An efficient data-based off-policy Q-learning algorithm for optimal output feedback control of linear systems

Mohammad Alsalti   ALSALTI@IRT.UNI-HANNOVER.DE
Victor G. Lopez   LOPEZ@IRT.UNI-HANNOVER.DE
Matthias A. Müller   MUELLER@IRT.UNI-HANNOVER.DE
Leibniz University Hannover, Institute of Automatic Control, Hannover 30167, Germany

Abstract

In this paper, we present a Q-learning algorithm to solve the optimal output regulation problem for discrete-time LTI systems. This off-policy algorithm only relies on using persistently exciting input-output data, measured offline. No model knowledge or state measurements are needed and the obtained optimal policy only uses past input-output information. Moreover, our formulation of the proposed algorithm renders it computationally efficient. We provide conditions that guarantee the convergence of the algorithm to the optimal solution. Finally, the performance of our method is compared to existing algorithms in the literature.

Keywords: Data-based control, optimal output regulation, Q-learning, reinforcement learning.

1. Introduction

Reinforcement learning (RL) (Sutton and Barto, 2018) is a suitable framework for learning optimal control policies for unknown dynamical systems (Bertsekas, 2012). One of the most investigated problems in this area is the problem of optimal control of discrete-time linear time-invariant systems (LTI), also referred to as linear quadratic regulator (LQR) problem. For this problem, a control policy is sought such that the (weighted) deviation of the states from the origin and the control expenditure are minimized. Among many works that have addressed this problem in the RL literature are, e.g., (Bradtke et al., 1994; Kiumarsi et al., 2017; Fazel et al., 2018; Lopez et al., 2023).

When full state information is not available, the problem of optimal output regulation seeks to find a control policy such that the deviation of the outputs from the origin and the control expenditure are minimized. This problem has been addressed using RL approaches in, e.g., (Lewis and Vamvoudakis, 2010; Kiumarsi et al., 2015; Rizvi and Lin, 2019). These works present *on-policy* algorithms, i.e., they require the application of each new policy to the system at each iteration. However, it is known that such algorithms lead, in general, to biased solutions due to the introduction of probing signals at each iteration, see, e.g., (Lewis and Vamvoudakis, 2010; Kiumarsi et al., 2015).

Addressing optimal output regulation using *off-policy* algorithms was done in (Jiang et al., 2020), but this work required state measurements for the learning process. In (Zhang et al., 2023), an off-policy algorithm for solving the H_∞ optimal output regulation problem was presented. The proposed method requires additional constraints on the class of systems under consideration, specifically that the system is stabilizable by static output-feedback (see, e.g., Kučera and De Souza (1995)), thus restricting its applicability. To solve the Bellman equation, both works mentioned above propose to use the batch least squares method, which becomes computationally inefficient for systems with high orders and large number of inputs. Finally, in those works it is unclear how to

excite the system such that persistence of excitation conditions are met, or how to choose an initial stabilizing policy.

Apart from RL, other approaches have been recently proposed to solve the data-based optimal output regulation problem based on semi-definite programming (SDP) (cf. Dai et al. (2023)), which make use of data-based representations of LTI systems (see, e.g., Willems et al. (2005)). However, such SDPs are computationally expensive for large system dimensions.

Contribution: In this work, we present an efficient *off-policy* Q-learning algorithm to solve the optimal output regulation problem. The proposed algorithm does not use model knowledge or state information and does not require that the system is stabilizable by static output-feedback. Instead, we use input-output data that are collected offline only once from the system. The resulting optimal policy is an output-feedback controller which uses only past input-output information. Our formulation relies on a suitable definition of a non-minimal state of the system which, along with an easily enforced persistence of excitation condition, renders the algorithm to be highly computationally efficient. We further show that the proposed algorithm is guaranteed to converge to the optimal gain at a quadratic rate of convergence, and a data-based method to design an initial stabilizing policy is given. This work serves as an extension of the algorithm presented in (Lopez et al., 2023) for the LQR problem with state information.

Paper structure: Section 2 formulates the problem and recalls existing results. Section 3 includes the main contribution of this paper. Section 4 contains numerical examples and a comparison to existing methods. Finally, Section 5 discusses the results and concludes the paper.

Notation: We use I_m to denote an $m \times m$ identity matrix and 0 to denote a zero matrix of appropriate dimensions. A positive (semi-)definite matrix is denoted by $M \succ 0$ ($M \succeq 0$). For a sequence $\{s_k\}_{k=0}^{N-1}$ with $s_k \in \mathbb{R}^\eta$, we denote its stacked vector as $s = \begin{bmatrix} s_0^\top & s_1^\top & \ldots & s_{N-1}^\top \end{bmatrix}^\top$ and a stacked window of it as $s_{[l,j]} = \begin{bmatrix} s_l^\top & s_{l+1}^\top & \ldots & s_j^\top \end{bmatrix}^\top$ for $0 \le l < j$. The following definition of persistence of excitation of a sequence is used throughout the paper.

Definition 1 (Willems et al. (2005)) *A sequence s is said to be persistently exciting (PE) of order L if $\text{rank}(\mathscr{H}_L(s)) = \eta L$, where the Hankel matrix $\mathscr{H}_L(s)$ takes the form*

$$\mathscr{H}_L(s) = \begin{bmatrix} s_{[0,L-1]} & s_{[1,L]} & \cdots & s_{[T-L,T-1]} \end{bmatrix} \tag{1}$$

In this work, we consider discrete-time LTI systems of the following form

$$x_{k+1} = Ax_k + Bu_k, \qquad y_k = Cx_k, \tag{2}$$

where $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^m$, $y_k \in \mathbb{R}^p$ are the state, input and output vectors. The lag ℓ of the system (observability index) is defined as the smallest integer j such that the matrix $\mathscr{O}_j = \begin{bmatrix} C^\top & (CA)^\top & \cdots & (CA^{j-1})^\top \end{bmatrix}^\top$ has rank n. Note that the following relationship holds $\ell \le n \le p\ell$ (Willems, 1986). The matrices (A, B, C) are unknown but assumed to correspond to some minimal state-space realization and, hence, the pair (A, B) is controllable and the pair (A, C) is observable.

2. Problem formulation

We consider the optimal output regulation problem of the form

$$\min_{\{u_0, u_1, \cdots\}} \sum_{k=0}^{\infty} c(y_k, u_k) = \min_{\{u_0, u_1, \cdots\}} \sum_{k=0}^{\infty} y_k^\top Q_y y_k + u_k^\top R u_k, \tag{3}$$

with $Q_y, R \succ 0$. Notice that the stage cost can also be written in terms of the state x_k using the output equation in (2). This results in a standard LQR problem with $c(y_k, u_k) = x_k^\top Q_x x_k + u_k^\top R u_k$, where $Q_x := C^\top Q_y C \succeq 0$ is, in general, a positive semi-definite matrix with $\sqrt{Q_x} = \sqrt{Q_y} C$. Since the pair (A, C) is observable and since $Q_y \succ 0$, the pair $(A, \sqrt{Q_x})$ is also observable. This, along with controllability of (A, B), implies the existence and uniqueness of a solution to the LQR problem of the form $u_k = -K_x^* x_k$, where

$$K_x^* = (R + B^\top P B)^{-1} B^\top P A, \tag{4}$$

and $P \succ 0$ is the unique solution of the discrete algebraic Riccati equation (Lewis et al., 2012). Since the matrices (A, B, C) are unknown, one cannot evaluate the expression in (4) and, moreover, since state measurements are not available, one cannot implement the policy $u_k = -K_x^* x_k$.

In (Alsalti et al., 2023), it was shown that a non-minimal state $z_k \in \mathbb{R}^{m\ell+n}$ for system (2) can be suitably constructed from past inputs and outputs[1]. Such a non-minimal state takes the form

$$z_k = \begin{bmatrix} u_{[k-\ell,k-1]} \\ \Gamma\, y_{[k-\ell,k-1]} \end{bmatrix} \in \mathbb{R}^{m\ell+n}, \tag{5}$$

where $\Gamma \in \mathbb{R}^{n \times p\ell}$ is a matrix that is computed from data as summarized in the following remark.

Remark 2 (Alsalti et al. (2023)) *Let $\{u_k, y_k\}_{k=-\ell}^{N-1}$ be input-output data collected from (2) with the input being PE of order $\ell + n + 1$. Arrange the data in a Hankel matrix of the form $\begin{bmatrix} \mathscr{H}_\ell(u_{[-\ell,N-2]}) \\ \mathscr{H}_\ell(y_{[-\ell,N-2]}) \end{bmatrix}$ which, by minimality of the system and PE of the input, is guaranteed to have rank $m\ell + n$ (cf. Willems et al. (2005)). One can now select $m\ell + n$ linearly independent rows of this matrix and arrange them in a matrix $Z_0 \in \mathbb{R}^{m\ell+n \times N}$, while the remaining rows are arranged in $\Phi_0 \in \mathbb{R}^{p\ell-n \times N}$. Next, one can find a permutation matrix $\Pi \in \mathbb{R}^{p\ell \times p\ell}$ which makes the following equation hold*

$$\begin{bmatrix} \mathscr{H}_\ell(u_{[-\ell,N-2]}) \\ \mathscr{H}_\ell(y_{[-\ell,N-2]}) \end{bmatrix} = \begin{bmatrix} I_{m\ell} & 0 \\ 0 & \Pi \end{bmatrix} \begin{bmatrix} Z_0 \\ \Phi_0 \end{bmatrix}. \tag{6}$$

Finally, the matrix $\Gamma \in \mathbb{R}^{n \times p\ell}$ can be found as follows $\Pi^{-1} = \begin{bmatrix} \Gamma^\top & G^\top \end{bmatrix}^\top$, where $G \in \mathbb{R}^{p\ell-n \times p\ell}$.

The minimal state x of the system in (2) and the non-minimal state z in (5) are related by $x_k = T z_k$, where T is a full row rank matrix (see (Alsalti et al., 2023, Thm. 4)) that depends on the unknown matrices (A, B, C). Using $x_k = T z_k$, one can express the optimal solution to (3) as

$$u_k = -K_x^* x_k = -K_x^* T z_k =: -K_z^* z_k, \tag{7}$$

which is a control law that can be implemented, provided that the value of $K_z^* \in \mathbb{R}^{m \times m\ell+n}$ is found, since z_k only consists of past inputs and outputs (cf. (5)). Notice that, given a non-minimal state z_k, the matrix K_z^* is unique.

In (Lopez et al., 2023), an efficient off-policy Q-learning algorithm was proposed to solve the LQR problem, assuming availability of state measurements. In this paper, we only consider availability of input-output data and, hence, the results of (Lopez et al., 2023) are not directly applicable.

---

1. Although other non-minimal state definitions can be made (cf. (Goodwin and Sin, 2014, Section 3.4)), the non-minimal state proposed in (Alsalti et al., 2023) ensures the satisfaction of certain rank conditions which facilitate data-driven design of stabilizing controllers.

The goal of this work is to extend the results of (Lopez et al., 2023) to the case of output measurements and provide an efficient off-policy algorithm that converges to K_z^* in (7) (thus solving (3)), while maintaining the same uniqueness and convergence properties of (Lopez et al., 2023).

Before reviewing the results of (Lopez et al., 2023) in Section 2.1, we first recall the following lemma, which provides conditions on the input such that certain rank conditions on the matrix of input and non-minimal state data are satisfied. This lemma is important for the results of Section 3.

Lemma 3 (Alsalti et al. (2023)) *Let $\{u_k, y_k\}_{k=-\ell}^{N-1}$ be input-output data collected from system (2) with the input being PE of order $\ell + n + 1$. Then* $\mathrm{rank}\left(\begin{bmatrix} Z_0 \\ U_0 \end{bmatrix}\right) = m(\ell+1) + n$, where

$$
\begin{bmatrix} Z_0 \\ U_0 \end{bmatrix} = \begin{bmatrix} z_0 & z_1 & \cdots & z_{N-1} \\ u_0 & u_1 & \cdots & u_{N-1} \end{bmatrix} \stackrel{(5)}{=} \begin{bmatrix} I_{m\ell} & 0 & 0 \\ 0 & \Gamma & 0 \\ 0 & 0 & I_m \end{bmatrix} \begin{bmatrix} u_{[-\ell,-1]} & u_{[-\ell+1,0]} & \cdots & u_{[N-\ell-1,N-2]} \\ y_{[-\ell,-1]} & y_{[-\ell+1,0]} & \cdots & y_{[N-\ell-1,N-2]} \\ u_0 & u_1 & \cdots & u_{N-1} \end{bmatrix}. \tag{8}
$$

2.1. Q-learning algorithm for the LQR problem

In (Lopez et al., 2023), an efficient off-policy Q-learning algorithm was proposed to solve the LQR problem, assuming availability of state measurements. At each iteration of the algorithm, the value function $V^{(i)}(x_k) = x_k^\top P^{(i)} x_k$, for $P^{(i)} \succ 0$, evaluates the cost of using a particular control input of the form $u = -K_x^{(i)} x$ from time k onward. Similarly, a Q-function evaluates the cost of taking an arbitrary action at time k, then using the policy $u = -K_x^{(i)} x$ from time $k+1$ onward

$$
\mathcal{Q}^{(i+1)}(u_k, x_k) = \chi_k^\top \Theta_x^{(i+1)} \chi_k, \tag{9}
$$

where $\chi_k := \begin{bmatrix} x_k^\top & u_k^\top \end{bmatrix}^\top$ and

$$
\Theta_x^{(i+1)} = \begin{bmatrix} Q_x + A^\top P^{(i)} A & A^\top P^{(i)} B \\ B^\top P^{(i)} A & R + B^\top P^{(i)} B \end{bmatrix} =: \begin{bmatrix} \Theta_{xx}^{(i+1)} & (\Theta_{ux}^{(i+1)})^\top \\ \Theta_{ux}^{(i+1)} & \Theta_{uu}^{(i+1)} \end{bmatrix}. \tag{10}
$$

The core of the proposed algorithm in (Lopez et al., 2023, Alg. 1) consists of two parts: First, the solution of the Bellman equation

$$
\chi_k^\top \Theta_x^{(i+1)} \chi_k = \chi_k^\top \overline{Q} \chi_k + \begin{bmatrix} x_{k+1}^\top & (-K_x^{(i)} x_{k+1})^\top \end{bmatrix}^\top \Theta_x^{(i+1)} \begin{bmatrix} x_{k+1} \\ -K_x^{(i)} x_{k+1} \end{bmatrix}, \tag{11}
$$

where $\overline{Q} := \mathrm{diag}(Q_x, R)$ and second, the policy improvement step. Specifically, an improved policy $K_x^{(i+1)}$ at the next iteration (in the sense that $V^{(i+1)}(x_k) \leq V^{(i)}(x_k)$) can be obtained using

$$
K_x^{(i+1)} = (\Theta_{uu}^{(i+1)})^{-1} \Theta_{ux}^{(i+1)}. \tag{12}
$$

Equations (11) and (12) were used to construct an iterative off-policy algorithm that converges to the solution (4) of the LQR problem. Due to space constraints, we refer the reader to (Lopez et al., 2023) for more details on the construction of this algorithm. There, it was shown that solving the Bellman equation (11) is equivalent to solving a discrete-time Lyapunov equation of the form

$$
\Theta_x^{(i+1)} = \overline{Q} + \Phi_i^\top \Theta_x^{(i+1)} \Phi_i, \qquad \text{where } \Phi_i := \begin{bmatrix} A & B \\ -K_x^{(i)} A & -K_x^{(i)} B \end{bmatrix}. \tag{13}
$$

4

In (Lopez et al., 2023), only positive definite matrices $\overline{Q}$ were considered. Clearly, if Φ_i is Schur stable, then for any $\overline{Q} \succ 0$ there exists a unique and positive definite solution $\Theta_x^{(i+1)} \succ 0$ to (13). In the following lemma, the case where $\overline{Q} \succeq 0$ is studied.

Lemma 4 *If $K_x^{(i)}$ is a stabilizing gain, then for $\overline{Q} = \mathrm{diag}(Q_x, R) \succeq 0$ (with $Q_x = C^\top Q_y C$ and $Q_y \succ 0$) there exists a unique and positive definite solution $\Theta_x^{(i+1)} \succ 0$ to (13).*

Proof Equation (13) is a discrete-time Lyapunov equation that is known to have a unique and positive definite solution $\Theta^{(i+1)} \succ 0$ for any $\overline{Q} \succeq 0$ if (i) Φ_i is Schur stable and (ii) $(\Phi_i, \sqrt{\overline{Q}})$ is observable (Lewis et al., 2012, p. 37). Since $K_x^{(i)}$ is stabilizing, then matrix Φ_i is Schur stable (cf. (Lopez et al., 2023, Lemma 2)). To show observability of the pair $(\Phi_i, \sqrt{\overline{Q}})$, it suffices to show that the pair $\left(\Phi_i, \begin{bmatrix} C & 0 \\ 0 & \sqrt{R} \end{bmatrix}\right)$ is observable. This is because $\sqrt{\overline{Q}}$ can be expressed as

$$\sqrt{\overline{Q}} = \begin{bmatrix} \sqrt{Q_x} & 0 \\ 0 & \sqrt{R} \end{bmatrix} = \begin{bmatrix} \sqrt{Q_y}C & 0 \\ 0 & \sqrt{R} \end{bmatrix} = \begin{bmatrix} \sqrt{Q_y} & 0 \\ 0 & I_m \end{bmatrix} \begin{bmatrix} C & 0 \\ 0 & \sqrt{R} \end{bmatrix}. \tag{14}$$

Since the matrix $\mathrm{diag}(\sqrt{Q_y}, I_m)$ is invertible, the rank of the observability matrix of the pair $(\Phi_i, \sqrt{\overline{Q}})$ is equal to the rank of the observability matrix of $\left(\Phi_i, \begin{bmatrix} C & 0 \\ 0 & \sqrt{R} \end{bmatrix}\right)$. To show observability of the latter, first recall that observability is preserved under invertible coordinate transformations. Now consider a transformation $T_\phi = \begin{bmatrix} I_n & 0 \\ -K_x^{(i)} & I_m \end{bmatrix}$ such that $T_\phi^{-1}\Phi_i T_\phi = \begin{bmatrix} A - BK_x^{(i)} & B \\ 0 & 0 \end{bmatrix}$ and $\begin{bmatrix} C & 0 \\ 0 & \sqrt{R} \end{bmatrix} T_\phi = \begin{bmatrix} C & 0 \\ -\sqrt{R}K_x^{(i)} & \sqrt{R} \end{bmatrix}$. By the Popov–Belevitch–Hautus rank test (see Kailath (1980)), the pair $\left(\begin{bmatrix} A - BK_x^{(i)} & B \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} C & 0 \\ -\sqrt{R}K_x^{(i)} & \sqrt{R} \end{bmatrix}\right)$ is observable if and only if

$$\mathrm{rank}\,(\Omega_\lambda) = n + m, \quad \forall \lambda \in \mathbb{C}, \qquad \text{where } \Omega_\lambda := \begin{bmatrix} \lambda I - A + BK_x^{(i)} & -B \\ 0 & \lambda I \\ C & 0 \\ -\sqrt{R}K_x^{(i)} & \sqrt{R} \end{bmatrix}. \tag{15}$$

For all $\lambda \neq 0$, (15) holds since the pair $\left(A - BK_x^{(i)}, \begin{bmatrix} C \\ -\sqrt{R}K_x^{(i)} \end{bmatrix}\right)$ is observable (for any $K_x^{(i)}$) which follows from observability of the pair (A, C) (Lewis et al., 2012, p. 72). For $\lambda = 0$, suppose for contradiction that $\mathrm{rank}(\Omega_0) < n + m$. Then, there exists $r = \begin{bmatrix} r_1^\top & r_2^\top \end{bmatrix}^\top \neq 0$, with $r_1 \in \mathbb{R}^n$ and $r_2 \in \mathbb{R}^m$, such that $\Omega_0 r = 0$. This implies that $Cr_1 = 0$, $K_x^{(i)} r_1 = r_2$ and that $Ar_1 = BK_x^{(i)} r_1 - Br_2$. The last two equalities together imply that $Ar_1 = 0$, which together with $Cr_1 = 0$ and the observability of the pair (A, C), imply that $r_1 = 0$. Finally, since $K_x^{(i)} r_1 = r_2$ it follows that $r_2 = 0$, which contradicts the assumption that $r \neq 0$. Therefore, (15) holds and the pair $\left(\begin{bmatrix} A - BK_x^{(i)} & B \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} C & 0 \\ -\sqrt{R}K_x^{(i)} & \sqrt{R} \end{bmatrix}\right)$ is observable. ∎

In the following section, we extend the results of (Lopez et al., 2023) to the case of output measurements and provide an efficient off-policy Q-learning algorithm that converges to K_z^* in (7) (thus solving (3)), while maintaining the same uniqueness and convergence properties of (Lopez et al., 2023), which is the main contribution of the paper.

3. Q-learning algorithm for the optimal output regulation problem

As discussed in Section 2, the solution of the output regulation problem (3) has an equivalent parameterization in terms of the non-minimal state z (see (7)). In this section, we exploit the state transformation $x_k = Tz_k$ in order to develop an iterative algorithm which asymptotically converges to the optimal solution of (3) using only input-output data. First, notice that the following holds

$$\chi_k = \begin{bmatrix} x_k \\ u_k \end{bmatrix} = \begin{bmatrix} T & 0 \\ 0 & I_m \end{bmatrix} \begin{bmatrix} z_k \\ u_k \end{bmatrix} =: \mathscr{T}\zeta_k, \tag{16}$$

where $\zeta_k = [z_k^\top \ u_k^\top]^\top \in \mathbb{R}^{m(\ell+1)+n}$ and $\mathscr{T} \in \mathbb{R}^{n+m\times m(\ell+1)+n}$ is a full row rank matrix. This allows us to express the Q-function defined in (9) as follows

$$\mathcal{Q}^{(i+1)}(u_k, x_k) = \chi_k^\top \Theta_x^{(i+1)} \chi_k \overset{(16)}{=} \zeta_k^\top \mathscr{T}^\top \Theta_x^{(i+1)} \mathscr{T}\zeta_k =: \zeta_k^\top \Theta_z^{(i+1)} \zeta_k, \tag{17}$$

where

$$\Theta_z^{(i+1)} = \mathscr{T}^\top \Theta_x^{(i+1)} \mathscr{T} = \begin{bmatrix} T^\top \Theta_{xx}^{(i+1)} T & T^\top (\Theta_{ux}^{(i+1)})^\top \\ \Theta_{ux}^{(i+1)} T & \Theta_{uu}^{(i+1)} \end{bmatrix} =: \begin{bmatrix} \Theta_{zz}^{(i+1)} & (\Theta_{uz}^{(i+1)})^\top \\ \Theta_{uz}^{(i+1)} & \Theta_{uu}^{(i+1)} \end{bmatrix}, \tag{18}$$

with $\Theta_{zz} \in \mathbb{R}^{m\ell+n\times m\ell+n}, \Theta_{uz} \in \mathbb{R}^{m\times m\ell+n}$ and $\Theta_{uu} \in \mathbb{R}^{m\times m}$. Recall that $\Theta_x^{(i+1)}$ is unique and positive definite at each iteration (see Lemma 4). In contrast, $\Theta_z^{(i+1)}$ is unique but only positive semi-definite. This is due to the definition of $\Theta_z^{(i+1)}$ where $\Theta_x^{(i+1)}$ is pre- and post-multiplied by a full column and a full row rank matrix ($\mathscr{T}^\top$ and $\mathscr{T}$, respectively). Taking a closer look at the submatrices in (10) and (18), particularly, $\Theta_{uu}^{(i+1)}$, we see that this block is identical in both $\Theta_z^{(i+1)}$ and $\Theta_x^{(i+1)}$, which makes the following policy update step well-defined

$$\arg\min_u \mathcal{Q}^{(i+1)}(u_k, x_k) = \arg\min_u \chi_k^\top \Theta_x^{(i+1)} \chi_k = -(\Theta_{uu}^{(i+1)})^{-1} \Theta_{ux}^{(i+1)} x_k \overset{(12)}{=} -K_x^{(i+1)} x_k$$
$$= \arg\min_u \zeta_k^\top \Theta_z^{(i+1)} \zeta_k = -(\Theta_{uu}^{(i+1)})^{-1} \Theta_{uz}^{(i+1)} z_k =: -K_z^{(i+1)} z_k. \tag{19}$$

Notice that the updated policy at each iteration has two equivalent parameterizations in terms of the minimal state x and the non-minimal state z. Specifically, it holds that $K_z^{(i+1)} = K_x^{(i+1)} T$. The algorithm proposed in (Lopez et al., 2023) repeatedly solves for $K_x^{(i+1)}$ and was shown to converge to the optimal policy, i.e., $\lim_{i\to\infty} K_x^{(i+1)} = K_x^*$. However, that algorithm cannot be used to obtain $K_z^{(i+1)}$ for two reasons: (i) solving a set of $n + m$ equations of the form (11) for $\Theta_x^{(i+1)}$ (compare Section 2.1) requires state measurements, which are not available in our setting, and (ii) one cannot evaluate $K_z^{(i+1)} = K_x^{(i+1)} T$ because T is also unknown and depends on model parameters (A, B, C). Therefore, in the following we explain how one can solve for $\Theta_z^{(i+1)}$ directly, without requiring model knowledge and using only input-output data.

Recall from Lemma 3 that if $\{u_k, y_k\}_{k=-\ell}^{N-1}$ are input-output data collected from system (2), with u being persistently exciting of order $\ell + n + 1$, then the matrix $[Z_0^\top \ U_0^\top]^\top$ (defined as in (8)) has full row rank (specifically, its rank is equal to $\nu := m(\ell + 1) + n$). Therefore, there exist ν linearly independent columns of this matrix of the form $\zeta_{k_j} = [z_{k_j}^\top \ u_{k_j}^\top]^\top$, for $j \in \{1, \ldots, \nu\}$. These vectors can be arranged in a square non-singular matrix of the form

$$Z := \begin{bmatrix} \zeta_{k_1} & \cdots & \zeta_{k_\nu} \end{bmatrix} = \begin{bmatrix} z_{k_1} & \cdots & z_{k_\nu} \\ u_{k_1} & \cdots & u_{k_\nu} \end{bmatrix} \in \mathbb{R}^{\nu\times\nu}. \tag{20}$$

Pre- and post-multiplying (13) by $Z^\top \mathscr{T}^\top$ and $\mathscr{T}Z$, respectively, with $\mathscr{T}$ as in (16) results in

$$Z^\top \mathscr{T}^\top \Theta_x^{(i+1)} \mathscr{T}Z = Z^\top \mathscr{T}^\top \overline{Q} \mathscr{T}Z + Z^\top \mathscr{T}^\top \Phi_i^\top \Theta_x^{(i+1)} \Phi_i \mathscr{T}Z. \tag{21}$$

Since the product $\mathscr{T}Z$ results in a full row rank matrix, then a unique solution $\Theta_x^{(i+1)}$ of (13) implies the same unique solution of (21). Notice that (21) still depends on model parameters (through $\mathscr{T}, \overline{Q}$ and Φ_i). Our goal is to arrive at a model-free version of this equation. By recalling the definition of $\Theta_z^{(i+1)}$ in (18), one can rewrite (21) as

$$Z^\top \Theta_z^{(i+1)} Z = Z^\top \mathscr{T}^\top \overline{Q} \mathscr{T}Z + Z^\top \mathscr{T}^\top \Phi_i^\top \Theta_x^{(i+1)} \Phi_i \mathscr{T}Z. \tag{22}$$

Moreover, by closely examining the rightmost term on the right hand side, one can use $x_k = Tz_k$ as well as the state equation in (2) to rewrite $\Phi_i \mathscr{T}Z$ as follows

$$\begin{aligned}
\Phi_i \mathscr{T}Z &= \begin{bmatrix} AT & B \\ -K_x^{(i)}AT & -K_x^{(i)}B \end{bmatrix} \begin{bmatrix} z_{k_1} \cdots z_{k_\nu} \\ u_{k_1} \cdots u_{k_\nu} \end{bmatrix} = \begin{bmatrix} Tz_{k_1+1} & \cdots & Tz_{k_\nu+1} \\ -K_x^{(i)}Tz_{k_1+1} & \cdots & -K_x^{(i)}Tz_{k_\nu+1} \end{bmatrix} \\
&= \begin{bmatrix} T & 0 \\ 0 & I_m \end{bmatrix} \begin{bmatrix} z_{k_1+1} & \cdots & z_{k_\nu+1} \\ -K_z^{(i)}z_{k_1+1} & \cdots & -K_z^{(i)}z_{k_\nu+1} \end{bmatrix} =: \mathscr{T}\Sigma_{i,k+1}, \quad \underbrace{}_{\overset{(19)}{=} K_z^{(i)}}
\end{aligned} \tag{23}$$

where we have defined $\Sigma_{i,k+1}$ as

$$\Sigma_{i,k+1} := \begin{bmatrix} z_{k_1+1} & \cdots & z_{k_\nu+1} \\ -K_z^{(i)}z_{k_1+1} & \cdots & -K_z^{(i)}z_{k_\nu+1} \end{bmatrix} \in \mathbb{R}^{\nu \times \nu}. \tag{24}$$

Plugging this back into (22), we obtain the following equation

$$Z^\top \Theta_z^{(i+1)} Z = Z^\top \mathscr{T}^\top \overline{Q} \mathscr{T}Z + \Sigma_{i,k+1}^\top \underbrace{\mathscr{T}^\top \Theta_x^{(i+1)} \mathscr{T}}_{\overset{(18)}{=} \Theta_z^{(i+1)}} \Sigma_{i,k+1}. \tag{25}$$

In its current form, however, (25) still depends on model parameters through the constant term $Z^\top \mathscr{T}^\top \overline{Q} \mathscr{T}Z$. This term can be equivalently expressed as

$$\begin{aligned}
Z^\top \mathscr{T}^\top \overline{Q} \mathscr{T}Z &= \begin{bmatrix} z_{k_1}^\top & u_{k_1}^\top \\ \vdots & \vdots \\ z_{k_\nu}^\top & u_{k_\nu}^\top \end{bmatrix} \begin{bmatrix} T^\top & 0 \\ 0 & I_m \end{bmatrix} \begin{bmatrix} C^\top Q_y C & 0 \\ 0 & R \end{bmatrix} \begin{bmatrix} T & 0 \\ 0 & I_m \end{bmatrix} \begin{bmatrix} z_{k_1} & \cdots & z_{k_\nu} \\ u_{k_1} & \cdots & u_{k_\nu} \end{bmatrix} \\
&= \begin{bmatrix} (CTz_{k_1})^\top & u_{k_1}^\top \\ \vdots & \vdots \\ (CTz_{k_\nu})^\top & u_{k_\nu}^\top \end{bmatrix} \begin{bmatrix} Q_y & 0 \\ 0 & R \end{bmatrix} \begin{bmatrix} CTz_{k_1} & \cdots & CTz_{k_\nu} \\ u_{k_1} & \cdots & u_{k_\nu} \end{bmatrix}.
\end{aligned} \tag{26}$$

Using $x_k = Tz_k$, one can express the output as $y_k = Cx_k = CTz_k$. Furthermore, we define

$$W := \begin{bmatrix} y_{k_1} & \cdots & y_{k_\nu} \\ u_{k_1} & \cdots & u_{k_\nu} \end{bmatrix} \in \mathbb{R}^{m+p \times \nu}. \tag{27}$$

This, together with (26) allows us to express $Z^\top \mathscr{T}^\top \overline{Q} \mathscr{T}Z = W^\top \hat{Q}W$ where we have defined $\hat{Q} := \mathrm{diag}(Q_y, R) \succ 0$. Finally, plugging back into (25) results in

$$Z^\top \Theta_z^{(i+1)} Z = W^\top \hat{Q}W + \Sigma_{i,k+1}^\top \Theta_z^{(i+1)} \Sigma_{i,k+1}, \tag{28}$$

7

Algorithm 1: Off-Policy Q-Learning for optimal output regulation problem

---

1. Collect $N + \ell$ samples of data $\{u_k, y_k\}_{k=-\ell}^{N-1}$ (with $N \geq m(\ell + n + 1) + n - 1$) by applying a PE input of order $\ell + n + 1$ to the system (cf. Definition 1). Then, construct $\{z_k\}_{k=0}^{N-1}$ as in (5) (cf. Remark 2 and Alsalti et al. (2023)).

2. Select $\nu = m(\ell + 1) + n$ linearly independent vectors of the form $\zeta_{k_j} = [z_{k_j}^\top \ u_{k_j}^\top]^\top, j \in \{1, \ldots, \nu\}$. Then, construct the matrices Z, W as in (20) and (27), respectively.

3. Let $i = 0$ and find an initial stabilizing feedback policy $K_z^{(0)}$ (cf. Remark 5).

4. Using the gain $K_z^{(i)}$, construct the matrix $\Sigma_{i,k+1}$ in (24).

5. Solve the matrix equation (28) for $\Theta_z^{(i+1)}$ and update $K_z^{(i+1)} = (\Theta_{uu}^{(i+1)})^{-1}\Theta_{uz}^{(i+1)}$.

6. If $\|K_z^{(i+1)} - K_z^{(i)}\| > \varepsilon$ for some $\varepsilon > 0$, let $i = i + 1$ and go to Step 4. Otherwise, stop.

---

which is a special case of the generalized Sylvester matrix equation (specifically, a generalized discrete-time Lyapunov equation), for which efficient algorithms to solve it are known to exist (Sasaki and Chansangiam, 2020). Furthermore, this equation is model-free and can be solved for $\Theta_z^{(i+1)}$ at each iteration without requiring model-knowledge and only using input-output data.

We now introduce the main result of this paper, which is an iterative off-policy Q-learning algorithm that solves (3). This is summarized in Algorithm 1, which takes as an input previously collected, persistently exciting, input-output data as well as an initial stabilizing policy (see Remark 5 below for a discussion). At each iteration, we use the collected data to solve (28) for $\Theta_z^{(i+1)}$ and obtain an improved policy $K_z^{(i+1)} = (\Theta_{uu}^{(i+1)})^{-1}\Theta_{uz}^{(i+1)}$ as in (19). The policy can now be used to start a new iteration, by updating the matrix $\Sigma_{i,k+1}$ in (28) (see (24)). The algorithm is terminated once the difference between two successive policies is below a user-defined threshold $\varepsilon > 0$.

Remark 5 *One can use (Lopez et al., 2023, Alg. 2) with $\{u_k, z_k\}_{k=0}^{N-1}$ to obtain a deadbeat controller $u = -K_{\mathrm{db}}z$ which can be used as an initial stabilizing policy for Algorithm 1. Alternatively, an initial stabilizing policy can be obtained by solving an LMI (cf. (Alsalti et al., 2023, Th. 5)).*

The following theorem shows that Algorithm 1 enjoys all of the theoretical guarantees of its state-feedback counterpart (Lopez et al., 2023, Alg. 1).

Theorem 6 *Let the conditions in Lemma 3 hold. Given an initial stabilizing policy $K_z^{(0)}$, the solution $\Theta_z^{(i+1)}$ of (28) exists and is unique. Moreover, every policy $K_z^{(i+1)}$ obtained at each iteration of Algorithm 1 is stabilizing and $\lim_{i \to \infty} K_z^{(i)} = K_z^*$ converges quadratically.*

Proof By construction, the matrix equations in (28) and (13) are equivalent in the sense that $\Theta_z^{(i+1)} = \mathscr{T}^\top \Theta_x^{(i+1)} \mathscr{T}$ (see the developments below (20)). This implies that if $K_z^{(0)} = K_x^{(0)}T$ is used as an initial matrix in Algorithm 1, and $K_x^{(0)}$ is used as an initial matrix in (Lopez et al., 2023, Alg. 1), then $K_z^{(i+1)} = K_x^{(i+1)}T$ at every iteration i, compare (19). Existence and uniqueness of the solution $\Theta_x^{(i+1)}$ to (13) is given by Lemma 4, which implies existence and uniqueness of $\Theta_z^{(i+1)}$. Moreover, since $u = -K_x^{(i+1)}x$ is stabilizing for system (2) and since $x = Tz$, it holds that

$K_z^{(i+1)} = K_x^{(i+1)} T$ is also stabilizing. Finally, equivalence of (28) and (13) shows that Algorithm 1 and (Lopez et al., 2023, Alg. 1) converge at the same quadratic rate. ∎

Remark 7 *In the presence of small magnitudes of measurement noise, (Lopez et al., 2023, Alg. 1) converges to a solution that differs from the optimal one K_x^* by an amount that is dependent on the noise bound. Although a comprehensive discussion about the effect of noise in the data is out of the scope of this paper, we expect that Algorithm 1 possesses similar inherent robustness properties.*

4. Comparisons and Simulation examples

In this section, we discuss the main differences of Algorithm 1 compared to existing algorithms that were proposed to solve problem (3). First, notice that (Jiang et al., 2020) uses *state* measurements and thus it is not a purely input-output method. The work by (Zhang et al., 2023) restricts the class of systems (2) to those that are stabilizable by a static output feedback (cf. Kučera and De Souza (1995)). In contrast, our method is applicable to any LTI system of the form (2). Moreover, both works (Jiang et al., 2020; Zhang et al., 2023) propose to solve the Bellman equation using the batch least squares approach, and use a number of data points that scales with $n^2 + m^2 + nm$. In contrast, we only require $N + \ell$ data points (where $N \geq m(\ell + n + 1) + n - 1$) and solve the Bellman equation using efficient solvers for the generalized discrete-time Lyapunov equation. Finally, unlike (Jiang et al., 2020; Zhang et al., 2023), we provide methods to guarantee persistence of excitation (cf. Lemma 3) as well as the design of an initial stabilizing policy from data (cf. Remark 5).

A different approach to solving the optimal output regulation problem was proposed in (Dai et al., 2023), where a policy $u_k = \mathcal{K}\xi_k$, with $\xi_k = [u_{[k-n,k-1]}^\top \ y_{[k-n,k-1]}^\top]^\top$, can be found by solving a data-dependent SDP. However, such SDPs are computationally expensive for large system dimensions. For the state-feedback case, computational complexity of (Lopez et al., 2023, Alg. 1) was studied and shown to outperform SDP-based methods that solve the LQR problem (e.g., (De Persis and Tesi, 2021)), see (Lopez et al., 2023, Sec. VII). We expect similar conclusions for Algorithm 1 in comparison to (Dai et al., 2023) for the output-feedback case.

In the following, we will compare the performance of our proposed Algorithm 1 to the SDP-based method proposed in (Dai et al., 2023) in a numerical case study. We use the definition of the non-minimal state z_k in (5), instead of ξ_k, when solving the SDP. This is because, unlike ξ_k, the state z_k is such that $[Z_0^\top \ U_0^\top]^\top$ has full row rank (cf. Lemma 3). We do not compare against any of the RL off-policy methods mentioned above because (i) (Jiang et al., 2020) requires state measurements, which we do not and (ii) (Zhang et al., 2023) imposes additional assumptions on the class of systems, thus rendering any comparison inconsistent.

For simulations, we consider random LTI systems (both open-loop stable and unstable) of different orders $n = \{3, 5, 10, 30, 50\}$. For each state dimension, we generate 100 systems with number of inputs and outputs shown in Table 1. For each system, we run the Q-learning (QL) Algorithm 1 (for 10 iterations only, regardless of the system dimension) and the method proposed by (Dai et al., 2023) based on an SDP (adapted as described above) in order to solve (3) with $Q_y = 100I_p, R = I_m$. Since we use the non-minimal state z_k in the SDP, the two approaches (QL and SDP) use the same number of data points $N + \ell$, which we set to the lower bound, i.e., $N + \ell = (m + 1)(\ell + n + 1) - 2$. Hence, we use the same input-output data which we collect[2]

---

2. For open-loop unstable systems, using one (long) experiment results in output trajectories with arbitrarily large magnitudes. Instead, we use multiple shorter experiments whose inputs are *collectively PE*, see (van Waarde et al., 2020).

Table 1: Numerical simulations on 100 random systems of different dimensions.

Complexity			QL (Algorithm 1)		SDP (Dai et al., 2023)	
n	p	m	avg run time [s]	ϵ	avg run time [s]	ϵ
3	2	1	0.001	5.55×10^{-13}	0.398	9.04×10^{-6}
5	3	2	0.002	1.60×10^{-10}	1.256	0.3887
10	6	5	0.005	6.47×10^{-9}	14.130	22.5708
30	15	10	0.024	2.37×10^{-7}	n/a	n/a
50	20	15	0.116	1.66×10^{-4}	n/a	n/a

once offline by applying a PE input to the system sampled from a uniform random distribution $(\mathcal{U}(-1,1))^m$. For each run, we record the total run time as well as the error from the true optimal policy K_z^* given by $\epsilon = \left\| K_z^* - K_z^{\text{sol}} \right\|_2$, where K_z^{sol} is the solution returned by each scheme. Finally, we average the results over all 100 runs and report them in Table 1. For all simulations, Matlab R2021a was used on an Intel i7-10875H (2.30 GHz) with 16 GB of memory. The Matlab source code is available at https://doi.org/10.25835/zmlriehg.

It can be seen that Algorithm 1 (QL) outperforms the SDP method from (Dai et al., 2023) in terms of both total run time and the average error ϵ. For system dimensions $n = \{5, 10\}$, the SDP method was observed to be (highly) numerically unstable and, in many instances, no solution was found (using CVX (Grant and Boyd, 2014) with three different solvers SDPT3, MOSEK and SeDuMi). For those instances where no solution was found, we discard the experiment and repeat it until 100 successful runs are completed[3]. For even larger system dimensions, a solution to the SDP was never found on our machine for open-loop unstable systems. In comparison, our proposed Q-learning algorithm successfully converges to the optimal solution relatively fast, even for large system orders, due to efficiently solving the Bellman equation (28) using dlyap in Matlab. Note that QL requires an initial stabilizing policy, which was computed[4] as in (Lopez et al., 2023, Alg. 2), whereas the SDP method does not require any initialization.

5. Conclusion

In this paper, we presented a data-based off-policy Q-learning algorithm to solve the optimal output regulation problem for discrete-time LTI systems. This was done by extending an existing Q-learning algorithm for the LQR problem to the case of optimal output-feedback control, and exploiting the definition of a suitable non-minimal state of the system which depends only on past inputs and outputs. Our proposed algorithm only uses previously collected, persistently exciting input-output data. Persistence of excitation is easily enforced by a rank condition on the Hankel matrix of the input data. Furthermore, the algorithm uses an initial stabilizing policy which can be designed from data. Finally, it was shown that the algorithm asymptotically converges to the optimal policy. When compared to an existing method from the literature, our algorithm was shown to outperform it both in terms of total run time as well as scalability to large dimensional systems.

---

3. Out of these 100 successful runs of the SDP method, some did not return the optimal solution which explains the (large) average errors reported in Table 1

4. The computation time of the initial stabilizing policy is included in the values reported in Table 1.

Acknowledgments

References

Mohammad Alsalti, Victor G. Lopez, and Matthias A. Müller. Notes on data-driven output-feedback control of linear MIMO systems. *arXiv:2311.17484*, 2023.

Dimitri Bertsekas. *Dynamic programming and optimal control: Volume I*. Athena scientific, 2012.

Steven J Bradtke, B Erik Ydstie, and Andrew G Barto. Adaptive linear quadratic control using policy iteration. In *Proceedings of 1994 American Control Conference-ACC'94*, volume 3, pages 3475–3479. IEEE, 1994.

Xiaoyan Dai, Claudio De Persis, and Nima Monshizadeh. Data-driven optimal output feedback control of linear systems from input-output data. In *Proceedings of the 22nd IFAC World Congress, Yokohama, Japan*, 2023.

Claudio De Persis and Pietro Tesi. Low-complexity learning of linear quadratic regulators from noisy data. *Automatica*, 128:109548, 2021.

Maryam Fazel, Rong Ge, Sham Kakade, and Mehran Mesbahi. Global convergence of policy gradient methods for the linear quadratic regulator. In *International conference on machine learning*, pages 1467–1476. PMLR, 2018.

Graham C Goodwin and Kwai Sang Sin. *Adaptive filtering prediction and control*. Courier Corporation, 2014.

Michael Grant and Stephen Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. http://cvxr.com/cvx, March 2014.

Yi Jiang, Bahare Kiumarsi, Jialu Fan, Tianyou Chai, Jinna Li, and Frank L. Lewis. Optimal output regulation of linear discrete-time systems with unknown dynamics using reinforcement learning. *IEEE Transactions on Cybernetics*, 50(7):3147–3156, 2020. doi: 10.1109/TCYB.2018.2890046.

Thomas Kailath. *Linear systems*, volume 156. Prentice-Hall Englewood Cliffs, NJ, 1980.

Bahare Kiumarsi, Frank L Lewis, Mohammad-Bagher Naghibi-Sistani, and Ali Karimpour. Optimal tracking control of unknown discrete-time linear systems using input-output measured data. *IEEE transactions on cybernetics*, 45(12):2770–2779, 2015.

Bahare Kiumarsi, Frank L Lewis, and Zhong-Ping Jiang. H_∞ control of linear discrete-time systems: Off-policy reinforcement learning. *Automatica*, 78:144–152, 2017.

V Kučera and Carlos E De Souza. A necessary and sufficient condition for output feedback stabilizability. *Automatica*, 31(9):1357–1359, 1995.

Frank L Lewis and Kyriakos G Vamvoudakis. Reinforcement learning for partially observable dynamic processes: Adaptive dynamic programming using measured output data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 41(1):14–25, 2010.

Frank L Lewis, Draguna Vrabie, and Vassilis L Syrmos. *Optimal control*. John Wiley & Sons, 3rd edition, 2012.

Victor G. Lopez, Mohammad Alsalti, and Matthias A. Müller. Efficient off-policy Q-learning for data-based discrete-time LQR problems. *IEEE Transactions on Automatic Control*, 68(5):2922–2933, 2023. doi: 10.1109/TAC.2023.3235967.

Syed Ali Asad Rizvi and Zongli Lin. Output feedback Q-learning control for the discrete-time linear quadratic regulator problem. *IEEE Transactions on Neural Networks and Learning Systems*, 30 (5):1523–1536, 2019. doi: 10.1109/TNNLS.2018.2870075.

Nopparut Sasaki and Pattrawut Chansangiam. Modified Jacobi-gradient iterative method for generalized Sylvester matrix equation. *Symmetry*, 12(11):1831, 2020.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Henk J van Waarde, Claudio De Persis, M Kanat Camlibel, and Pietro Tesi. Willems' fundamental lemma for state-space systems and its extension to multiple datasets. *IEEE Control Systems Letters*, 4(3):602–607, 2020.

Jan C. Willems. From time series to linear system—part I: Finite dimensional linear time invariant systems, part II: Exact modelling, part III: Approximate modelling. *Automatica*, 22(5):561–580, 675–694, 87–115, 1986. ISSN 0005-1098. doi: https://doi.org/10.1016/0005-1098(86)90066-X.

Jan C. Willems, Paolo Rapisarda, Ivan Markovsky, and Bart L.M. De Moor. A note on persistency of excitation. *Systems & Control Letters*, 54(4):325–329, 2005. ISSN 0167-6911.

Li Zhang, Jialu Fan, Wenqian Xue, Victor G. Lopez, Jinna Li, Tianyou Chai, and Frank L. Lewis. Data-driven H_∞ optimal output feedback control for linear discrete-time systems based on off-policy Q-learning. *IEEE Transactions on Neural Networks and Learning Systems*, 34(7):3553–3567, 2023. doi: 10.1109/TNNLS.2021.3112457.