^{m,n}$, we let $\text{Rank}(A)$, $\text{Basis}(A)$, $\text{Ker}(A)$, $A^T$, $A_{\text{min}}(A)$, $A_{\text{max}}(A)$ denote the rank, a basis of the column space, the kernel, the transpos, and the smallest and largest singular values of $A$, respectively. We let $\text{blk-diag}(A_1, \ldots, A_r)$ be the block diagonal matrix with blocks $A_i \in \mathbb{R}^{m_i,n_i}$. We denote the Moore–Penrose pseudoinverse of matrix $A$ with $A^+$. We indicate the 2-norm of a matrix or vector with $\| \|$.

We let $\Lambda > 0$ denote a positive definite (positive semidefinite) matrix. We let $\text{vec}(A)$ be the vectorization of matrix $A$. For a positive semidefinite matrix $W \in \mathbb{R}^{n,n}$ and vector $x \in \mathbb{R}^n$, we let $|x|_W = \sqrt{x^T W x}$, $I_n \in \mathbb{R}^{n,n}$ denote the $n \times n$ identity matrix and $x^T x$ the zero matrix, respectively (subscripts will be omitted when clear from the context). For a random vector $x : \mathbb{S} \rightarrow \mathbb{R}^n$, we let $P[x \in S]$ and $E[x]$ be the probability that $x$ takes on a value in a set $S \subseteq \mathbb{R}^n$ and the expected value of $x$, respectively. We let a.s. denote almost surely, and $\rightarrow a.s.$ almost sure convergence.

2. Problem setting and preliminary notions

We study the problem of designing control inputs for linear time-invariant systems to solve a variety of optimal control and robustness problems. We do this without knowing the system dynamics and by, instead, leveraging a set of pre-recorded input–output trajectories, together with state trajectories when needed. In particular, we consider systems with linear, discrete-time, time-invariant dynamics of the form

$$
\begin{align*}
x(t+1) &= Ax(t) + Bu(t), \\
y(t) &= Cx(t) + Du(t),
\end{align*}
$$

(1)

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$ and $y \in \mathbb{R}^p$ are the state, input and output vectors at time $t \in \mathbb{N}$, respectively, and the matrices $A, B, C$ and $D$ are unknown. We assume that inputs $u$, states $x$ and outputs $y$ are available from a set of $N \in \mathbb{N}$ control experiments with finite horizon $T \in \mathbb{N}$:

$$
U = [u_T^1 \quad \cdots \quad u_T^N] \in \mathbb{R}^{mT \times N},
$$

(2a)

$$
X = [x_T^1 \quad \cdots \quad x_T^N] \in \mathbb{R}^{pT \times N},
$$

(2b)

$$
Y = [y_T^1 \quad \cdots \quad y_T^N] \in \mathbb{R}^{qT \times N}.
$$

(2d)

In (2), $u_T^i$, $x_T^i$, and $y_T^i$ are the vectors containing the inputs, states, and outputs of the $i$th experiment:

$$
\begin{align*}
u_T^i &= \text{vec}(u(0), \ldots, u(T-1)), \\
x_T^i &= \text{vec}(x(1), x(2), \ldots, x(T)), \\
y_T^i &= \text{vec}(y(0), \ldots, y(T-1)).
\end{align*}
$$

(2e)

We remark that the full input-state-output dataset (2) is not always needed to solve the problems described in this tutorial paper; the subset of required data will be specified based on the problem at hand. Further, when convenient, we will use the following matrices, which can be extracted from (2):

$$
\begin{align*}
X_T &= [x(0) \quad \cdots \quad x(N)]\in \mathbb{R}^{pN}, \\
Y_T &= [y(0) \quad \cdots \quad y(N)]\in \mathbb{R}^{qN}.
\end{align*}
$$

(2f)

For the noiseless system (1), knowledge of the system matrices $A$, $B$, $C$ and $D$ is equivalent to the availability of a (sufficiently rich) dataset (2). In fact, any dataset (2) can be generated with the matrices $A$, $B$, $C$ and $D$ using (1), and, in turn, such matrices can be reconstructed uniquely (under mild data rank conditions) using the dataset (2):

$$
\begin{align*}
X_m &= [A \quad B]\begin{bmatrix} X_m \\ U_m \end{bmatrix}, \\
Y_m &= [C \quad D]\begin{bmatrix} X_m \\ U_m \end{bmatrix},
\end{align*}
$$

where

$$
\begin{align*}
U_m &= [u(0), \ldots, u(T-1), \ldots, u(N)\ldots, u(N)\ldots] \\
X_m &= [x(0), \ldots, x(T), \ldots, x(0)\ldots, x(0)\ldots] \\
Y_m &= [y(0), \ldots, y(T), \ldots, y(0)\ldots, y(0)\ldots] \\
X_m &= [x(1), \ldots, x(T), \ldots, x(0)\ldots, x(0)\ldots] \\
Y_m &= [y(1), \ldots, y(T), \ldots, y(0)\ldots, y(0)\ldots] \\
X_m &= [x(1), \ldots, x(T), \ldots, x(0)\ldots, x(0)\ldots] \\
Y_m &= [y(1), \ldots, y(T), \ldots, y(0)\ldots, y(0)\ldots]
\end{align*}
$$

Clearly, the system matrices can be computed uniquely whenever the matrix $X_m \in \mathbb{R}^{pN}$ is full row rank. While this analysis seems to discourage the pursuit of data-driven methods, since sufficiently-rich datasets are effectively a model of the system dynamics, in this review we will show that, instead, data-driven computations allow for alternative and sometimes more insightful, direct, and computationally-favorable solutions to classic control problems, thus contributing to the theory of systems and enriching our control tools. We remark that a detailed analysis of the tradeoffs between direct data-driven methods and classic identification-based approaches in noisy settings is much more nuanced (Krishnan & Pasqualetti, 2021), deserves a dedicated treatment, and will not be addressed here. Rather, we will show how our direct solutions obtained with noiseless data can be used to study, and modified to counteract, the effect of noise and perturbations on the datasets.

A powerful result underlying most data-driven approaches is Willems’ Fundamental Lemma (Willems et al., 2005), which, loosely speaking, gives a sufficient conditions under which any $T$-long trajectory of the system (1) can be constructed as linear combinations of those in an appropriately constructed input–output dataset (Markovsky & Dörfler, 2021). We next state a reformulation of Willems’ Fundamental Lemma that uses the dataset (2) and that allows us to distinguish between the trajectories of (1) obtained with and without a control input. This result will be instrumental for our derivations.

1 Similar formulas are contained also in De Persis & Tesi (2020), among others.
Lemma 2.1 (Data-based Free and Forced Representation Celi & Pasqualetti, 2022). Let (2)–(3) be the data generated by the system (1). Assume that
\[
\text{Rank} \left( \begin{bmatrix} X_0 & U \end{bmatrix} \right) = mT + n. \tag{4}
\]
Let \( K_U = \text{Basis}(\text{Ker}(U)) \) and \( K_0 = \text{Basis}(\text{Ker}(X_0)) \). Then, for any initial state \( x_0 \) and input \( u_T \), there exist vectors \( \alpha \) and \( \beta \) such that \( x_0 = X_0 K_U \alpha \) and \( u_T = U K_0 \beta \). Moreover,
\[
\begin{bmatrix} x_T \vspace{1em} \alpha \end{bmatrix} = \begin{bmatrix} X_K & X_K \beta \vspace{1em} \alpha \end{bmatrix}[x] \begin{bmatrix} Y \vspace{1em} \beta \end{bmatrix} \tag{5}
\]
are the state and output trajectories of length \( T \) of (1) generated by \( x_0 \) and \( u_T \).

Lemma 2.1 states that any trajectory of the system (1) can be written as a linear combination of a collection of previously-recorded trajectories. Yet, Lemma 2.1 provides a more granular decomposition of the trajectories of the system (1) given data, as it identifies the free response of the system from the initial condition free of the trajectories of the system (1) given data, as it identifies the and \( u_T \) and \( \beta \). Let \( \Sigma_T = \begin{bmatrix} C^T & \cdots & (C A^{T-1})^T \end{bmatrix}^T \), is the \( T \)-steps observability matrix of (1). Similarly,
\[
\text{Rank}(C_T) = \text{Rank}(X_K K_0), \tag{7}
\]
where \( C_T = \begin{bmatrix} A^{T-1} B \cdots A B \end{bmatrix} \) is the \( T \)-steps controllability matrix of (1) and \( X_K \) is as in (3).

Lemma 2.2 relates the rank of the \( T \)-step observability matrix \( O_T \) to the data-driven matrix \( Y K_0 \). Clearly, when \( T \geq n \), the system (1) is observable if and only if \( \text{Rank}(Y K_0) = n \). Similar comments hold in (7) for the reachability subspace of (1).

Remark 1 (Single vs. Multiple Data Trajectories). A single experimental trajectory may be sufficient to obtain a data-driven representation of the system dynamics. A single trajectory, in fact, is used in Willems’ Fundamental Lemma and in several reformulations of this result, e.g., Lopez and Müller (2022), Schmitz, Faulwasser, and Worthmann (2022), Verhoek, Tóth, Haesaert, and Koch (2021), Willems et al. (2005) and Yu et al. (2021). We remark that the use of multiple trajectories, as we do in Lemma 2.1, carries some advantages. First, the formulas in Lemma 2.1 remain valid if only a single, long, trajectory is available. In fact, the case of a single, long trajectory organized as a Hankel matrix is a special case of our formalism. To see this, using the notation in van Waarde, De Persis et al. (2020), the data collected from a single trajectory of length \( \tau \) is organized as
\[
H_T(x) = \begin{bmatrix} x(0) \vspace{1em} x(1) \vspace{1em} \cdots \vspace{1em} x(\tau - T) \end{bmatrix}, \tag{28}
\]
\[
H_T(u) = \begin{bmatrix} u(0) \vspace{1em} u(1) \vspace{1em} \cdots \vspace{1em} u(\tau - T) \vspace{1em} \vdots \vspace{1em} \vdots \vspace{1em} \vdots \vspace{1em} \vdots \vspace{1em} \vdots \vspace{1em} \vdots \vspace{1em} \vdots \vspace{1em} \vdots \vspace{1em} u(T - 1) \end{bmatrix}. \tag{29}
\]
Clearly, one can set \( X_T = H_T(x) \) and \( u_T \) equal to the \( t \)-th column of \( H_T(u) \) to equivalently express the data as in our framework. Thus, considering multiple, short trajectories as in (2) effectively generalizes and includes the case of a single, long trajectory. Second, the use of multiple trajectories is convenient when the dynamics are unstable, since the system needs to be simulated for a shorter time horizon compared to the case of a single trajectory. More generally, using multiple trajectories produces data matrices that are numerically better conditioned and yield more reliable computations. Finally, the use of multiple trajectories is convenient from a statistical perspective when the collected data is corrupted by noise (Tu, Frostig, & Soltanolkotabi, 2022).

Remark 2 (State vs. Output Measurements). We assume here that the state of the system (1) can be directly measured. Notice, however, that this is not a restrictive assumption since a state measurement can be replaced with a finite window of inputs and outputs to solve appropriate control problems. In fact, the dynamics of (1) can be equivalently written using only inputs and outputs as done, e.g., in Al Makdah, Krishnan, Katewa, and Pasqualetti (2022) for the data-driven LQG control problem.

3. Data-driven formulas for open-loop LQ control

We start by studying the LQ control problem
\[
\text{minimize} \quad \sum_{t=0}^{T-1} \left( \|y(t)\|_Q^2 + \|u(t)\|_R^2 \right) \tag{8}
\]
subject to \( x(t + 1) = A x(t) + B u(t), \)
\( y(t) = C x(t) + D u(t), \)
\( x(0) = x_0, \quad y(T - 1) = y_T, \)
where \( Q_r \geq 0 \) and \( R_r > 0 \) are the (time-varying) output and input weighting matrices, \( x_0 \) is the initial state, and \( y_T \) the desired value of the output at time \( T - 1 \). Throughout the paper, we assume that (1) is output controllable in \( T - 1 \) steps\(^2\) to guarantee feasibility of (8) for any choice of \( x_0 \in \mathbb{R}^n \) and \( y_T \in \mathbb{R}^n \), and that the rank condition (4) holds. Problem (8) generalizes the classic (open-loop) linear–quadratic control framework by including the possibility of minimizing a linear function of the state (as opposed to the whole state) in addition to the control input. Let \( u_T = \text{vec}(u(0), \ldots, u(T - 1)) \) be the vector of inputs, and
\[
Q = \text{blk-diag}(Q_0, \ldots, Q_{T-1}), \quad R = \text{blk-diag}(R_0, \ldots, R_{T-1}), \tag{9}
\]
Using the notation in Section 2, we now present a closed-form solution to the LQ control problem (8) that relies only on the data collected in (2). First, notice that the cost in (8) can be written in vector form as
\[
\sum_{t=0}^{T-1} \left( \|y(t)\|_Q^2 + \|u(t)\|_R^2 \right) = y_T^T Q y_T + u_T^T R u_T. \tag{10}
\]
Second, from Lemma 2.1 we have \( y_T = Y K_U \alpha + Y K_0 \beta \) and \( u_T = U K_0 \beta \), for some vectors \( \alpha \) and \( \beta \). Then, by letting \( K = [K_U, K_0] \), the input and output trajectories can be equivalently written as
\[
y_T = Y K \begin{bmatrix} \alpha \beta \end{bmatrix} \quad \text{and} \quad u_T = U K \begin{bmatrix} \alpha \beta \end{bmatrix}, \tag{11}
\]
respectively, and the cost in (8) becomes
\[
\begin{bmatrix} \alpha \beta \end{bmatrix}^T \left( (Y K)^T Q (Y K) + (U K)^T R (U K) \right) \begin{bmatrix} \alpha \beta \end{bmatrix}. \tag{12}
\]
Similarly, the equality constraints in (8) can be written as
\[
\begin{bmatrix} x_0 \vspace{1em} \alpha \vspace{1em} \beta \end{bmatrix} \begin{bmatrix} Y \vspace{1em} K \vspace{1em} K \end{bmatrix} = \begin{bmatrix} x_0 \vspace{1em} y_T \end{bmatrix}. \tag{13}
\]
\(^2\) System (1) is output controllable in \( T \) steps if \( [C A^{T-1} B \cdots C A B \ C B \ C D] \) has full row rank. We remark that Lemma 2.2 can be adapted to verify whether a system is output controllable directly from data, substituting \( X_K \) with \( Y_K \).
The above reasoning allows us to reformulate the LQ control problem (8) as the data-based problem

\[
\begin{align*}
\text{minimize} & \quad \|Ly\|_2^2 \\
\text{subject to} & \quad W\gamma = z,
\end{align*}
\]

(10)

where

\[
\gamma = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \quad L = \begin{bmatrix} Q^{1/2}YK \\ R^{1/2}U \end{bmatrix}, \quad W = \begin{bmatrix} X_0 & Y_0 \end{bmatrix}, \quad z = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix},
\]

which admits the solution

\[
\gamma^* = (I - K_W(LK_W)^\dagger L)W^{-1}z,
\]

with \(K_W = \text{Basis(Ker}(W))\). This leads to the following data-driven solution to Problem (8).

**Theorem 3.1 (Data-driven LQ Control).** The input \(u_t^*\) that solves the LQ control problem (8) is

\[
u_t^* = UK(I - K_W(LK_W)^\dagger L)W^{-1}\begin{bmatrix} x_0 \\ y_0 \end{bmatrix}.
\]

(11)

With the additional assumption that \(\text{Rank}(Q^{1/2}Q^T) = n_i^3\) an alternative expression for \(u_t^*\) that does not require the computation of kernel matrices is extracted from

\[
x_t = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = P^{-1/2} \begin{bmatrix} X_0 \\ Y_0 \end{bmatrix} U^\dagger \begin{bmatrix} \alpha \\ \beta \end{bmatrix},
\]

(12)

where

\[
P = \begin{bmatrix} Y & X_0 \\ \begin{bmatrix} 0 \\ U \end{bmatrix} \end{bmatrix}^\dagger Q \begin{bmatrix} X_0 \\ Y_0 \end{bmatrix} U^\dagger + \begin{bmatrix} 0 & 0 \\ 0 & R \end{bmatrix}.
\]

(13)

A proof of (12) is postponed to the Appendix. We remark that (11) and (12) rely on assumption (4). In practice, this imposes a lower bound on the number \(N\) of trajectories recorded in (2), i.e., \(N \geq mT + n\). In fact, when this condition is not satisfied, then the Problem (10) may be infeasible or not return the desired optimal control for some choices of the initial condition \(x_0\) and target output \(y_t\).

**Remark 3 (Closed-form Solutions vs. Iterative and Optimization-Based Solutions).** The distinctive aspect of the above results lies in the utilization of a closed-form formula, as shown in Theorem 3.1 and in most of the results discussed in this work. While exceptions do exist, such as those in Pellegriino, Blanchini, Fenu, and Salvato (2023a, 2023b) and da Silva, Bazanella, Lorenzini, and Campestrini (2018), it is noteworthy that much of the existing literature on data-driven control predominantly revolves around iterative and optimization-based methodologies. It is essential to acknowledge that certain problems necessitate optimization-based approaches, often those involving non-linear systems or input–output constraints. Nevertheless, when the opportunity for closed-form solutions arises, closed-form expressions provide not only valuable insights into problem solutions but also confer numerical advantages over alternative methods, see e.g., Celi et al. (2022) for a discussion on the computational advantages of closed-form formulas over optimization-based solutions. Further, closed-form formulas have been recently used in the context of data-driven control to solve a set of diverse problems, such as the computation of the Kalman Filter and of the Linear Quadratic Gaussian regulator (Al Makhad \\& Pasqualetti, 2023), the identification of geometric invariant subspaces (Celi \\& Pasqualetti, 2022), and the solution of the eigenstructure assignment problem (Celi, Baggio, \\& Pasqualetti, 2023a).

---

**Example 1.** We now apply the result from Theorem 3.1 to control a simplified model of a quadcopter. The quadcopter is dimensioned after a Crazyflie drone, with data (2) collected using Matlab simulations. For more details on the model of the system, we refer the reader to Wang, Man, Cao, Zheng, and Zhao (2016). The state of the system is

\[
x = \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \\ y \\ \dot{y} \\ \phi \\ \dot{\phi} \\ z \\ \dot{z} \\ \psi \\ \dot{\psi} \end{bmatrix}^T,
\]

where \((x, y, z)\) and \((\dot{x}, \dot{y}, \dot{z})\) are the coordinates and linear velocities, respectively, and \((\theta, \dot{\theta}, \phi, \dot{\phi}, \psi, \dot{\psi})\) are the roll, pitch and yaw angles and their respective velocities. The measured outputs are limited to the coordinates \((x, y, z)\) and the asset angles \((\theta, \phi, \psi)\). The system is simulated in Fig. 1, where \(u_t\) is obtained from (11).

---

**3.1. Minimum energy control**

The minimum-energy control problem (ME) is obtained from (8) by letting \(Q_t = 0\) and \(R_t = I\). It reads as (Kailath, 1980; Pasqualetti, Zampieri, \\& Bullo, 2014)

\[
\min_t \sum_{t=1}^{T} \|u(t)\|_2^2
\]

s.t. \(x(t+1) = Ax(t) + Bu(t),\)

\(y(t) = Cx(t) + Du(t),\)

\(x(0) = x_0, \quad y(T-1) = y_f.\)

Clearly, a solution to this problem can be obtained from Theorem 3.1 by simply letting \(L = UK\). An insightful instance of the minimum energy control problem (14) is when \(x_0 = 0\) and \(y_f = 0\), that is, the initial conditions of the experimental data and of the control problem are all equal to zero. In this case, although Lemma 2.1 cannot be used for a data-based representation of the system trajectories since \(x_0\) is not full-row rank, a direct expression for the minimum energy input can readily be obtained.

**Theorem 3.2 (Data-driven ME Control Baggio et al., 2019).** Let \(x_0 = 0\) and \(\text{Rank}(U) = nT\). Then, the input \(u_t^*\) that solves the minimum energy control problem (14) is

\[
u_t^* = U(I - K_{Yf}(UK_{Ye})^\dagger U)^{\dagger} y_t,
\]

(15)

where \(Y_f\) are the final states of the recorded trajectories (3) and \(K_{Ye} = \text{Basis(Ker}(Y_f)).\)

Eq. (15) is obtained from (11) leveraging the simplifications due to \(x_0 = 0\). Along the lines of the derivation of (12), the following alternative minimum-energy control expression holds:

\[
u_t^* = (Y_f^\dagger)^{\dagger} y_t.
\]

(16)
involved in computing and manipulating the controllability Gramian. We let than the exact data-driven formulas in (16) and (19) for systems with a large number of states. This loss of numerical precision is a consequence of the computational process corresponding errors in the final state, as a function of the system dimension yet suboptimal (the cost achieved by the control is not optimal) solution for finite values of $N$.

When $C = I$ and $D = 0$ this formula offers a data-driven way to compute the $T$-step Gramian and its eigenvalues. In fact, it can be seen that in this case Eq. (16) can be written as a function of $X_T$ and the controllability matrix in (7) equals $C_T = X_T U^T$. Consequently, the Gramian satisfies

$$W_T = C_T C_T^T = X_T U U^T X_T^T.$$  \hspace{1cm} (17)

Similarly, since the smallest (resp. largest) eigenvalues of the Gramian identify the states that require largest (resp. smallest) input energy, these can also be computed as

$$
\sigma_{\min}^{-1}(W_T) = \max_{1 \leq i \leq n} \| (X_T U)^{\dagger} x_i \|_2^2 = \sigma_{\max}^2 ((X_T U)^{\dagger})^2),
$$

$$
\sigma_{\max}^{-1}(W_T) = \min_{1 \leq i \leq n} \| (X_T U)^{\dagger} x_i \|_2^2 = \sigma_{\min}^2 ((X_T U)^{\dagger})^2).$$  \hspace{1cm} (18)

Finally, when the entries of $U$ are i.i.d. random variables with zero mean and nonzero finite variance, one can obtain simplified expressions of the minimum energy control input. For instance, the simplified expression

$$\hat{u}_T = U X_T^\text{d} x_1$$

converges to the minimum-energy input $u^*_T$ almost surely, as the number of trajectories $N$ increases (Baggio et al., 2019). Since the minimum energy input is unique and for any finite value of $N$ it generally holds $\hat{u}_T \neq u^*_T$, it follows that (19) is a suboptimal input for the minimum energy problem (14). In particular, the input (19) drives the system to the desired final state $x_T$ with non-minimum energy, while requiring fewer numerical operations for its computation when compared with (15), (16). This property can be useful when dealing with large (network) systems, for which these calculations are generally ill-conditioned (Baggio et al., 2021, 2019; Pasqualetti et al., 2014). In support of these claims, in Fig. 2 we perform a series of numerical experiments, where we assess the numerical robustness and accuracy of direct data-driven controls. We notice that the accuracy in computing the minimum-energy control input using the data-driven expression (16) is comparable to that achieved when using the model-based counterpart, yet numerically more accurate than the model-based Gramian formula. Further, in Fig. 2(c)-(d) we notice that the accuracy of the Gramian-based control input decreases as $n$ increases, while the data-driven expressions of the minimum-energy control inputs remain accurate for systems of considerably larger dimension.

As we progress through this paper, the closed-form expressions outlined in this section will form the foundational building blocks for deriving novel closed-form solutions to a variety of problems. Among these, in the following sections we will cover the infinite-horizon linear quadratic regulator problem (Celi et al., 2022), the linear quadratic Gaussian control problem (Al Makdah & Pasqualetti, 2023), the distributed data-driven linear quadratic control problem (Celi et al., 2023b). Further, we will answer several robustness questions related to data-driven control (Angluri, Al Makdah, Katewa, & Pasqualetti, 2020; Celi et al., 2023b).

### 3.2. Datasets with heterogeneous control horizons

The results presented thus far assume that the experimental trajectories have the same length. In practice, however, it may be more convenient to collect data from heterogeneous experiments with different control horizons. The first question that we answer is whether the trajectories of the system can be represented as a linear combination of trajectories of different lengths, and for which control horizon. Is it possible to represent trajectories of length greater than those of the control experiments? Ultimately, this analysis leads to an extension of Lemma 2.1.

To formalize the discussion, assume that the control experiments are performed using $M$ distinct horizons $T_i \in \mathbb{N}$, $i \in \{1, \ldots, M\}$, and that the available data is organized as $(U_i, X_{0,i}, X_i, Y_i)$ for $i \in \{1, \ldots, M\}$, where the $i$th set contains $N_i$ experiments, and $X_{0,i} \in \mathbb{R}^{n \times N_i}$, $U_i \in \mathbb{R}^{n \times N_i}$, $X_i \in \mathbb{R}^{n \times N_i}$, and $Y_i \in \mathbb{R}^{p \times N_i}$ denote the matrices containing the initial states of the experiments, and the input, state, and output sequences with horizon $T_i$. Finally, let $X(f_i)$ contain the last $n$ rows of $X_i$, and $D_H = \{(U_i, X_{0,i}, X_i, Y_i)\}_{i=1}^{M}$ be the set of heterogeneous data. See Fig. 3 for an illustration of the heterogeneous dataset.

We now present an extension of Lemma 2.1 that allows us to represent trajectories of (1) of length $T$ using the heterogeneous data $D_H$, where the horizon $T$ is an integer-weighted combination of the experimental horizons $T_i$ (e.g., if $T_1 = 2$, $T_2 = 3$, and $T_3 = 8$, we could take $T = T_2 + 2T_3 = 19$). The main ideas behind this result are that a trajectory of length $T$ can be broken up in multiple sub-trajectories, where the initial state of each sub-trajectory equals the final state of the previous sub-trajectory, and each sub-trajectory admits a data-based representation as in Lemma 2.1 (see also Fig. 4). We remark that the decomposition of a trajectory into multiple parts of a certain length may not be unique, thus leading to potentially multiple data-driven representations of the same trajectory when using heterogeneous data (for instance, if $T_1 = 2$, $T_2 = 3$, and $T_3 = 8$, then we could take $T = 19 = 2T_1 + T_2 + 2T_3 = T_1 + T_2 + T_3$).
of Lemma 3.3 is obtained by suitably “gluing” together the data-based representations of system trajectories of lengths \( T_1, \ldots, T_M \).
Note in particular that the horizon \( T \) can be longer than the horizons of the experimental trajectories, thus allowing for the representation of trajectories that have never been observed during the experiments. As a special case, when the experimental trajectories have the same horizon, that is \( M = 1 \), it is possible to reconstruct from data system trajectories with horizons equal to any multiple integer of \( T_1 \). This is illustrated in the next example.

**Example 2** (Data-based Representation of Trajectories Longer Than The Experimental Data as in Lemma 3.3). Consider the scalar system

\[ x(t + 1) = ax(t) + u(t), \quad a \in \mathbb{R}, \]

and the dataset with horizon \( T = 2 \)

\[ U_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad X_{0,1} = \begin{bmatrix} 1 & 0 \\ 0 \end{bmatrix}, \quad X_{1} = \begin{bmatrix} a & 1 \\ a^2 & a \end{bmatrix}. \]

It holds

\[ K_{U,1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad K_{0,1} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad V_1 = a^2, \quad Z_1 = \begin{bmatrix} a & 1 \end{bmatrix}. \]

From Lemma 3.3, for any \( x_0 \) and \( u_i \), there exist \( a \) and \( \beta \) such that

\[
\begin{align*}
x_0 &= X_{0,1} K_{U,1} a = a, \\
u_i &= \text{blk-diag}(U_1 K_{0,1}, U_1 K_{1,1}) \beta = \beta,
\end{align*}
\]

and the state trajectory with initial condition \( x_0 \) and input sequence \( u_i \) in the interval \([1,4]\) is given by

\[
\begin{bmatrix} x_i \end{bmatrix} = \begin{bmatrix} X_{1} K_{U,1} & X_{1} K_{0,1} & 0 \\ X_{1} F_{U,1} V_1 & X_{1} F_{0,1} Z_1 & X_{1} K_{0,1} \end{bmatrix} \begin{bmatrix} a \\ a^2 \\ a^3 \\ a^4 \\ a^5 \\ a^6 \end{bmatrix} \beta.
\]

In addition to providing a new data-based representation of the trajectories of \((1)\) that allows for multiple heterogeneous experiments, Lemma 3.3 can also be used to derive data-driven formulas that solve LQ control problems, among others. For instance, under the assumption of Lemma 3.3, the data-driven LQ control expression in (3.1) remains valid when the dataset \( D_\mu \) is used by redefining \( L \) and \( W \) as follows:

\[
L = \begin{bmatrix} Q^{1/2} Y_{\ell,1} F_{\ell,1} \\ R^{1/2} U_{\ell,1} F_{\ell,1} \end{bmatrix}, \quad W = \begin{bmatrix} X_{0,\ell} K_{U,\ell} \\ X_{0,\ell} K_{0,\ell} \end{bmatrix} Y_{\ell,1,F},
\]

where \( U_{\ell,1,F} = \text{blk-diag}(U_{\ell,1} K_{0,\ell}, \ldots, U_{\ell,1} K_{0,\ell}) \) and \( Y_{\ell,1,F} \) denote the matrices consisting of the last \( \rho T \) and \( \rho \) rows, respectively, of the matrix in (21).

**Remark 4** (Alternative Approaches to Using Heterogeneous Datasets). Alternative data-based representations of system trajectories with heterogeneous datasets have been proposed in Baggio and Pasqualetti (2020) and van Waarde, De Persis et al. (2020). In Baggio and Pasqualetti (2020), the experimental data consist of inputs, initial and final state recordings with different time horizons \( T_i \). The resulting data-based representation of state trajectories is given in sampled form and exploits a data-based reconstruction of \( A^T \). In van Waarde, De Persis et al. (2020) a generalization of Willems’ Fundamental Lemma to the case of multiple trajectories is presented. Differently from our approach, this generalization does not seem to allow for the computation of trajectories longer than the experiments. □
Box I.

4. Robustness of data-driven open-loop LQ control

One of the main advantages of the direct data-driven approach with explicit formulas presented thus far is the possibility to analytically explore the impact of perturbations on these formulas and easily compute their sensitivity to parameter variations. In fact, such an analysis yields a way to quantify and improve the robustness of the direct data-driven approach to noisy and corrupted datasets. We consider two cases, which differ in the available knowledge of the statistics of the perturbations affecting the data. Let the perturbed dataset be

\[ U = U + \Delta U, \]
\[ \tilde{X}_0 = X_0 + \Delta_0, \]
\[ \tilde{X} = X + \Delta X, \]
\[ \tilde{Y} = Y + \Delta Y, \]

where \( U, X_0, X, Y \) denote the ground truth values as in (2) and \( \Delta_0, \Delta_0, \Delta_X, \Delta_Y \) contain stochastic perturbations. For the purpose of this discussion, we focus on the data-driven control in (12). However, the analysis can be adapted to other data-driven control expressions.

4.1. Data-driven LQ control with known noise statistics

We start with the standard scenario of i.i.d. perturbations characterized by known second-order statistics. In particular, we assume that \( \Delta_0, \Delta_0, \Delta_X, \Delta_Y \) are random matrices consisting of i.i.d. entries with zero mean and variance \( \sigma_0^2, \sigma_0^2, \sigma_X^2, \) and \( \sigma_Y^2 \), respectively. In this setting, the data-driven control (12) is not consistent; that is, it does not converge to the true optimal control input even when the amount of available data grows to infinity. This lack of consistency is due to the pseuoinverse operation

\[
\begin{bmatrix} X_0 \end{bmatrix}^T = \begin{bmatrix} X_0 \end{bmatrix}^T \begin{bmatrix} \tilde{X}_0 \end{bmatrix} \begin{bmatrix} \tilde{X}_0 \end{bmatrix}^T, \]

which contains quadratic terms (namely, \( \tilde{X}_0 \tilde{X}_0^T, \tilde{U} \tilde{U}^T \) that, by the law of large numbers, introduce variance-dependent biases (namely, \( \sigma_0^2 N I \) and \( \sigma_Y^2 N I \)) as \( N \) grows. However, since the noise variances are known, we can include correction terms in (12) that compensate for these biases and achieve asymptotically accurate data-driven formulas. Specifically, (12) can be modified as follows:

\[
\begin{bmatrix} X_0 \end{bmatrix}^T = (\tilde{X}_0 \tilde{X}_0^T, \tilde{U} \tilde{U}^T)^T, \]

and the following consistency result holds (see the Appendix for the proof).

**Theorem 4.1 (Asymptotic Consistency of 23).** Assume that

(i) the columns of \( X_0 \) and \( U \) are i.i.d. and satisfy the condition (4) almost surely as \( N \to \infty \), and

(ii) the entries of \( \Delta_0, \Delta_0, \) and \( \Delta_X \) are i.i.d. with zero mean and variances \( \sigma_0^2, \sigma_0^2, \sigma_X^2 \).

Then, \( u^c \) in (23) converges almost surely to \( u^c \) as \( N \to \infty \).

**Fig. 5.** This figure shows the asymptotic consistency of 23 for problem (8). The left panel shows how the controller computed with a noisy dataset with noise compensation in 23 (solid lines) asymptotically converges to the optimal controller \( u^c \) for (8), as opposed to the non-compensated expression in (12) (dashed lines), for varying noise statistics. The experiment is performed over a randomly generated system with \( n = 5, m = 3, p = 2, T = 15 \). Finally, \( \sigma_0 = \sigma_0 = 0.1 \), while different values of \( \sigma_0 \) are shown in the legend. The right panel further supports this result by showing how the difference between \( P \) and \( P_c \) evolves as the number of experiment increases and the noise in the data is appropriately accounted for as shown in Theorem 4.1 (here, \( \sigma_0 = 0.5 \)).

4.2. Data-driven LQ control with unknown noise statistics

In the more general case of (possibly correlated) perturbations with unknown second-order statistics, the robustness of (12) can be assessed through a local sensitivity analysis. Here, we assume for simplicity that the noise acts on the output measurements only, that is \( \Delta_0 = \Delta_0 = 0 \). However, an analysis similar to the one that follows can be carried out also for noisy inputs and states. Let

\[
F(U, X_0, Y) = (\tilde{X}_0 \tilde{X}_0^T, \tilde{U} \tilde{U}^T)^T + \begin{bmatrix} \sigma_0^2 N I & 0 \\ 0 & \sigma_Y^2 N I \end{bmatrix}, \]

denote the data-driven control map (12) with \( P \) as in (13), and \( \supp(\Delta_0) = \{ i : \delta_{x,i} \neq 0 \} \), with \( \delta_{x,i} = \text{vec}(\Delta_0)_i \), denote the set of corrupted entries of \( Y \). Since \( F \) is Fréchet-differentiable with respect to
Asymptotic Robustness for Sublinear Number of Perturbations. In addition to the assumptions in Lemma 4.2, assume also that the entries of $\Delta_Y$ are independent of $N$. Then, if the cardinality of $\supp(\Delta_Y)$ grows sublinearly with $N$, for any $\tau > 0$,

$$
\lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} \mathbb{P}(\|\Delta_Y\| \geq \tau) = 0.
$$

Theorem 4.3 guarantees that the error in the final output decreases to zero when $N$ increases, regardless of $\Delta_Y$ (see Fig. 6 for a numerical example). This ensures the robustness of the data-driven control action for small, possibly adversarial, perturbations.

Remark 5 (Comparison with Existing Approaches). Numerous studies have focused on developing data-driven controllers that can effectively handle disturbances generated by worst-case or stochastic noise models. Most of the existing approaches rely on robustified versions of data-based optimization problems, typically achieved through suitable regularizations (e.g., see Berberich, Kübler, Müller, & Allgöwer, 2020; Breschi, Chioua, & Formentin, 2023; Coulson, Lygeros, & Dörfler, 2021; De Persis & Tesi, 2021; Dörfler, Tesi, & De Persis, 2023) or by leveraging classic robust control tools (e.g., see Berberich, Scherer, & Allgöwer, 2022; Bissi, De Persis, & Tesi, 2021, 2022; van Waarde, Camlibel, & Mesbah, 2020). The distinctive feature of the approach of this paper lies in the use of closed-form expressions which allows to asymptotically compensate the influence of noise and to explicitly characterize the sensitivity of data-driven controls.
In particular, we have $K$ input, state, and output trajectories collected offline, and to analyze the problem via static feedback in a purely data-driven manner. Celi et al. (2023a), which solve the general eigenstructure assignment and Pasqualetti (2022), De Persis and Tesi (2020) and Bianchin (2023), back gains can be found, among others, in Al Makdah et al. (2022), Celi Alternative data-driven approaches to Theorem 5.1 to compute feed-
Zare, Soltanolkotabi, & Jovanović, 2019; Recht, 2019), among others. & Mesbahi, 2018; Gravell, Esfahani, & Summers, 2020; Mohammadi, 1994; Dean, Mania, Matni, Recht, & Tu, 2020; Fazel, Ge, Kakade, Dörfler, Tesi, & De Persis, 2022; Rotulo, De Persis, & Tesi, 2020), schemes (Coulson, Lygeros, & Dörfler, 2019; De Persis & Tesi, 2020; Dörfler, Tesi, & De Persis, 2022; Rotulo, De Persis, & Tesi, 2020), or iterative schemes inspired by reinforcement learning approaches (Abbasi-Yadkori, Lazic, & Szepesvári, 2019; Bradtke, Ydstie, & Barto, 1994; Dean, Mania, Matni, Recht, & Tu, 2020; Fazel, Ge, Kakade, & Mesbahi, 2018; Gravell, Esfahani, & Summers, 2020; Mohammadi, Zare, Soltanolkotabi, & Jovanović, 2019; Recht, 2019), among others. Alternative data-driven approaches to Theorem 5.1 to compute feedback gains can be found, among others, in Al Makdah et al. (2022), Celi and Pasqualetti (2022), De Persis and Tesi (2020) and Bianchin (2023), Celi et al. (2023a), which solve the general eigenstructure assignment problem via static feedback in a purely data-driven manner.

6. Conclusion and future work

This tutorial paper presents a framework to solve a variety of linear quadratic control problems for linear systems using a dataset of input, state, and output trajectories collected offline, and to analyze the robustness of these solutions to noise and arbitrary perturbations. Differently from approaches relying on system identification, optimization, and policy iteration, this paper presents a set of closed-form expressions for optimal and sub-optimal control sequences, which are computationally efficient, insightful, and enable a direct sensitivity analysis of data-driven control. In fact, in some cases, these formulas display favorable computational properties even when compared to classic model-based solutions, while also avoiding the solution of typically implicit and recursive Riccati equations. The paper contains also a number of numerical studies to validate the approach and showcase its tradeoffs.

This paper shows that the data-driven approach to control may offer solutions that are computationally advantageous with respect to classic methods in the state–space, frequency, or geometric approaches. However, a detailed analysis of when data-driven methods should be preferred to classic ones, a comprehensive comparison of direct and indirect methods, the extension beyond linear quadratic control, among others, remain outstanding timely questions and potentially interesting research avenues.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Appendix

A.1. Proof of the results in Section 2

We start with the following technical result.

Lemma A.1 (Rank of Block Matrices). Let $A \in \mathbb{R}^{n_1 \times m}$ and $B \in \mathbb{R}^{n_2 \times m}$, with $m \geq n_1 + n_2$. Let $K_A = \text{Basis}(\text{Ker}(A))$ and $K_B = \text{Basis}(\text{Ker}(B))$. The following statements are equivalent:

(i) $\text{Rank} \left( \begin{bmatrix} A \\ B \end{bmatrix} \right) = n_1 + n_2$;

(ii) $\text{Rank}(AK_B) = n_1$ and $\text{Rank}(BK_A) = n_2$.

Proof. ((ii) implies (i)) We will show that statement (ii) is violated when statement (i) is violated. Let vectors $v_A$ and $v_B$ satisfy $v_A A + v_B B = 0$, with $[v_A, v_B] \neq 0$. Then, $v_A A K_B + v_B B K_B = v_A A K_B = 0$, which implies that either $v_A = 0$ or $\text{Rank}(AK_B) < n_1$. Similarly, $v_A A K_A + v_B B K_A = v_B B K_A = 0$, which implies that either $v_B = 0$ or $\text{Rank}(BK_A) < n_2$. Since $v_A$ and $v_B$ cannot be simultaneously zero, we conclude that statement (ii) implies (i).

((i) implies (ii)) Notice that $\text{Rank} \left( \begin{bmatrix} A \\ B \end{bmatrix} \right) = \text{Rank} \left( \begin{bmatrix} A_C \\ K_B \\ B \end{bmatrix} \right)$. with $C$ any invertible matrix of appropriate dimension. Let $C = [K_B \ B^T]$. Then,

\[
\begin{align*}
n_1 + n_2 &= \text{Rank} \left( \begin{bmatrix} A \\ B \end{bmatrix} \right) = \text{Rank} \left( \begin{bmatrix} A \\ K_B \\ B^T \end{bmatrix} \right) \\
&= \text{Rank} \left( \begin{bmatrix} AK_B \\ AB^T \\ 0 \\ BB^T \end{bmatrix} \right),
\end{align*}
\]

which implies that $\text{Rank}(AK_B) = n_1$. Repeating the reasoning with $C = [K_A \ A^T]$ concludes the proof.

We are now ready to prove Lemma 2.1.
Proof of Lemma 2.1. Using (4) and Lemma A.1, the matrices $X_\delta K_U$ and $U K_\delta$ are full-row rank. Hence, for every $x_0$ and $u_\tau$ there exist $a$ and $\beta$ such that $x_0 = X_\delta K_U a$ and $u_\tau = U K_\delta \beta$. Notice that the data matrices satisfy the relations
\[
\begin{bmatrix}
X &= O_X T F_X^T [X_0]
\end{bmatrix},
\tag{35}
\]
where
\[
O_X^T = \begin{bmatrix}
A \\
A^2 \\
\vdots \\
A^T
\end{bmatrix},
\quad
F_X = \begin{bmatrix}
B & \cdots & 0 \\
AB & \cdots & 0 \\
\vdots & \ddots & \vdots \\
A^{T-1}B & \cdots & AB
\end{bmatrix}.
\]

Notice that
\[
X K_U = O_X^T X_\delta K_U + F_X^T U K_\delta = O_X X_0 K_U + F_X^T U K_0,
\]
\[
X K_\delta = O_X^T X_\delta K_\delta + F_\delta^T U K_\delta = O_X X_\delta K_\delta + F_\delta^T U K_0.
\]
Then, the state trajectory $x_T$ of (1) with input $u_\tau = U K_\delta \beta$ and initial state $x_0 = X_\delta K_U a$ can be written as
\[
x_T = O_X T x_0 + F_X T u_\tau = O_X X_0 K_U a + F_X^T U K_\delta \beta
= X K_U a + X K_\delta \beta.
\]
The claimed expression for $y_T$ is obtained similarly using the matrices $O^T_Y$ and $F_Y^T$, thus concluding the proof. \hfill \blacksquare

Proof of Lemma 2.2. Notice from (35) that $Y K_U = O_Y^T X_\delta K_U$. From Bernstein (2009, Fact 2.10.2), $\text{Im}(O_Y^T X_\delta K_U) = \text{Im}(O_Y^T)$ since $X_\delta K_U$ is full row-rank (cf. (4) and Lemma A.1). Hence, $\text{Rank}(O_Y^T) = \text{Rank}(Y K_U)$. Similarly, $X_\delta K_0 = C_T U K_0$ and $U K_\delta$ is full row-rank, thus implying that $\text{Rank}(C_T) = \text{Rank}(X_\delta K_0)$, which concludes the proof. \hfill \blacksquare

A.2. Proof of the results in Section 3

Proof of Eq. (12). Problem (8) can be equivalently rewritten as
\[
\begin{align*}
\text{minimize} & \quad \| p^2 x(0) u_T \|_2^2 \\
\text{subject to} & \quad \begin{bmatrix}
I \\
O_Y^T \\
F_Y^T
\end{bmatrix} x(0) u_T = \begin{bmatrix}
x_0 \\
u_T
\end{bmatrix},
\end{align*}
\tag{36}
\]
where $O_Y^T$ and $F_Y^T$ denotes the last $p$ rows of the matrices $O_Y^T$ and $F_Y^T$, and
\[
P = \begin{bmatrix} O_Y^T & F_Y^T \end{bmatrix}^T O_Y^T \begin{bmatrix} O_Y^T & F_Y^T \end{bmatrix} + \begin{bmatrix} 0 & 0 \end{bmatrix} R
\tag{37}
\]
From the assumptions $R > 0$ and $\text{Rank}(O_Y^T) = n$, it follows that $P > 0$ (see Celi et al. (2023b, Theorem III.1) for a proof of this fact). Then, by defining
\[
v = p^2 x(0) u_T,
\]
we can rewrite (36) as
\[
\begin{align*}
\text{minimize} & \quad \| v \|_2^2 \\
\text{subject to} & \quad \begin{bmatrix}
I \\
O_Y^T \\
F_Y^T
\end{bmatrix} p^2 v = \begin{bmatrix}
x_0 \\
u_T
\end{bmatrix}.
\end{align*}
\tag{38}
\]
The minimizer of (38) is
\[
v^* = \left( \begin{bmatrix} I & 0 \\
O_Y^T & F_Y^T
\end{bmatrix} \right)^{-1} \begin{bmatrix} x_0 \\
u_T
\end{bmatrix}. \tag{39}
\]
Thus, the solution $u^*_T$ to (36) satisfies
\[
\begin{bmatrix}
x_0 \\
u_T
\end{bmatrix} = p^{-2} v^* = p^{-2} \left( \begin{bmatrix} I & 0 \\
O_Y^T & F_Y^T
\end{bmatrix} \right)^{-1} \begin{bmatrix} x_0 \\
u_T
\end{bmatrix}.
\]
To conclude, note that, under assumption (4), $P$ in (37) equals (13) and
\[
\begin{bmatrix}
I_a & 0 \\
O_Y^T & F_Y^T
\end{bmatrix} = \begin{bmatrix} x_0 \\
u_T
\end{bmatrix} \begin{bmatrix} x_0 \\
u_T
\end{bmatrix}^T,
\]
from which (12) follows. \hfill \blacksquare

Proof of Lemma 3.3. From Lemma 2.1, there exist vectors $a$ and $\beta$, such that $x_0 = X_0 x_0 K_U a$ and $u_T = U K_0 u_T \beta$, and the corresponding state trajectory in the interval $[1, T_T]$ is
\[
x_{T_T} = X_{T_T} K_{U,T_T} a + X_{T_T} K_{0,T_T} \beta.
\]
In particular, it holds
\[
\text{tr}(x_{T_T}) = \text{tr}(X_{T_T} K_{U,T_T} a + X_{T_T} K_{0,T_T} \beta).
\]
Next, since $X_{0,T_T} K_{U,T_T}$ has full rank, we have
\[
\text{tr}(x_{T_T}) = X_{0,T_T} K_{U,T_T} r_T,
\]
where
\[
r_T = (X_{0,T_T} K_{U,T_T})^* (X_{T_T} K_{U,T_T} a + X_{T_T} K_{0,T_T} \beta)
= V_T + Z_T \beta.
\]
Thus, using Lemma 2.1 again, there exist a vector $\beta$ such that the input in the interval $[0, T_T + T_T - 1]$ can be written as
\[
\begin{bmatrix}
U_{T_T} \\
U_{0,T_T}
\end{bmatrix} = \begin{bmatrix} x_{T_T} & 0 \\
0 & \beta
\end{bmatrix},
\]
and the corresponding state trajectory in $[1, T_T + T_T]$ as
\[
x_{T_T + T_T} = \begin{bmatrix}
X_{T_T} K_{U,T_T} a + X_{T_T} K_{0,T_T} \beta \\
X_{0,T_T} K_{U,T_T} a + X_{0,T_T} K_{0,T_T} \beta
\end{bmatrix}
= \begin{bmatrix}
X_{T_T} K_{U,T_T} a + X_{T_T} K_{0,T_T} \beta \\
X_{0,T_T} K_{U,T_T} a + X_{0,T_T} K_{0,T_T} \beta
\end{bmatrix}
\]
\[
= \begin{bmatrix}
X_{T_T} K_{U,T_T} a \\
X_{0,T_T} K_{U,T_T} a
\end{bmatrix} \begin{bmatrix} \beta \\
\beta
\end{bmatrix}.
\]
The expression of $x_T$ in (21) follows by iterating the previous argument $p$ times and collecting all $\beta$, as $\beta = [\beta_1^T \cdots \beta_p^T]^T$. A similar reasoning holds for the output trajectory $y_T$. \hfill \blacksquare

A.3. Proof of the results in Section 4

Proof of Theorem 4.1. By the strong law of large numbers (Van der Vaart, 2000) and the assumptions (i)-(ii) on the experiments and noise, as $N$ grows, the entries of $\frac{1}{N} \Delta_0 a_U$, $\frac{1}{N} \Delta_U a_U$, $\frac{1}{N} U a_U$, $\frac{1}{N} U a_U^*$ tend to zero almost surely, while $\frac{1}{N} A a_U = \frac{1}{N} \Delta a_U$ tend to $\sigma_I^2 I$ almost surely. This implies that, as $N \to \infty$,
\[
\begin{bmatrix}
X_0 \\
U
\end{bmatrix} \to_{a.s.} \begin{bmatrix} \frac{1}{N} X_0 \\
\frac{1}{N} U
\end{bmatrix} = \begin{bmatrix}
X_0 \\
U
\end{bmatrix} + \begin{bmatrix}
\sigma^2 I \\
0
\end{bmatrix} \begin{bmatrix}
\sigma^2_{Ut} I \\
\sigma^2_{I}
\end{bmatrix}.
\tag{39}
\]
Similarly, by the strong law of large numbers, the entries of $\frac{1}{N} Y a_U^*$, $\frac{1}{N} Y a_U^*$, $\frac{1}{N} Y a_U^*$, $\frac{1}{N} a_U a_U^*$, $\frac{1}{N} a_U a_U^*$ tend to zero almost surely with $N$, so that, as $N \to \infty$,
\[
\begin{bmatrix}
Y \\
U
\end{bmatrix} \to_{a.s.} \begin{bmatrix}
\frac{1}{N} Y \\
\frac{1}{N} U
\end{bmatrix} = \begin{bmatrix}
Y \\
U
\end{bmatrix}.
\tag{40}
\]
Eqs. (39) and (40) imply that, as $N \to \infty$,
\[
\mathcal{L} \left[ X_0 \right] = \frac{1}{N} \mathcal{L} \left[ \frac{X_0}{U} \right] \left[ \frac{X_0}{U} \right] T = \left[ \sigma^2_0 I_n 0 \right] \left[ \sigma^2_0 I_n T \right] ^T = \left[ \sigma^2_0 I_n 0 \right] \left[ \sigma^2_0 I_n T \right] ^T
\]
\[
\rightarrow y \left[ X_0 \right] \left[ X_0 \right] ^T = y \left[ X_0 \right] ^T
\]

Since the map in (23) is a continuous function of $U$, $\chi_0$, $\chi$ at $\Delta_U = \Delta_y = 0$, from the previous equation and the continuous mapping theorem (Van der Vaart, 2000), as $N \to \infty$,
\[
\left[ \begin{array}{c}
\chi_0 \\
\u_t
\end{array} \right] = \sum_{\chi \in \chi(t)} \left( \begin{array}{c}
\chi_0 \\
\u_t
\end{array} \right) \left[ \begin{array}{c}
\Omega_F \\
\Omega_U
\end{array} \right] T
\]
where we used the identity $X_0 \left[ X_0 \right] ^T = [I_n 0]$, which holds when $X_0$ has full row rank, and the fact that (4) is satisfied almost surely as $N \to \infty$.}

The following result, whose proof follows from Anguluri et al. (2020, Lemma1), confirms that when the expected norm of the perturbation $\Delta_U$ is small so is the residual $r$ of the expansion (25).

\textbf{Lemma A.2 (First-Order Approximation of $F(U, X_0, Y)$).} Let $\mathcal{V}_{Y_j}$ be the $i$th column of $\mathcal{V}_Y$ and
\[
\Sigma = F(U, X_0, Y) - F(U, X_0, Y) - \sum_{i \in \text{supp}(\Delta_U)} \delta_{Y_i} \mathcal{V}_{Y_j}(U, X_0, Y).
\]

Then, for any $\tau > 0$,
\[
\lim_{\Sigma \to 0} \mathbb{E}[\Sigma] = \tau \mathbb{E}[\Sigma] \mathbb{E}[\Sigma] = 0.
\]

\textbf{Proof of Lemma 4.2.} Let
\[
G = \left[ \begin{array}{c}
X_0 \\
Y_f \\
0
\end{array} \right]
\]
and rewrite (24) as
\[
F(U, X_0, Y) = P^{i \cdot \frac{1}{2}} \left( GP^{i \cdot \frac{1}{2}} \right) \left[ \begin{array}{c}
X_0 \\
Y_i
\end{array} \right] = P^{i \cdot G^T (GP^{i \cdot G})^{-1} X_0 \left[ Y_i \right]}
\]
Let $y_i$ denote the $i$th element of $\text{vec}(Y)$, it holds
\[
\mathcal{V}_{Y_j} = \frac{\partial F(U, X_0, Y)}{\partial y_i} = \frac{\partial P^{-1} G^T (GP^{-1} G)^{-1} X_0 \left[ Y_i \right]}{\partial y_i}
\]
\[
= \left[ \begin{array}{c}
Q_F \\
Q_P
\end{array} \right] T \left[ \begin{array}{c}
Q_F \\
Q_P
\end{array} \right] T
\]
\[
= \left[ \begin{array}{c}
\Omega_F \\
\Omega_P
\end{array} \right] T \left[ \begin{array}{c}
\Omega_F \\
\Omega_P
\end{array} \right] T
\]
\[
\frac{\partial}{\partial y_i} \left[ \begin{array}{c}
Q_F \\
Q_P
\end{array} \right] T \left[ \begin{array}{c}
Q_F \\
Q_P
\end{array} \right] T = \left[ \begin{array}{c}
\Omega_F \\
\Omega_P
\end{array} \right] T \left[ \begin{array}{c}
\Omega_F \\
\Omega_P
\end{array} \right] T
\]
Notice that
\[
\frac{\partial}{\partial y_i} \left[ \begin{array}{c}
Q_F \\
Q_P
\end{array} \right] T \left[ \begin{array}{c}
Q_F \\
Q_P
\end{array} \right] T = \left[ \begin{array}{c}
\Omega_F \\
\Omega_P
\end{array} \right] T \left[ \begin{array}{c}
\Omega_F \\
\Omega_P
\end{array} \right] T
\]
\[
= \left[ \begin{array}{c}
\Omega_F \\
\Omega_P
\end{array} \right] T \left[ \begin{array}{c}
\Omega_F \\
\Omega_P
\end{array} \right] T
\]
\[
= \left[ \begin{array}{c}
\Omega_F \\
\Omega_P
\end{array} \right] T \left[ \begin{array}{c}
\Omega_F \\
\Omega_P
\end{array} \right] T
\]
\[
= \left[ \begin{array}{c}
\Omega_F \\
\Omega_P
\end{array} \right] T \left[ \begin{array}{c}
\Omega_F \\
\Omega_P
\end{array} \right] T
\]
\[
= \left[ \begin{array}{c}
\Omega_F \\
\Omega_P
\end{array} \right] T \left[ \begin{array}{c}
\Omega_F \\
\Omega_P
\end{array} \right] T
\]
\[
= \left[ \begin{array}{c}
\Omega_F \\
\Omega_P
\end{array} \right] T \left[ \begin{array}{c}
\Omega_F \\
\Omega_P
\end{array} \right] T
\]
\[
= \left[ \begin{array}{c}
\Omega_F \\
\Omega_P
\end{array} \right] T \left[ \begin{array}{c}
\Omega_F \\
\Omega_P
\end{array} \right] T
\]
\[
= \left[ \begin{array}{c}
\Omega_F \\
\Omega_P
\end{array} \right] T \left[ \begin{array}{c}
\Omega_F \\
\Omega_P
\end{array} \right] T
\]
\[
= \left[ \begin{array}{c}
\Omega_F \\
\Omega_P
\end{array} \right] T \left[ \begin{array}{c}
\Omega_F \\
\Omega_P
\end{array} \right] T
\]
\[
= \left[ \begin{array}{c}
\Omega_F \\
\Omega_P
\end{array} \right] T \left[ \begin{array}{c}
\Omega_F \\
\Omega_P
\end{array} \right] T
\]
\[
= \left[ \begin{array}{c}
\Omega_F \\
\Omega_P
\end{array} \right] T \left[ \begin{array}{c}
\Omega_F \\
\Omega_P
\end{array} \right] T
\]
\[
= \left[ \begin{array}{c}
\Omega_F \\
\Omega_P
\end{array} \right] T \left[ \begin{array}{c}
\Omega_F \\
\Omega_P
\end{array} \right] T
\]
Proof of Theorem 4.3. By definition of $\Delta y_t$, 
\[
\Delta y_t = \|y_t - y_{t-1}\|_2 = \sum_{i \in \text{supp}(\Delta y_t)} \|\Delta y_{t,i}\|_{\mathbb{R}},
\]
\[
\leq \sum_{i \in \text{supp}(\Delta y_t)} |\Delta y_{t,i}| \|\Omega_{Y} F_T^{T} \| \| F_{Y,t} \|_2 = \Delta y_t.
\]
By the monotonicity of probability measures, for any $t > 0$, the set inclusion $\{y_t \geq r\} \subseteq \{\overline{y}_t \geq r\}$ holds, which implies $P[\Delta y_t \geq r] \leq P[\overline{y}_t \geq r]$. Note that $|\Delta y_{t,i}|$ are non-negative random variables. Thus, by Markov's inequality (Van der Vaart, 2000) and the linearity of the expected value for any $t > 0$, 
\[
P[\Delta y_t \geq r] \leq \mathbb{E}[\overline{y}_t \geq r] \leq \frac{1}{r} \sum_{i \in \text{supp}(\Delta y_t)} \mathbb{E}[|\Delta y_{t,i}|] \leq \frac{\sum_{i \in \text{supp}(\Delta y_t)} \mathbb{E}[|\Delta y_{t,i}|]}{r} (52)
\]
\[
\leq \frac{1}{r} \sum_{i \in \text{supp}(\Delta y_t)} \mathbb{E}[|\Delta y_{t,i}|] \leq \frac{c}{r} \sum_{i \in \text{supp}(\Delta y_t)} \mathbb{E}[|\Delta y_{t,i}|] \leq \frac{c}{r} \sum_{i \in \text{supp}(\Delta y_t)} \max\{\|\Omega_{Y} F_T^{T} \|, \| F_{Y,t} \|_2\} \max\{\mathbb{E}[|\Delta y_{t,i}|]\} (53)
\]
where $c = \|\Omega_{Y} F_T^{T} \|_2$ and $|\text{supp}(\Delta y_t)|$ stands for the cardinality of supp($\Delta y_t$). Since the distributions of $\Delta x_t$ are independent of $N$ so are $\mathbb{E}[|\Delta y_{t,i}|]$. Hence, by (53) and Lemma 4.2, it follows that 
\[
P[\Delta y_t \geq r] \leq \frac{k_T}{r} |\text{supp}(\Delta y_t)| \frac{N}{N} (54)
\]
where $k_T > 0$ is a constant independent of $N$. This concludes the proof.

A.4. Proof of the results in Section 5

Proof of Theorem 5.1. First, we provide a data-driven solution to the problem in (28). Following a procedure similar to the proof of Theorem 3.1, Problem (28) can be written as in (10), with 
\[
\gamma = \beta, \quad L = \frac{Q^{1/2}X}{R^{1/2}U}, \quad W = X_0 K, \quad \text{and} \quad z = x_0.
\]
Then, the optimal input and state trajectories of (28) for a given $x_0$ can be computed as 
\[
u^*_t = U K(I_N - K_W L K_W^T) L W^T x_0,
\]
\[
x^*_t = X K(I_N - K_W L K_W^T) L W^T x_0.
\]
From (29) we know that the time-varying controller that solves Problem (28) satisfies $u(t) = K^T(t)$, and it does not depend on the initial condition. Let $U_t$ and $X_t$ be as in (31). Notice that the $i$th column of $X_t$ equals the state at time $t$ of the optimal state trajectory for (28) with initial state given by the $i$th column of the identity matrix. Then, $U_t = K^T(t)X_t$. To conclude, we show that if $A$ is invertible then $X_t$ is invertible. To this end, let $A_t = A + BK^T(t)$ denote the (time-varying) state matrix of the closed-loop system and notice that 
\[
A_t = (I - B(R_t + B^T P_{t+1} B)^{-1} B^T P_{t+1}) A_t (55)
\]
where $P_t \geq 0$ satisfies the Riccati equation (30). Using the push-through identity $(I + XY)^{-1} X = (I + YX)^{-1}$, which holds for any matrices $X, Y$ such that $(I + XY)$ is invertible (Bernstein, 2009, Fact 2.16.16), $H_t$ can be written as 
\[
H_t = I - B(R_t + B^T P_{t+1} B)^{-1} B^T P_{t+1}
\]
\[
= I - B R_t^{-1}(I + B^T P_{t+1} B R_t^{-1})^{-1} B^T P_{t+1}
\]
\[
= I - B R_t^{-1} B^T P_{t+1} (I + B R_t^{-1} B^T P_{t+1})^{-1}.
\]
From the last identity it follows that 
\[
H_t(I + B R_t^{-1} B^T P_{t+1}) = I,
\]
which implies that $H_t$ is invertible with inverse $H_t^{-1} = I + B R_t^{-1} B^T P_{t+1}$. Finally, observe that, for $t \geq 1$, 
\[
X_t = A_{t-1} A_{t-2} \ldots A_0 H_{t-1} A_{t-2} \ldots H_0 A.
\]
Since the product of invertible matrices is invertible, it follows that if $A$ is invertible then $X_t$ is invertible.

References


