An Iterative Data-Driven Linear Quadratic Method to Solve Nonlinear Discrete-Time Tracking Problems

Corrado Possieri, Gian Paolo Incremona, Giuseppe C. Calafiore and Antonella Ferrara

***Abstract*—The objective of this note is to introduce a novel data-driven iterative linear quadratic control method for solving a class of nonlinear optimal tracking problems. Specifically, an algorithm is proposed to approximate the Q-factors arising from linear quadratic stochastic optimal tracking problems. This algorithm is then coupled with iterative linear quadratic methods for determining local solutions to nonlinear optimal tracking problems in a purely data-driven setting. Simulation results highlight the potential of this method for field applications.**

***Index Terms*—Data-driven control design, linear quadratic control, optimal control, dynamic programming.**

I. Introduction

Reinforcement learning seeks to determine an efficient control policy without knowledge of the system model, by coupling features of adaptive [1] and optimal [2] control. These two methodologies are based on different paradigms: the former learns on-line how to control an unknown system based on measurements but does not have optimality as primary objective [3], whereas the latter allows one to determine the optimal feedback policy but requires a model for the system dynamics [4]. In turn, reinforcement learning methods aim at designing adaptive controllers that, on the basis of observations of the correspondence between actions and penalties/rewards, dynamically determine the optimal control policy [5].

The dynamic programming algorithm [6] constitutes one of the most intuitive approaches for dealing with dynamic optimization problems. It allows to break the complexity of a cumulative optimization problem by subdividing it into sub-problems in a recursive manner [7]. When dealing with nonlinear optimal control problems, this reduces to computing the solution of a recursive equation (see [6, Eq. (3.2), (3.3)]), whose analytical expression is generically hard to obtain in practice [8]. On the other hand, if the system is linear, the dynamic programming algorithm reduces to a simple difference equation to be solved backwards in time: the so-called difference Riccati equation [9]. In [10]–[12], iterative versions of linear quadratic (LQ) methods have been shown how to use such a difference equation for finding the solution to nonlinear optimal control problems. However, in all the above cases, perfect knowledge of the model of the system is required.

Reinforcement learning (also referred to as approximate dynamic programming, or neuro-dynamic programming) methods have been proposed to overcome such a requirement and for dealing with problems in which the dynamic programming equation is not analytically solvable [13]. Among these techniques, Q-learning is one of the most commonly used [14]. The key idea behind Q-learning is to employ samples of the system trajectories in order to find an approximation of the state-action value function of the dynamic programming algorithm [15].

Q-learning for linear discrete-time systems has a relatively long history. A policy iteration-based Q-learning algorithm that requires an initial stabilizing feedback gain has been proposed for instance in [16] to solve the LQ regulator problem. This requirement has been removed in [17] by designing a value-iteration based Q-learning algorithm, while in [18] both policy-iteration and value-iteration based algorithms have been proposed to solve the LQ regulator problem by output feedback. In [19], [20], similar techniques have been used to solve the infinite-horizon LQ tracking problem.

Differently from [16]–[18], [20], in this note we deal with LQ optimal tracking over finite-horizon rather than with the LQ regulator problem over infinite-horizon. Furthermore, differently from [19], where policy-iteration and value-iteration based algorithms have been proposed to solve the deterministic LQ optimal tracking problem, here we propose a value function approximation method for dealing with stochastic LQ optimal tracking problems over finite-horizon. This approximation method can be viewed as a particular instance of the *normalized advantage function method* (briefly, NAF) firstly introduced in [21], and suitable for systems featuring continuous state and action spaces such as those commonly found in robotics (see e.g., [22], [23]). This novel technique is instrumental for determining a locally optimal control policy for nonlinear tracking problems. In fact, it is readily amenable to coupling with iterative LQ methods, thus allowing one to break the complexity of nonlinear problems by iteratively applying the proposed value function approximation approach. Overall, our proposed method belongs to the class of data-driven (model-free) control algorithms, which are gaining increasing popularity, as testified by e.g., [24]–[29].

This is the final version of the manuscript accepted for publication in IEEE Transactions on Automatic Control, doi:10.1109/TAC.2021.3056398. This work has been partially supported by the Italian Ministry for Research in the framework of the 2017 Program for Research Projects of National Interest (PRIN), Grant no. 2017YKXYXJ.

C. Possieri is with Istituto di Analisi dei Sistemi ed Informatica "A. Ruberti", Consiglio Nazionale delle Ricerche (IASI-CNR), 00185 Roma, Italy (e-mail: corrado.possieri@iasi.cnr.it).

G. P. Incremona is with Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, 20133 Milan, Italy (e-mail: gianpaolo.incremona@polimi.it)

G. C. Calafiore is with Dipartimento di Elettronica e Telecomunicazioni, Politecnico di Torino, 10129 Torino, Italy, and also with IEIIT-CNR Torino, 10129 Torino, Italy (e-mail: giuseppe.calafiore@polito.it).

A. Ferrara is with Dipartimento di Ingegneria Industriale e dell'Informazione, University of Pavia, 27100 Pavia, Italy (e-mail: antonella.ferrara@unipv.it).

II. Approximation in value space of

LQ STOCHASTIC PROBLEMS

A. Linear quadratic finite-horizon optimal control problem

Consider the discrete-time stochastic linear system

$$x_{k+1} = A_k x_k + B_k u_k + G_k w_k, \tag{1}$$

where $k \in \mathbb{N}$ is a time index, $x_k \in \mathbb{R}^n$ is the state of the system, $u_k \in \mathbb{R}^m$ is the control input, and $w_k \in \mathbb{R}^p$ is a disturbance acting on the system. In particular, we assume that $\{w_k\}_{k\in\mathbb{N}}$ is a sequence of independent random variables with mean $\mathbb{E}[w_k] = \mu_k$ and variance $\mathrm{Var}[w_k] = \Sigma_k$. Define then the *quadratic, finite-horizon cost function*

$$J(x_0, u_0, \dots, u_{N-1}) = \mathbb{E}\Bigg[\sum_{k=0}^{N-1} \Big(\|x_k - x_k^\diamond\|_{W_k} + \|u_k - u_k^\diamond\|_{R_k} \Big) + \|x_N - x_N^\diamond\|_{W_N} \Bigg], \tag{2}$$

where $N \in \mathbb{N}$, $W_k \in \mathbb{R}^{n\times n}$, $W_k \succeq 0$, $k \in \{0, \dots, N\}$, $R_k \in \mathbb{R}^{m\times m}$, $R_k \succ 0$, $k \in \{0, \dots, N-1\}$, $\|x\|_W = x^\top W x$, $\{x_k^\diamond\}_{k=0}^N$ is a desired trajectory for the state of (1), and $\{u_k^\diamond\}_{k=0}^N$ is a reference control input. Note that, differently from other techniques proposed in the literature, e.g., [30], we do not require feasibility of the references $\{x_k^\diamond\}_{k=0}^N$ and $\{u_k^\diamond\}_{k=0}^N$, i.e., we admit the case that $x_{k+1}^\diamond \neq A_k x_k^\diamond + B_k u_k^\diamond$.

In order to determine a solution to the optimal control problem (1), (2), it is possible to use the *dynamic programming algorithm* [6]. Such a procedure iteratively constructs the cost-to-go functions $J_\kappa^\diamond(x_\kappa) := \min_{u_\kappa,\dots,u_{N-1}} \mathbb{E}[\sum_{k=\kappa}^{N-1}(\|x_k - x_k^\diamond\|_{W_k} + \|u_k - u_k^\diamond\|_{R_k}) + \|x_N - x_N^\diamond\|_{W_N}]$, $\kappa = 0, \dots, N$. This method starts by letting $J_N^\diamond(x_N) = \|x_N - x_N^\diamond\|_{W_N}$, and, going backwards for $\kappa = N-1, \dots, 0$, defining the *Q-factors* (sometimes referred to also as *state-action value function* [5])

$$Q_\kappa(x_\kappa, u_\kappa) = \mathbb{E}\Big[\|x_\kappa - x_\kappa^\diamond\|_{W_\kappa} + \|u_\kappa - u_\kappa^\diamond\|_{R_\kappa} + J_{\kappa+1}^\diamond(A_\kappa x_\kappa + B_\kappa u_\kappa + G_\kappa w_\kappa) \Big], \tag{3}$$

and solving the problem $J_\kappa^\diamond(x_\kappa) = \min_{u_\kappa} Q_\kappa(x_\kappa, u_\kappa)$.

The dynamic programming iteration can be equivalently formulated in terms of the Q-factors by letting $Q_{N-1}(x_{N-1}, u_{N-1}) = \mathbb{E}[\|x_{N-1} - x_{N-1}^\diamond\|_{W_{N-1}} + \|u_{N-1} - u_{N-1}^\diamond\|_{R_{N-1}} + \|A_{N-1}x_{N-1} + B_{N-1}u_{N-1} + G_{N-1}w_{N-1} - x_N^\diamond\|_{W_N}]$, and, going backwards for $\kappa = N-2, \dots, 0$, letting

$$Q_\kappa(x_\kappa, u_\kappa) = \mathbb{E}\Big[\|x_\kappa - x_\kappa^\diamond\|_{W_\kappa} + \|u_\kappa - u_\kappa^\diamond\|_{R_\kappa} + \min_{u_{\kappa+1}} Q_{\kappa+1}(A_\kappa x_\kappa + B_\kappa u_\kappa + G_\kappa w_\kappa, u_{\kappa+1}) \Big]. \tag{4}$$

With such a construction, the input solving the optimal control problem (1), (2) is given by

$$u_\kappa^\star = \operatorname*{argmin}_{u_\kappa} Q_\kappa(x_\kappa, u_\kappa), \quad \kappa = 0, \dots, N-1.$$

The following theorem provides the solution to the optimal control problem (1), (2), and its proof follows by classical preview control (see, e.g., [31, Sec. 2.1]).

Theorem 1 (Solution to LQ stochastic problems)**.** *The solution to the optimal control problem* (1)*,* (2) *is given by*

$$u_\kappa^\star = -(R_\kappa + B_\kappa^\top P_{\kappa+1} B_\kappa)^{-1} \Big(B_\kappa^\top P_{\kappa+1} A_\kappa x_\kappa + B_\kappa^\top P_{\kappa+1} G_\kappa \mu_\kappa + \tfrac{1}{2} B_\kappa^\top D_{\kappa+1} - R_\kappa u_\kappa^\diamond \Big), \tag{5}$$

where the matrices $P_\kappa \in \mathbb{R}^{n\times n}$, $D_\kappa \in \mathbb{R}^n$, and $c_\kappa \in \mathbb{R}$ are computed iteratively starting from

$$P_N = W_N, \quad D_N = -2W_N x_N^\diamond, \quad c_N = x_N^{\diamond\top} W_N x_N^\diamond. \tag{6}$$

and proceeding backwards as

$$\begin{aligned}
P_\kappa &= A_\kappa^\top P_{\kappa+1} A_\kappa + W_\kappa \\
&\quad - A_\kappa^\top P_{\kappa+1} B_\kappa (R_\kappa + B_\kappa^\top P_{\kappa+1} B_\kappa)^{-1} B_\kappa^\top P_{\kappa+1} A_\kappa, && \text{(7a)}\\
D_\kappa &= A_\kappa^\top D_{\kappa+1} - 2W_\kappa x_\kappa^\diamond + 2A_\kappa^\top P_{\kappa+1} G_\kappa \mu_\kappa \\
&\quad - A_\kappa^\top P_{\kappa+1} B_\kappa (R_\kappa + B_\kappa^\top P_{\kappa+1} B_\kappa)^{-1} B_\kappa^\top D_{\kappa+1} \\
&\quad + 2A_\kappa^\top P_{\kappa+1} B_\kappa (R_\kappa + B_\kappa^\top P_{\kappa+1} B_\kappa)^{-1} R_\kappa u_\kappa^\diamond \\
&\quad - 2A_\kappa^\top P_{\kappa+1} B_\kappa (R_\kappa + B_\kappa^\top P_{\kappa+1} B_\kappa)^{-1} B_\kappa^\top P_{\kappa+1} G_\kappa \mu_\kappa, && \text{(7b)}\\
c_k &= c_{\kappa+1} + \mathrm{tr}(G_\kappa^\top P_{\kappa+1} G_\kappa \Sigma_\kappa) + u_\kappa^{\diamond\top} R_\kappa u_\kappa^\diamond \\
&\quad + D_{\kappa+1} G_\kappa \mu_\kappa + x_\kappa^{\diamond\top} W_\kappa x_\kappa^\diamond + \mu_\kappa^\top G_\kappa^\top P_{\kappa+1} G_\kappa \mu_\kappa \\
&\quad - \mu_\kappa^\top G_\kappa^\top P_{\kappa+1} B_\kappa (R_\kappa + B_\kappa^\top P_{\kappa+1} B_\kappa)^{-1} B_\kappa^\top P_{\kappa+1} G_\kappa \mu_\kappa \\
&\quad + 2\mu_\kappa^\top G_\kappa^\top P_{\kappa+1} B_\kappa (R_\kappa + B_\kappa^\top P_{\kappa+1} B_\kappa)^{-1} R_\kappa u_\kappa^\diamond \\
&\quad - D_{\kappa+1}^\top B_\kappa (R_\kappa + B_\kappa^\top P_{\kappa+1} B_\kappa)^{-1} B_\kappa^\top P_{\kappa+1} G_\kappa \mu_\kappa \\
&\quad - \tfrac{1}{4} D_{\kappa+1}^\top B_\kappa (R_\kappa + B_\kappa^\top P_{\kappa+1} B_\kappa)^{-1} B_\kappa^\top D_{\kappa+1}. \\
&\quad + D_{\kappa+1}^\top B_\kappa (R_\kappa + B_\kappa^\top P_{\kappa+1} B_\kappa)^{-1} R_\kappa u_\kappa^\diamond \\
&\quad - u_\kappa^{\diamond\top} R_\kappa (R_\kappa + B_\kappa^\top P_{\kappa+1} B_\kappa)^{-1} R_\kappa u_\kappa^\diamond. && \text{(7c)}
\end{aligned}$$

Furthermore, the cost-to-go functions $J_\kappa^\diamond(x_\kappa)$ are given by

$$J_\kappa^\diamond(x_\kappa) = \|x_\kappa\|_{P_\kappa} + D_\kappa^\top x_\kappa + c_\kappa. \tag{8}$$

We thus consider the following corollary.

Corollary 1 (Q-factors in LQ stochastic problems)**.** *Define $\eta_\kappa = \mathrm{col}(x_\kappa, u_\kappa)$. There exist $\Theta_\kappa \in \mathbb{R}^{(n+m)\times(n+m)}$, $\Psi_\kappa \in \mathbb{R}^{n+m}$ and $\phi_\kappa \in \mathbb{R}$ such that*

$$Q_\kappa(\eta_\kappa) = \|\eta_\kappa\|_{\Theta_\kappa} + \Psi_\kappa^\top \eta_\kappa + \phi_\kappa, \tag{9}$$

with $\Theta_\kappa \succeq 0$, for $\kappa \in \{0, \dots, N-1\}$. Furthermore, if $W_\kappa \succ 0$ for $\kappa = 0, \dots, N$, then $\Theta_0, \dots, \Theta_{N-1} \succ 0$ and

$$\begin{bmatrix} \phi_\kappa & \frac{1}{2}\Psi_\kappa^\top \\ \frac{1}{2}\Psi_\kappa & \Theta_\kappa \end{bmatrix} \succeq 0, \quad \kappa = 0, \dots, N-1. \tag{10}$$

Proof. By Theorem 1, we have that (9) holds with

$$\begin{aligned}
\Theta_\kappa &= \begin{bmatrix} W_\kappa + A_\kappa^\top P_{\kappa+1} A_\kappa & A_\kappa^\top P_{\kappa+1} B_\kappa \\ B_\kappa^\top P_{\kappa+1} A_\kappa & R_\kappa + B_\kappa^\top P_{\kappa+1} B_\kappa \end{bmatrix}, && \text{(11a)}\\
\Psi_\kappa &= \begin{bmatrix} A_\kappa^\top D_{\kappa+1} - 2W_\kappa x_\kappa^\diamond + 2A_\kappa^\top P_{\kappa+1} G_\kappa \mu_\kappa \\ B_\kappa^\top D_{\kappa+1} - 2R_\kappa u_\kappa^\diamond + 2B_\kappa^\top P_{\kappa+1} G_\kappa \mu_\kappa \end{bmatrix}, && \text{(11b)}\\
\phi_k &= x_\kappa^{\diamond\top} W_\kappa x_\kappa^\diamond + u_\kappa^{\diamond\top} R_\kappa u_\kappa^\diamond + \mathrm{tr}(G_\kappa^\top P_{\kappa+1} G_\kappa \Sigma_k) \\
&\quad + \mu_\kappa^\top G_\kappa^\top P_{\kappa+1} G_\kappa \mu_\kappa + D_{\kappa+1}^\top G_\kappa \mu_\kappa + c_{\kappa+1}. && \text{(11c)}
\end{aligned}$$

Furthermore, note that (7a) is the well-known discrete-time Riccati equation. Thus, by classical results about LQ optimal control [9], [32], one has $P_\kappa \succeq 0$, $\kappa = 0, \dots, N$. Therefore, we

have that $\Theta_\kappa \succeq 0$ for $\kappa = 0, \dots, N$ since $R_\kappa + B_\kappa^\top P_{\kappa+1} B_\kappa \succ 0$ and the Schur complement of Θ_κ equals $P_\kappa \succeq 0$ [33].

If, additionally, one has $W_\kappa \succ 0$, $\kappa = 0, \dots, N$, then $P_\kappa \succ 0$, $\kappa = 0, \dots, N$, thus implying that $\Theta_\kappa \succ 0$, $\kappa = 0, \dots, N$, by the same reasoning given above. Therefore, by (3) and Theorem 1, it results that $Q_\kappa(\eta_\kappa) \geq 0$ for all $\eta_\kappa \in \mathbb{R}^{n+m}$. Hence, by completing the squares in (9), one obtains

$$Q_\kappa(\eta_\kappa) = \|\eta_\kappa + \tfrac{1}{2}\Theta_\kappa^{-1}\Psi_\kappa\|_{\Theta_\kappa} - \tfrac{1}{4}\Psi_\kappa^\top \Theta_\kappa^{-1}\Psi + \phi_\kappa,$$

which implies that $\phi_\kappa - \frac{1}{4}\Psi_\kappa^\top \Theta_\kappa^{-1}\Psi \geq 0$. Thus, the inequality in (10) follows by classical Schur complement arguments. □

By partitioning Θ_κ and Ψ_κ, $\kappa = 0, \dots, N-1$, as

$$\Theta_\kappa = \begin{bmatrix} \Theta_{\kappa,1} & \Theta_{\kappa,2} \\ \Theta_{\kappa,2}^\top & \Theta_{\kappa,3} \end{bmatrix}, \qquad \Psi_\kappa = \begin{bmatrix} \Psi_{\kappa,1} \\ \Psi_{\kappa,2} \end{bmatrix}, \tag{12}$$

with $\Theta_{\kappa,1} \in \mathbb{R}^{n\times n}$, $\Theta_{\kappa,2} \in \mathbb{R}^{n\times m}$, $\Theta_{\kappa,3} \in \mathbb{R}^{m\times m}$, $\Psi_{\kappa,1} \in \mathbb{R}^n$, $\Psi_{\kappa,2} \in \mathbb{R}^m$, by Corollary 1, if $W_\kappa \succeq 0$ for $\kappa = 0, \dots, N$, then $\Theta_{\kappa,1} \succeq 0$ and $\Theta_{\kappa,3} \succ 0$. Furthermore, under such an hypothesis, letting $P_\kappa \in \mathbb{R}^{n\times n}$, $D_\kappa \in \mathbb{R}^n$ and $\phi \in \mathbb{R}$ be such that (8) holds, by (11), it results that, for $\kappa = 0, \dots, N-1$,

$$u_\kappa^\star = -\Theta_{\kappa,3}^{-1}(\Theta_{\kappa,2}^\top x_\kappa + \tfrac{1}{2}\Psi_{\kappa,2}), \tag{13a}$$
$$P_\kappa = \Theta_{\kappa,1} - \Theta_{\kappa,2}\Theta_{\kappa,3}^{-1}\Theta_{\kappa,2}^\top, \tag{13b}$$
$$D_\kappa = \Psi_{\kappa,1} - \Theta_{\kappa,2}\Theta_{\kappa,3}^{-1}\Psi_{\kappa,2}, \tag{13c}$$
$$c_\kappa = \phi_\kappa - \tfrac{1}{4}\Psi_{\kappa,2}^\top \Theta_{\kappa,3}^{-1}\Psi_{\kappa,2}. \tag{13d}$$

As shown in Theorem 1 and Corollary 1, the optimal policy is unaffected when the disturbances $\{w_k\}_{k=0}^N$ are replaced by their means $\{\mu_k\}_{k=0}^N$, i.e., the *certainty equivalence* property [13] holds for the stochastic optimization problem (1), (2). In particular, as shown in (7) and (8), the presence of w_k resulted in an additional constant term $\mathrm{tr}(G_\kappa^\top P_{\kappa+1} G_\kappa \Sigma_\kappa)$, that is irrelevant for the optimal control policy. Therefore, solving the stochastic optimal control problem (1), (2) is equivalent to minimizing (2) with respect to the deterministic dynamics

$$x_{k+1} = A_k x_k + B_k u_k + G_k \mu_k. \tag{14}$$

Thus, define for each $\kappa \in \{0, \dots, N-1\}$ the function

$$\Omega_\kappa(x_\kappa, u_\kappa, \mu_\kappa) = \|x_\kappa - x_\kappa^\diamond\|_{W_\kappa} + \|u_k\|_{R_\kappa} + J_{\kappa+1}^\diamond(A_\kappa x_\kappa + B_\kappa u_\kappa + G_\kappa \mu_\kappa), \tag{15}$$

which is the certainty equivalent of the Q-factor Q_κ given in (3), and consider the following corollary, whose proof is identical to that of Corollary 1.

Corollary 2 (Certainty equivalent of Q-factors in LQ problems)**.** *Let* $\zeta_\kappa = \mathrm{col}(x_\kappa, u_\kappa, \mu_k)$. *There exist* $\Pi_\kappa \in \mathbb{R}^{(n+m+p)\times(n+m+p)}$, $\Gamma_\kappa \in \mathbb{R}^{n+m+p}$ *and* $\sigma_\kappa \in \mathbb{R}$ *such that*

$$\Omega_\kappa(\zeta_\kappa) = \|\zeta_\kappa\|_{\Pi_\kappa} + \Gamma_\kappa^\top \zeta_\kappa + \sigma_\kappa, \quad \kappa \in \{0, \dots, N-1\}. \tag{16}$$

Although Corollary 2 is of theoretical interest, it cannot be used in practice since the values μ_κ are not usually known.

B. Approximation of the Q-factors

We propose a technique to approximate the Q-factors of LQ stochastic control problems. Assume to have at one's disposal state-control-successor triplets $(x_\kappa^{(i)}, u_\kappa^{(i)}, x_{\kappa+1}^{(i)})$, where $i \in \{1, \dots, S\}$ is the experiment number and $\kappa \in \{0, \dots, N-1\}$ is the discrete-time in the i-th experiment, with

$$x_{\kappa+1}^{(i)} = A_\kappa x_\kappa^{(i)} + B_\kappa u_\kappa^{(i)} + G_\kappa w_\kappa^{(i)}, \tag{17}$$

for $i = 1, \dots, S$ and $\kappa = 0, \dots, N-1$. In view of the results given in Section II-A, Algorithm 1 allows one to approximate the Q-factors on the basis of the available data.

Algorithm 1 Data-driven approximation of the Q-factors

Input: state-control-successor triplets $(x_\kappa^{(i)}, u_\kappa^{(i)}, x_{\kappa+1}^{(i)})$, $i = 1, \dots, S$, $\kappa = 0, \dots, N-1$, reference signals $\{u_\kappa^\diamond\}_{\kappa=0}^{N-1}$ and $\{x_\kappa^\diamond\}_{\kappa=0}^N$, weights $\{W_\kappa\}_{\kappa=0}^N$ and $\{R_\kappa\}_{\kappa=0}^{N-1}$

Output: estimates of the Q-factors Q_κ, $\kappa = 0, \dots, N-1$

for $\kappa = N-1$ **to** 0 **do**
  if $\kappa = N-1$ **then**
    for $i = 1, \dots, S$, define

$$\gamma_{N-1}^{(i)} := \|x_{N-1}^{(i)} - x_{N-1}^\diamond\|_{W_{N-1}} + \|u_{N-1}^{(i)} - u_{N-1}^\diamond\|_{R_{N-1}} + \|x_N^{(i)} - x_N^\diamond\|_{W_N} \tag{18}$$

  else
    for $i = 1, \dots, S$, define

$$\gamma_\kappa^{(i)} := \|x_\kappa^{(i)} - x_\kappa^\diamond\|_{W_\kappa} + \|u_\kappa^{(i)} - u_\kappa^\diamond\|_{R_\kappa} + \|x_{\kappa+1}^{(i)}\|_{\hat{P}_{\kappa+1}} + \hat{D}_{\kappa+1}^\top x_{\kappa+1}^{(i)} + \hat{c}_{\kappa+1} \tag{19}$$

  let $\eta_\kappa^{(i)} := \mathrm{col}(x_\kappa^{(i)}, u_\kappa^{(i)})$, $i = 1, \dots, S$
  let $\hat{\Theta}_\kappa$, $\hat{\Psi}_\kappa$, and $\hat{\phi}_\kappa$ be the solution to

$$\min_{\Theta_\kappa, \Psi_\kappa, \phi_\kappa} \sum_{i=1}^S \left(\|\eta_\kappa^{(i)}\|_{\Theta_\kappa} + \Psi_\kappa^\top \eta_\kappa^{(i)} + \phi_\kappa - \gamma_\kappa^{(i)}\right)^2 \quad \text{with } \Theta_{\kappa,3} \succ 0, \quad \Theta_\kappa \succeq 0 \tag{20}$$

  letting $\hat{\Theta}_\kappa$ and $\hat{\Psi}_\kappa$ be partitioned as in (12), let

$$\hat{P}_\kappa = \hat{\Theta}_{\kappa,1} - \hat{\Theta}_{\kappa,2}\hat{\Theta}_{\kappa,3}^{-1}\hat{\Theta}_{\kappa,2}^\top,$$
$$\hat{D}_\kappa = \hat{\Psi}_{\kappa,1} - \hat{\Theta}_{\kappa,2}\hat{\Theta}_{\kappa,3}^{-1}\hat{\Psi}_{\kappa,2},$$
$$\hat{c}_\kappa = \hat{\phi}_\kappa - \tfrac{1}{4}\hat{\Psi}_{\kappa,2}^\top \hat{\Theta}_{\kappa,3}^{-1}\hat{\Psi}_{\kappa,2}.$$

return $\hat{Q}_\kappa(\eta_\kappa) = \|\eta_\kappa\|_{\hat{\Theta}_\kappa} + \hat{\Psi}_\kappa^\top \eta_\kappa + \hat{\phi}_\kappa$, $\kappa = 0, \dots, N-1$

The following theorem shows that the outputs of Algorithm 1 are unbiased estimates of the Q-factors.

Theorem 2 (Estimation of the Q-factors)**.** *The function* $\hat{Q}_{N-1}$ *returned by Algorithm 1 is a feasible and unbiased estimate of* Q_{N-1}. *Similarly, the function* $\hat{Q}_k$ *returned by Algorithm 1 is a feasible and unbiased estimate of* Q_k *given the estimate* $\hat{Q}_{k+1}$ *of* Q_{k+1}, *for* $k = 0, \dots, N-2$.

Proof. Let $\eta_\kappa^{(i)} = \mathrm{col}(x_\kappa^{(i)}, u_\kappa^{(i)})$, $\kappa = 0, \dots, N-1$, $i = 1, \dots, S$. Letting $\gamma_{N-1}^{(i)}$ be defined as in (18) and following

the construction given in the proof of Corollary 1, we have

$$\gamma_{N-1}^{(i)} = Q_{N-1}(\eta_{N-1}^{(i)}) + \epsilon_{N-1}^{(i)}, \tag{21a}$$

$$\epsilon_{N-1}^{(i)} = M_{N-1}^{(i)}\varpi_{N-1}^{(i)} + \|\varpi_{N-1}^{(i)}\|_{G_{N-1}^\top P_N G_{N-1}} - \operatorname{tr}(G_{N-1}^\top P_N G_{N-1}\Sigma), \tag{21b}$$

$$M_{N-1}^{(i)} = 2x_{N-1}^{(i)\top}A_{N-1}^\top P_N G_{N-1} + 2u_{N-1}^{(i)\top}B_{N-1}^\top P_N G_{N-1} + 2\mu_{N-1}^\top G_{N-1}^\top P_N G_{N-1} + D_N^\top G, \tag{21c}$$

$$\varpi_{N-1}^{(i)} = w_{N-1}^{(i)} - \mu_{N-1}. \tag{21d}$$

Since $\mathbb{E}[w_{N-1}] = \mu_{N-1}$ and $\mathbb{E}[\|\varpi_{N-1}\|_{G_{N-1}^\top P_N G_{N-1}}] = \operatorname{tr}(G_{N-1}^\top P_N G_{N-1}\Sigma)$, we have that $\mathbb{E}[\epsilon_{N-1}] = 0$. Hence, by [34, Thm. 10.1], since $\mathbb{E}[\epsilon_{N-1}] = 0$, the function $\hat{Q}_{N-1}$ obtained solving the ordinary least squares (OLS) problem (20) with $\kappa = N-1$ is a feasible and unbiased estimate of Q_{N-1}.

Following the construction given in (3), (8), (13), and using the results given in [13, Sec. 2.1.4], one has that

$$\gamma_\kappa^{(i)} = \|x_\kappa^{(i)} - x_\kappa^\diamond\|_{W_\kappa} + \|u_\kappa^{(i)} - u_\kappa^\diamond\|_{R_\kappa} + \min_{u_{\kappa+1}} \hat{Q}_{\kappa+1}(x_{\kappa+1}^{(i)}, u_{\kappa+1}) + \epsilon_\kappa^{(i)}, \tag{22a}$$

$$\epsilon_\kappa^{(i)} = M_\kappa^{(i)}\varpi_\kappa^{(i)} + \|\varpi_\kappa^{(i)}\|_{G_\kappa^\top \hat{P}_{\kappa+1} G_\kappa} - \operatorname{tr}(G_\kappa^\top \hat{P}_{\kappa+1} G_\kappa \Sigma), \tag{22b}$$

$$M_\kappa^{(i)} = 2x_\kappa^{(i)\top}A_\kappa^\top \hat{P}_{\kappa+1} G_\kappa + 2u_\kappa^{(i)\top}B_\kappa^\top \hat{P}_{\kappa+1} G_\kappa + 2\mu_\kappa^\top G_\kappa^\top \hat{P}_{\kappa+1} G_\kappa + \hat{D}_{\kappa+1}^\top G_\kappa, \tag{22c}$$

$$\varpi_\kappa^{(i)} = w_\kappa^{(i)} - \mu_\kappa. \tag{22d}$$

Note that, given the estimate $\hat{Q}_{k+1}$ of the Q-factor Q_{k+1}, the matrix $\hat{P}_\kappa$ is fixed and is given by (13b). Therefore, by the same reasoning given above, $\mathbb{E}[\epsilon_\kappa] = 0$. Thus, the function $\hat{Q}_k$ obtained solving the OLS (20) is a feasible and unbiased estimate of Q_k, given the estimate $\hat{Q}_{k+1}$ of Q_{k+1}. □

Once an estimate of the Q-factors has been obtained via Algorithm 1, and letting $\hat{\Theta}_k$ and $\hat{\Psi}_k$ be partitioned as in (12), an estimate of the optimal control $u_k^\star$ is obtained as

$$\hat{u}_k^\star = -\hat{\Theta}_{k,3}^{-1}(\hat{\Theta}_{k,2}^\top x_k + \tfrac{1}{2}\hat{\Psi}_{k,2}).$$

Building upon the proof of Theorem 2, the next remark deals with the case of noisy samples $x_\kappa^{(i)}$ and $u_\kappa^{(i)}$.

Remark 1 (Noisy samples). Let the measured samples $\bar{x}_\kappa^{(i)}$ and $\bar{u}_\kappa^{(i)}$ be affected by zero mean random noise, that is

$$\bar{x}_\kappa^{(i)} = x_\kappa^{(i)} + \vartheta_\kappa^{(i)}, \qquad \bar{u}_\kappa^{(i)} = u_\kappa^{(i)} + \varkappa_\kappa^{(i)},$$

where $x_\kappa^{(i)}$, $u_\kappa^{(i)}$ satisfy (17), $\mathbb{E}[\vartheta_\kappa] = 0$, and $\mathbb{E}[\varkappa_\kappa] = 0$. Letting $\varrho_\kappa^{(i)} = \operatorname{col}(\vartheta_\kappa^{(i)}, \varkappa_\kappa^{(i)})$, this implies that $\gamma_\kappa^{(i)}$ computed as in (21a) or as in (22a) with $x_\kappa^{(i)}$ and $u_\kappa^{(i)}$ substituted by $\bar{x}_\kappa^{(i)}$ and $\bar{u}_\kappa^{(i)}$, respectively, satisfies $\gamma_\kappa^{(i)} = Q_\kappa(\eta_\kappa^{(i)}) + \chi_\kappa^{(i)}$, where

$$\chi_\kappa^{(i)} = 2\eta_\kappa^{(i)\top}\Theta_\kappa\varrho_\kappa^{(i)} + \|\varrho_\kappa^{(i)}\|_{\Theta_\kappa} + \Psi_\kappa^\top\varrho_\kappa^{(i)} + \epsilon_\kappa^{(i)}.$$

Since $\mathbb{E}[\chi_\kappa] = \operatorname{tr}(\Theta_\kappa \operatorname{Var}[\varrho_\kappa])$, the presence of measurement noise leads to biased estimates $\hat{\phi}_\kappa$, although, by construction, it does not alter the estimated optimal control $\hat{u}_\kappa^\star$. △

Note that, although the estimates of the Q-factors obtained via Algorithm 1 are unbiased, they need not be the best estimates in terms of variance of $\operatorname{col}(\operatorname{vec}(\hat{\Theta}_{N-1}), \hat{\Psi}_{N-1}, \hat{\phi}_{N-1})$ due to the presence of heteroscedasticity (see (21), (22) and [34, Sec. 11.1]). In particular, since $\operatorname{Var}[\epsilon_\kappa]$ depends on the values of $\eta_\kappa^{(i)}$, statistical inference based on OLS may be misleading, i.e., the OLS variance estimator $\operatorname{Var}[\operatorname{col}(\operatorname{vec}(\hat{\Theta}_\kappa), \hat{\Psi}_\kappa, \hat{\phi}_\kappa)]$, where $\operatorname{vec}(\hat{\Theta}_\kappa)$ is the vector formed by stacking the columns of matrix $\hat{\Theta}_\kappa$ does not provide a consistent estimate of the variance of the OLS estimates (see [34, Sec. 10.2.2]). In principle, this aspect can be mitigated by letting $S \to +\infty$ and using the feasible generalized least squares estimator (see [34, Sec. 11.6]). However, for small and medium sizes of S, it may be convenient to use the OLS estimator (20) since it is more efficient (see [34, Chap. 11]).

In the following corollary, we characterize the minimum number of samples that are generically required in the certainty equivalent so as to let the output of Algorithm 1 be the Q-factors $Q_0(x_0, u_0), \ldots, Q_{N-1}(x_{N-1}, u_{N-1})$.

Corollary 3 (Sample complexity in the certainty equivalent)**.** *Let $w_k = \mu_k$ for all $k \in \{0, \ldots, N-1\}$. If*

$$S \geq \frac{1}{2}(m+n+1)(m+n+2), \tag{23}$$

then, for almost all initial conditions $x_0^{(i)} \in \mathbb{R}^n$ and control sequences $\{u_k^{(i)}\}_{k=0,\ldots,N-1}$, $i = 0, \ldots, \ell$, the output of Algorithm 1 are the Q-factors $Q_k(x_k, u_k)$, $k = 0, \ldots, N-1$.

Proof. Following the proof of Theorem 2, since $w_{N-1} = \mu_{N-1}$, one has $\gamma_{N-1}^{(i)} = Q_{N-1}(\eta_{N-1}^{(i)})$. Note that the Q-factor given in (9) can be rewritten as $Q_{N-1}(\eta_{N-1}) = \operatorname{col}(\eta_{N-1} \otimes \eta_{N-1}, \eta_{N-1}, 1)^\top \operatorname{col}(\operatorname{vec}(\Theta_{N-1}), \Psi_{N-1}, \phi_{N-1})$, where $\otimes$ is the Kronecker product. Following the construction made in [35, p. 40], note that $\eta_{N-1} \otimes \eta_{N-1}$ is the quadratic polynomial vector containing all possible products of the $n+m$ components of η_{N-1}. Since Θ_{N-1} is a symmetric matrix and it has only $\frac{1}{2}(n+m)(n+m+1)$ independent elements, it is possible to define a quadratic basis set $\bar{\eta}_{N-1}$ having $\frac{1}{2}(n+m)(n+m+1)$ independent elements by removing redundant entries from $\eta_{N-1} \otimes \eta_{N-1}$. Hence, letting $\overline{\operatorname{vec}}(\Theta_{N-1})$ be the vectorization of the corresponding elements of Θ_{N-1} and $\gamma_{N-1} = \operatorname{col}(\gamma_{N-1}^{(1)}, \ldots, \gamma_{N-1}^{(S)})$, $\Lambda_{N-1} = \operatorname{col}(\overline{\operatorname{vec}}(\Theta_{N-1}), \Psi_{N-1}, \phi_{N-1})$, $\delta_{N-1}^{(i)} = \operatorname{col}(\bar{\eta}_{N-1}^{(i)}, \eta_{N-1}^{(i)}, 1)$, $i = 1, \ldots, S$, $\Xi_{N-1} = \operatorname{col}(\delta_{N-1}^{(1)\top}, \ldots, \delta_{N-1}^{(S)\top})$, the objective function of the optimization problem (20) with $\kappa = N-1$ can be rewritten as $\|\Xi_{N-1}\Lambda_{N-1} - \gamma_{N-1}\|^2$. By [36], if (23) holds, that is the number of rows of Ξ_{N-1} is greater than or equal to the dimension of the vector Λ_{N-1}, then the matrix Ξ_{N-1} has full rank for almost all $x_0^{(i)}$ and control sequences $\{u_k^{(i)}\}_{k=0,\ldots,N-1}$, $i = 0, \ldots, \ell$. In all these cases, there is an unique solution to the unconstrained ordinary least squares problem $\min_{\Lambda_{N-1}} \|\Xi_{N-1}\Lambda_{N-1} - \gamma_{N-1}\|^2$,

$$\hat{\Lambda}_{N-1} = (\Xi_{N-1}^\top \Xi_{N-1})^{-1}\Xi_{N-1}^\top \gamma_{N-1}.$$

Reshaping $\hat{\Lambda}_{N-1}$ in order to obtain the matrices $\hat{\Theta}_{N-1}$, $\hat{\Psi}_{N-1}$ and the constant $\hat{\phi}_{N-1}$, one obtains the matrix given in (11) with $\kappa = N-1$, which satisfies the constraints of the constrained ordinary least squares problem (20) with $\kappa = N-1$ and let the corresponding value of the objective function be 0. Hence, the matrix $\hat{\Theta}_{N-1}$, the vector $\hat{\Psi}_{N-1}$ and the constant

$\hat{\phi}_{N-1}$ obtained reshaping $\hat{\Lambda}_{N-1}$ are the unique solution to the ordinary least squares problem (20) with $\kappa = N-1$ and $\hat{\Theta}_{N-1} = \Theta_{N-1}$, $\hat{\Psi}_{N-1} = \Psi_{N-1}$, $\hat{\phi}_{N-1} = \Phi_{N-1}$. Thus, the function $\hat{Q}_{N-1}$ returned by Algorithm 1 is the Q-factor Q_{N-1}. Therefore, following the same induction employed in the proof of Theorem 2 and repeating verbatim the reasoning given above with $N-1$ substituted by k, we have that the output $\hat{Q}_k$ of Algorithm 1 matches with the Q-factors Q_k. □

Note that if (23) does not hold, then the matrix Ξ_κ is singular and hence such a condition is necessary and sufficient in the certainty equivalence case.

The proof of Corollary 3 suggests how to simplify the computations to be carried out in Algorithm 1 when the certainty equivalent system (14) is considered. Namely, if the disturbances acting on the system are small (so that the dynamics of the system are essentially given by its certainty equivalent) and (23) holds, then, following the notation and the constructions used in the proof of Theorem 2 and Corollary 3, the solution to the optimization problems (20) is

$$\hat{\Lambda}_\kappa = (\Xi_\kappa^\top \Xi_\kappa)^{-1} \Xi_\kappa^\top \gamma_\kappa. \tag{24}$$

Then, the matrix $\hat{\Theta}_\kappa$, the vector $\hat{\Psi}_\kappa$ and the constant $\hat{\phi}_\kappa$ can be obtained by reshaping $\hat{\Lambda}_\kappa$. In this case, if singular value decomposition (SVD) is used to compute the pseudo-inverse of Ξ_κ, the computational complexity of using the formula given in (24) is $O(S^2)$ [37, Sec. 5.12]. On the other hand, by [38, Sec. 6.4] the computational complexity of determining a solution to (20) via interior point methods is $O(S^3)$. Therefore, using (24) may be preferred for solving (20) provided that certainty equivalence essentially holds.

Algorithm 1 has been presented as a batch learning method. However, such an algorithm can be easily modified into an iterative learning algorithm, as discussed in the next remark.

Remark 2 (Iterative implementation of Algorithm 1). Assume that the disturbance acting on the system is small so as to allow to use the formulas given in (24) to determine a solution to the optimization problem (20) and let (23) hold. Note that, following the notation used in the proof of Corollary 3, the formula in (24) can be rewritten as

$$\hat{\Lambda}_\kappa = \left(\sum_{i=1}^{S} \delta_\kappa^{(i)} \delta_\kappa^{(i)\top} \right)^{-1} \sum_{i=1}^{S} \gamma_\kappa^{(i)} \delta_\kappa^{(i)}.$$

Hence, if a new sequence of state-control-successor triplets $(x_\kappa^{(S+1)}, u_\kappa^{(S+1)}, x_{\kappa+1}^{(S+1)})$, $\kappa \in \{0, \dots, N-1\}$, is gathered, the estimates of the Q-factors can be updated using recursive least squares (see [39, pp. 8-12]). Namely, assuming that the matrices $\hat{\Theta}_{\kappa+1}$, $\hat{\Psi}_{\kappa+1}$, $\hat{\phi}_{\kappa+1}$ are known, define $\hat{P}_{\kappa+1}$, $\hat{D}_{\kappa+1}$, and $\hat{c}_{\kappa+1}$ as in Step 8 of Algorithm 1 with κ substituted by $\kappa+1$. Hence, defining $Z_\kappa^S = (\sum_{i=1}^{S} \delta_\kappa^{(i)} \delta_\kappa^{(i)\top})^{-1}$, $\rho_\kappa^S = \sum_{i=1}^{S} \gamma_\kappa^{(i)} \delta_\kappa^{(i)}$, and $\gamma_\kappa^{(S+1)} := \|x_\kappa^{(S+1)} - x_\kappa^\diamond\|_{W_\kappa} + \|u_\kappa^{(S+1)} - u_\kappa^\diamond\|_{R_\kappa} + \|x_{\kappa+1}^{(S+1)}\|_{\hat{P}_{\kappa+1}} + \hat{D}_{\kappa+1}^\top x_{\kappa+1}^{(S+1)} + \hat{c}_{\kappa+1}$, and let $\delta_\kappa^{(S+1)}$ be defined as in the proof of Corollary 3. The matrix Z_κ^S and the vector ρ_κ^S can be then updated as

$$Z_\kappa^{S+1} = Z_\kappa^S + \frac{Z_\kappa^S \delta_\kappa^{(S+1)} \delta_\kappa^{(S+1)\top} Z_\kappa^S}{1 + \delta_\kappa^{(S+1)\top} Z_\kappa^S \delta_\kappa^{(S+1)}}, \tag{25a}$$

$$\rho_\kappa^{S+1} = \rho_\kappa^S + \gamma_\kappa^S \delta_\kappa^{(S+1)}. \tag{25b}$$

and the estimate of the matrix Λ_κ can be updated as

$$\hat{\Lambda}_\kappa = Z_\kappa^{S+1} \rho_\kappa^{S+1}. \tag{25c}$$

The matrix $\hat{\Theta}_\kappa$, the vector $\hat{\Psi}_\kappa$ and the constant $\hat{\phi}_\kappa$ can be then obtained by reshaping the vector $\hat{\Lambda}_\kappa$. Hence, as new state-control-successor triplets $(x_\kappa^{(i)}, u_\kappa^{(i)}, x_{\kappa+1}^{(i)})$, $\kappa \in \{0, \dots, N-1\}$, are gathered, the formulas in (25) can be used to iteratively update the estimates of the Q-factors. △

The references $\{x_k^\diamond\}_{k=0}^N$ and $\{u_k^\diamond\}_{k=0}^N$ are used in Algorithm 1 to define the samples $\gamma_k^{(i)}$ of the Q-factors. The following remark shows how to apply Algorithm 1 in the case that such references are not known explicitly.

Remark 3 (Unknown references). Algorithm 1 can be used even if the reference trajectory $\{x_k^\diamond\}_{k=0}^N$ and input $\{u_k^\diamond\}_{k=0}^N$ are not known explicitly. Namely, if these references are not known, but the running cost $\varsigma_k(x_k, u_k) = \|x_k - x_k^\diamond\|_{W_k} + \|u_k - u_k^\diamond\|_{R_k}$ and the final cost $\varsigma_N(x_N) = \|x_N - x_N^\diamond\|_{W_N}$ are measurable, as, e.g., when dealing with output tracking problems or with vision based control where relative positions error can be measured but not absolute positions, it suffices to redefine $\gamma_{N-1}^{(i)}$ in (18) as $\gamma_{N-1}^{(i)} = \varsigma_{N-1}(x_{N-1}^{(i)}, u_{N-1}^{(i)}) + c_N(x_N^{(i)})$ and $\gamma_k^{(i)}$ in (19) as $\gamma_\kappa^{(i)} = \varsigma_\kappa(x_\kappa^{(i)}, u_\kappa^{(i)}) + \|x_{\kappa+1}^{(i)}\|_{P_{\kappa+1}} + D_{\kappa+1}^\top x_{\kappa+1}^{(i)} + c_{\kappa+1}$. This aspect, together with the fact that the references $\{x_k^\diamond\}_{k=0}^N$ and $\{u_k^\diamond\}_{k=0}^N$ need not be feasible, further motivates the interest in the technique given in this section. In particular, note that it does not require any dynamics inversion but just samples of the state, of the input, and of the cost. △

The next remark discusses the relation between Algorithm 1 and the normalized advantage function method [21].

Remark 4 (Relation with NAF). Algorithm 1 can be viewed as a particular instance of the NAF method [21]. In fact, by partitioning the matrices Θ_κ and Ψ_κ as in (12), since $Q_\kappa(x_\kappa, u_\kappa) = x_\kappa^\top \Theta_{\kappa,1} x_\kappa + 2 x_\kappa^\top \Theta_{\kappa,2} u_\kappa + u_\kappa^\top \Theta_{\kappa,3} u_\kappa + \Psi_{\kappa,1}^\top x_\kappa + \Psi_{\kappa,2}^\top u_\kappa + \phi_\kappa$, by completing the squares [37] with respect to u_κ, one obtains

$$\begin{aligned} Q_\kappa(x_\kappa, u_\kappa) &= \|u_\kappa + \Theta_{\kappa,3}^{-1}(\Theta_{\kappa,2}^\top x_\kappa + \tfrac{1}{2}\Psi_{\kappa,2})\|_{\Theta_{\kappa,3}} \\ &\quad - \|\Theta_{\kappa,2}^\top x_\kappa + \tfrac{1}{2}\Psi_{\kappa,2}\|_{\Theta_{\kappa,3}^{-1}} + x_\kappa^\top \Theta_{\kappa,1} x_\kappa + \Psi_{\kappa,1}^\top x_\kappa + \phi_\kappa. \end{aligned}$$

Therefore, by defining the *advantage functions*

$$A_\kappa(x_\kappa, u_\kappa) := Q_\kappa(x_\kappa, u_\kappa) - J_\kappa^\diamond(x_\kappa),$$

for $\kappa = 0, \dots, N-1$, by (13), one obtains that $A_\kappa(x_\kappa, u_\kappa) = \|u_\kappa + \Theta_{\kappa,3}^{-1}(\Theta_{\kappa,2}^\top x_\kappa + \frac{1}{2}\Psi_{\kappa,2})\|_{\Theta_{\kappa,3}}$. Hence, Algorithm 1 essentially consists in a NAF with affine policies. △

III. Data-driven iterative LQ control

In Section II, an algorithm has been proposed to estimate the Q-factors of LQ stochastic optimal tracking problems from data. In this section, we show how such an algorithm can be

coupled with the techniques given in [10] to solve a class of nonlinear optimal tracking problems in a data-driven setting. Namely, consider the system

$$\xi_{k+1} = f_k(\xi_k, \nu_k), \tag{26}$$

where $\xi_k \in \mathbb{R}^n$ is the state, $\nu_k \in \mathbb{R}^m$ is the input, $f_k : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is at least C^2, $k = 0, \dots, N-1$, and the cost

$$\Phi = \sum_{k=0}^{N-1} \left(\|\xi_k - \xi_k^\diamond\|_{W_k} + \|\nu_k\|_{R_k} \right) + \|\xi_N - \xi_N^\diamond\|_{W_N}, \tag{27}$$

where $N \in \mathbb{N}$, $R_k \succ 0$, $k \in \{0, \dots, N-1\}$, $W_k \succeq 0$, $k \in \{0, \dots, N\}$, and $\{\xi_k^\diamond\}_{k=0}^N$ is a given reference signal. By coupling Algorithm 1 with the techniques given in [10], the following Algorithm 2 allows us to determine a locally optimal solution to the optimal control problem (26), (27) from the initial condition $\xi_0 \in \mathbb{R}^n$ in a data-driven setting.

In fact, following the construction in [10], Algorithm 2 iterates the next procedure until convergence. Letting $\{\bar\xi_\kappa\}_{\kappa=0}^N$ be a nominal trajectory of system (26) corresponding to the control sequence $\{\bar\nu_\kappa\}_{\kappa=0}^{N-1}$, the linearization of system (26) about $\{(\bar\xi_\kappa, \bar\nu_\kappa)\}_{\kappa=0}^{N-1}$ is given by (1) with

$$A_k = \frac{\partial f_k}{\partial \xi_k}(\bar\xi_k, \bar\nu_k), \qquad B_k = \frac{\partial f_k}{\partial \nu_k}(\bar\xi_k, \bar\nu_k). \tag{28}$$

Furthermore, letting x_k and u_k denote the increment with respect to $\bar\xi_\kappa$ and $\bar\nu_\kappa$, respectively, $k = 0, \dots, N-1$, the corresponding value of the cost function Φ given in (27) is

$$\Phi = \sum_{k=0}^{N-1} \left(\|x_k - (\xi_k^\diamond - \bar\xi_k)\|_{W_k} + \|u_k - (-\bar\nu_k)\|_{R_k} \right) + \|x_N - (\xi_N^\diamond - \bar\xi_N)\|_{W_N}. \tag{29}$$

Therefore, Steps 2–10 of Algorithm 2 perform a data-driven approximation of the Q-factors of the optimal control problem (28), (29) using the results given in Section II-A. Then, Step 11 of Algorithm 2 determines an improved control sequence using the solution to the problem (28), (29). The following remark details how Algorithm 2 can be employed in a data-driven scenario.

Remark 5 (Data-driven scenario)*.* Algorithm 2 can be used even if a closed-form for the functions $f_0, \dots, f_{N-1}$ is not available. In fact, such functions are used in Steps 2, 7, and 11 to generate state-control-successor triplets to be fed to Algorithm 1. However, the same steps can be carried out either performing experiments or simulating the behavior of system (26) in a purely data-driven setting. △

The next theorem discusses the convergence of Algorithm 2.

Theorem 3 (Convergence to local minima)**.** *Suppose that* (23) *holds. There exists $\varepsilon > 0$ such that if $\|\Delta\xi_0^{(i)}\| \le \varepsilon$ and $\|\Delta\nu_\kappa^{(i)}\| \le \varepsilon$, then Algorithm 2 converges locally to a solution of problem* (26)*,* (27)*.*

Proof. By [12, Eq.s (4)–(11)], the deterministic problem (26), (27) is locally approximated by the LQ stochastic problem given by (29) and

$$x_\kappa = A_\kappa x_\kappa + B_\kappa u_\kappa + G_\kappa(x_\kappa, u_\kappa) w_\kappa, \tag{30}$$

where A_κ, B_κ are as in (28), G_κ is a continuous function, and w_κ are independent random variables. Therefore, there exists $\varepsilon > 0$ such that if $\|\Delta\xi_0^{(i)}\| \le \varepsilon$ and $\|\Delta\nu_\kappa^{(i)}\| \le \varepsilon$, then $G_\kappa(x_\kappa, u_\kappa)$ can be further approximated as a constant matrix. Thus, Step 11 of Algorithm 2 determines an improved control sequence using the solution to the LQ optimal tracking problem (29), (30), which, by Theorem 2 and Corollary 3, is determined in Steps 2–10 by approximating its Q-factors using Algorithm 1. By the discussion given in [11, Sec. VI], these steps correspond to a Newton method using the true Hessian. Thus, by Kantorovich theorem [40, Thm. 2.2] Algorithm 2 converges locally to a solution of problem (26), (27). □

In view of Theorem 3, the following remark shows how to determine a locally optimal policy from Algorithm 2.

Algorithm 2 Data-driven iterative LQ control

Input: initial condition ξ_0, weights $\{W_\kappa\}_{\kappa=0}^N$ and $\{R_\kappa\}_{\kappa=0}^{N-1}$, reference signal $\{\xi_k^\diamond\}_{k=0}^N$, initial guess $\{\bar\nu_\kappa\}_{\kappa=0}^{N-1}$ on the optimal solution to (26), (27), number $S \in \mathbb{N}$ of experiments to perform the approximation

Output: a locally optimal control sequence for (26), (27)

1: **repeat**

2: compute $\bar\xi_{\kappa+1} = f_\kappa(\bar\xi_\kappa, \bar\nu_\kappa)$, for $\kappa = 0, \dots, N-1$, starting from the initial condition $\bar\xi_0 = \xi_0$ and let

$$\bar\Phi = \sum_{k=0}^{N-1} \left(\|\bar\xi_k - \xi_k^\diamond\|_{W_k} + \|\bar\nu_k\|_{R_k} \right) + \|\bar\xi_N - \xi_N^\diamond\|_{W_N}$$

3: let $x_\kappa^\diamond = \xi_k^\diamond - \bar\xi_\kappa$, $\kappa = 0, \dots, N$

4: let $u_\kappa^\diamond = -\bar\nu_\kappa$, $\kappa = 0, \dots, N-1$

5: **for** $i = 1$ **to** S **do**

6: pick at random sufficiently small $\Delta\xi_0^{(i)}, \Delta\nu_0^{(i)}, \dots, \Delta\nu_{N-1}^{(i)}$

7: compute $\xi_{\kappa+1}^{(i)} = f_\kappa(\xi_\kappa^{(i)}, \bar\nu_\kappa + \Delta\nu_\kappa^{(i)})$, for $\kappa = 0, \dots, N-1$, starting from $\xi_0^{(i)} = \xi_0 + \Delta\xi_0^{(i)}$

8: let $x_\kappa^{(i)} = \xi_\kappa^{(i)} - \bar\xi_\kappa$, $\kappa = 0, \dots, N$

9: let $u_\kappa^{(i)} = \Delta\nu_\kappa^{(i)}$, $\kappa = 0, \dots, N-1$

10: use Algorithm 1 with the data gathered in Steps 2–9 to compute $\hat\Theta_\kappa$, $\hat\Psi_\kappa$, and $\hat\phi_\kappa$, $\kappa = 0, \dots, N-1$

11: letting $\hat\Theta_\kappa$ and $\hat\Psi_\kappa$ be partitioned as in (12), compute

$$\hat\xi_{\kappa+1} = f_\kappa\left(\hat\xi_\kappa, \bar\nu_\kappa - \hat\Theta_{\kappa,3}^{-1}\left(\hat\Theta_{\kappa,2}^\top(\hat\xi_\kappa - \bar\xi_\kappa) + \tfrac{1}{2}\hat\Psi_{\kappa,2}\right)\right),$$

$\kappa = 0, \dots, N-1$, starting from $\hat\xi_0 = \xi_0$ and let

$$\hat\Phi = \|\hat\xi_N - \xi_N^\diamond\|_{W_N} + \sum_{k=0}^{N-1} \Bigg(\|\hat\xi_k - \xi_k^\diamond\|_{W_k} + \|\bar\nu_k - \hat\Theta_{k,3}^{-1}(\hat\Theta_{k,2}^\top(\hat\xi_k - \bar\xi_k) + \tfrac{1}{2}\hat\Psi_{k,2})\|_{R_k} \Bigg)$$

12: **if** $\hat\Phi < \bar\Phi$ **then**

13: assign $\bar\nu_\kappa \leftarrow \bar\nu_\kappa - \hat\Theta_{\kappa,3}^{-1}(\hat\Theta_{\kappa,2}^\top(\hat\xi_\kappa - \bar\xi_\kappa) + \tfrac{1}{2}\hat\Psi_{\kappa,2})$, $\kappa = 0, \dots, N-1$

14: **until** $\hat\Phi < \bar\Phi$

15: **return** $\{\bar\nu_\kappa\}_{\kappa=0}^{N-1}$

Remark 6 (Locally optimal policy). Algorithm 2 can be used also to generate a policy that locally solves the optimal control problem (26), (29). In fact, letting $\{\hat{\Theta}_\kappa\}_{\kappa=0}^{N-1}$ and $\{\hat{\Psi}_\kappa\}_{\kappa=0}^{N-1}$ be the matrices computed at Step 10 of Algorithm 2 partitioned as in (12), and letting $\{\bar{\xi}_\kappa\}_{\kappa=0}^{N-1}$ be the solution to system (26) with input $\{\bar{\nu}_\kappa\}_{\kappa=0}^{N-1}$, the control policy $\pi = \{\pi_0, \dots, \pi_{N-1}\}$,

$$\pi_\kappa(\xi_\kappa) = \bar{\nu}_\kappa - \hat{\Theta}_{\kappa,3}^{-1}\left(\hat{\Theta}_{\kappa,2}^\top(\xi_\kappa - \bar{\xi}_\kappa) + \tfrac{1}{2}\hat{\Psi}_{\kappa,2}\right),$$

$\kappa = 0, \dots, N-1$, constitutes a local solution to (26), (29). △

The next remark shows how to adapt the step size in Algorithm 2 in order to improve its convergence proprieties.

Remark 7 (Levenberg-Marquardt adaptation of the step-size). Following [11], it is possible to improve the convergence of Algorithm 2 by using a method related to the Levenberg-Marquardt algorithm [41]: by defining an additional positive parameter λ and letting $\overline{\lambda}$ be its upper bound (which essentially governs the minimum step size), we can substitute Step 11 of Algorithm 2 with the following procedure:

1: compute an eigenvalue decomposition of $\hat{\Theta}_{\kappa,3} = VDV^\top$, with D being a nonnegative diagonal matrix
2: **repeat**
3: let $\Upsilon = V(D + \lambda I)V^\top$
4: letting $\hat{\Theta}_\kappa$ and $\hat{\Psi}_\kappa$ be partitioned as in (12), compute

$$\hat{\xi}_{\kappa+1} = f_\kappa\left(\hat{\xi}_\kappa, \bar{\nu}_\kappa - \Upsilon^{-1}\left(\hat{\Theta}_{\kappa,2}^\top(\hat{\xi}_\kappa - \bar{\xi}_\kappa) + \tfrac{1}{2}\hat{\Psi}_{\kappa,2}\right)\right),$$

$\kappa = 0, \dots, N-1$, starting from $\hat{\xi}_0 = \xi_0$ and let

$$\hat{\Phi} = \|\hat{\xi}_N - \xi_N^\diamond\|_{W_N} + \sum_{k=0}^{N-1}\Big(\|\hat{\xi}_k - \xi_k^\diamond\|_{W_k} + \|\bar{\nu}_k - \hat{\Theta}_{k,3}^{-1}(\hat{\Theta}_{k,2}^\top(\hat{\xi}_k - \bar{\xi}_k) + \tfrac{1}{2}\hat{\Psi}_{k,2})\|_{R_k}\Big)$$

5: **if** $\lambda < \overline{\lambda}$ **then**
6: assign $\lambda \leftarrow 2\lambda$
7: **else**
8: assign $\lambda \leftarrow \frac{1}{2}\lambda$
9: **until** $\hat{\Phi} > \bar{\Phi}$ **and** $\lambda < \overline{\lambda}$

As shown in [11], if λ is close to zero, then we have a Newton method using the true Hessian of the optimization problem, whereas, if λ is large, then the Hessian of the optimization problem is replaced by λI, that is the algorithm takes small steps in the direction of the gradient. This procedure has been proved empirically to perform better than the plain Algorithm 2 in terms of robustness and convergence speed[1]. △

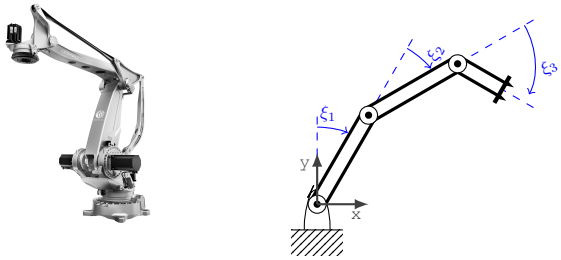
IV. Numerical example

Inspired by [22], [23], where, motivated by the uncertain nature of the plant, deep reinforcement learning methods with NAF have been applied to robot manipulators, hereafter the proposed algorithm is assessed relying on a model of a robot by Comau (see Figure 1), identified on real data.

[1]A MATLAB package implementing this procedure is available at the link:

https://github.com/Corrado-possieri/iterative_LQ_Qlearning.

At the same link, the MATLAB code that has been used to carry out the simulation reported in Section IV is made available together with other examples of application of the proposed algorithm.



(a) (b)

Fig. 1. The robot manipulator. (a) Comau industrial robot manipulator setup. (b) Schematic view of the simulated three joints robot manipulator

During our tests, for the sake of simplicity, making reference to the model identified in [42], only vertical planar motions of the robot manipulator were enabled, by locking three of the six joints. Note that the proposed algorithm is valid for any configuration of the manipulator, even in the spatial case. Hence, the dynamics of the system can be described in the joint space, by using the Lagrangian approach [43], as

$$B(\xi)\ddot{\xi} + C(\xi, \dot{\xi})\dot{\xi} + F_\mathrm{v}\dot{\xi} + F_\mathrm{s}\,\mathrm{sgn}(\dot{\xi}) + g(\xi) = \nu,$$

where $\xi \in \mathbb{R}^3$ is the vector of joint variables, $B(\xi) \in \mathbb{R}^{3\times3}$ is the inertia matrix, $C(\xi, \dot{\xi}) \in \mathbb{R}^{3\times3}$ represents centripetal and Coriolis torques, $F_\mathrm{v} \in \mathbb{R}^{3\times3}$ is the viscous friction matrix, $F_\mathrm{s} \in \mathbb{R}^{3\times3}$ is the static friction matrix, $g(\xi) \in \mathbb{R}^3$ is the vector of gravitational torques and $\nu \in \mathbb{R}^3$ represents the motors torques. The ode45 solver with input sampled each $T = 1\,\mathrm{s}$ is used also in this example. The initial position of the joints is set equal to $\xi_0 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\top$, the weights are $W_\kappa = 1 \times 10^{-6} I$, $W_N = I$ and $R_\kappa = 1 \times 10^{-12} I$, while the desired target is $\xi_k^\diamond = \begin{bmatrix} \frac{\pi}{3} & \frac{\pi}{4} & \frac{\pi}{3} \end{bmatrix}^\top$. The horizon is set equal to $N = 3$ and the initial guess is $\{\bar{\nu}_{1_\kappa}\}_{\kappa=0}^{N-1} = \{130, 160, 200\}$ for joint 1, and $\{\bar{\nu}_{2_\kappa}\}_{\kappa=0}^{N-1} = \{-100, 70, 20\}$, $\{\bar{\nu}_{3_\kappa}\}_{\kappa=0}^{N-1} = \{-50, 50, -20\}$ for joints 2 and 3, respectively. The number of experiments for Q-factors approximation is $S = 200$.

The value of the cost function Φ is minimized after few iterations (see its logarithmic value in Figure 2). In Figure 3, instead, the joint space and the corresponding velocity space are illustrated. More specifically, the dotted lines represent the case when the initial guess $\bar{\nu}_{i_\kappa}$, $i = 1, 2, 3$ is given as input. On the other hand, the solid lines refer to the case when Algorithm 2 is applied until convergence, and it can be observed that the joint positions reach the reference target, as well as velocities start from zero and, after $N - 1$ steps, are zeroed again in correspondence of the desired points.

V. Conclusions

In this paper we discussed dynamic programming solutions for linear quadratic optimal control problems in a data-driven setting. More precisely, a solution to stochastic problems in terms of Q-factors has been presented, and an approximation algorithm has been proposed. It is worth noticing that the approximation of Q-factors can be viewed as a particular case of the NAF algorithm, usually applied for deep-learning problems where the value function is approximated as a quadratic

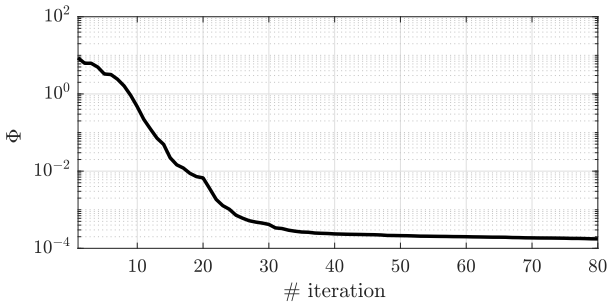


Fig. 2. Logarithmic value of the cost function Φ for the robot example

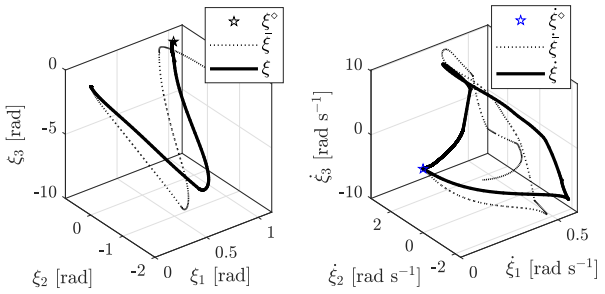


Fig. 3. Joint space and the corresponding velocity space in case of initial guess (dotted lines) and after the learning procedure (solid lines)

expression. Further, we proposed a new data-driven iterative linear quadratic control, capable of determining locally optimal solutions for a class of nonlinear tracking problems in a data-driven setting. A possible way for improving the convergence speed of the algorithm has been also suggested, and satisfactory simulated results have been obtained.

REFERENCES

[1] K. J. Åström and B. Wittenmark, *Adaptive control*. Mineola, NY, USA: Dover Publications, 2013.

[2] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal control*. Hoboken, NJ, USA: John Wiley & Sons, 2012.

[3] P. A. Ioannou and J. Sun, *Robust adaptive control*. Mineola, NY, USA: Dover Publications, 2012.

[4] D. Liberzon, *Calculus of variations and optimal control theory: a concise introduction*. Princeton, NJ, USA: Princeton University Press, 2011.

[5] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. Cambridge, MA, USA: MIT press, 2018.

[6] D. P. Bertsekas, *Dynamic programming and optimal control*. Belmont, MA, USA: Athena scientific, 2012, vol. I, II.

[7] D. E. Kirk, *Optimal control theory: an introduction*. Mineola, NY, USA: Dover Publications, 2004.

[8] P. Werbos, "Approximate dynamic programming for realtime control and neural modelling," in *Handbook of intelligent control: neural, fuzzy and adaptive approaches*, D. A. White and D. A. Sofge, Eds. New York, NY, USA: Van Nostrand, 1992, pp. 493–525.

[9] P. Dorato, V. Cerone, and C. Abdallah, *Linear-Quadratic Control: An Introduction*. Kent, OH, USA: Prentice-Hall, 1985.

[10] W. Li and E. Todorov, "Iterative linear quadratic regulator design for nonlinear biological movement systems," in *1st International Conference on Informatics in Control, Automation and Robotics*, Setúbal, Portugal, 2004, pp. 222–229.

[11] E. Todorov and W. Li, "A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems," in *American Control Conference*, vol. 1, Portland, OR, USA, 2005, pp. 300–306.

[12] W. Li and E. Todorov, "Iterative linearization methods for approximately optimal control and estimation of non-linear stochastic system," *International Journal of Control*, vol. 80, no. 9, pp. 1439–1453, 2007.

[13] D. P. Bertsekas, *Reinforcement Learning and Optimal Control*. Belmont, MA, USA: Athena scientific, 2019.

[14] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[15] F. L. Lewis, D. Vrabie, and K. G. Vamvoudakis, "Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers," *IEEE Control Systems Magazine*, vol. 32, no. 6, pp. 76–105, 2012.

[16] S. J. Bradtke, B. E. Ydstie, and A. G. Barto, "Adaptive linear quadratic control using policy iteration," in *American Control Conference*, vol. 3. Baltimore, MD, USA: IEEE, 1994, pp. 3475–3479.

[17] T. Landelius, "Reinforcement learning and distributed local model synthesis," Ph.D. dissertation, Linköping University, Linköping, Sweden, 1997.

[18] F. L. Lewis and K. 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)*, vol. 41, no. 1, pp. 14–25, 2010.

[19] B. Kiumarsi, F. L. Lewis, M.-B. Naghibi-Sistani, and A. Karimpour, "Optimal tracking control of unknown discrete-time linear systems using input-output measured data," *IEEE transactions on cybernetics*, vol. 45, no. 12, pp. 2770–2779, 2015.

[20] S. A. A. Rizvi and Z. Lin, "Output feedback Q-learning control for the discrete-time linear quadratic regulator problem," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 5, pp. 1523–1536, 2018.

[21] S. Gu, T. Lillicrap, I. Sutskever, and S. Levine, "Continuous deep Q-learning with model-based acceleration," in *International Conference on Machine Learning*, New York, NY, USA, 2016, pp. 2829–2838.

[22] B. Sangiovanni, A. Rendiniello, G. P. Incremona, A. Ferrara, and M. Piastra, "Deep reinforcement learning for collision avoidance of robotic manipulators," in *European Control Conference*, Limassol, Cyprus, 2018, pp. 2063–2068.

[23] B. Sangiovanni, G. P. Incremona, M. Piastra, and A. Ferrara, "Self-configuring robot path planning with obstacle avoidance via deep reinforcement learning," *IEEE Control Systems Letters*, vol. 5, no. 2, pp. 397–402, 2021.

[24] K. K. Tan, S. Zhao, and J. Xu, "Online automatic tuning of a proportional integral derivative controller based on an iterative learning control approach," *IET Control Theory Applications*, vol. 1, no. 1, pp. 90–96, 2007.

[25] S. Preitl, R.-E. Precup, Z. Preitl, S. Vaivoda, S. Kilyeni, and J. K. Tar, "Iterative feedback and learning control: Servo systems applications," *IFAC Proceedings Volumes*, vol. 40, no. 8, pp. 16–27, 2007, 1st IFAC Workshop on Convergence of Information Technologies and Control Methods with Power Plants and Power Systems.

[26] R.-C. Roman, R.-E. Precup, C.-A. Bojan-Dragos, and A.-I. Szedlak-Stinean, "Combined model-free adaptive control with fuzzy component by virtual reference feedback tuning for tower crane systems," *Procedia Computer Science*, vol. 162, pp. 267–274, 2019.

[27] G. C. Calafiore and C. Possieri, "Output feedback Q-learning for linear-quadratic discrete-time finite-horizon control problems," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[28] M. Zhou, Y. Yu, and X. Qu, "Development of an efficient driving strategy for connected and automated vehicles at signalized intersections: A reinforcement learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 1, pp. 433–443, 2020.

[29] G. C. Calafiore and C. Possieri, "Efficient model-free Q-factor approximation in value space via log-sum-exp neural networks," in *European Control Conference*. IEEE, 2020, pp. 23–28.

[30] J. Köhler, M. A. Müller, and F. Allgöwer, "Nonlinear reference tracking: An economic model predictive control perspective," *IEEE Transactions on Automatic Control*, vol. 64, no. 1, pp. 254–269, 2018.

[31] M. Tomizuka, "On the design of digital tracking controllers," *Journal of Dynamic Systems, Measurement, and Control*, vol. 115, no. 2B, pp. 412–418, 1993.

[32] B. D. O. Anderson and J. B. Moore, *Optimal control: linear quadratic methods*. North Chelmsford, MA, USA: Courier Corporation, 2007.

[33] F. Zhang, *The Schur complement and its applications*. New York, NY, USA: Springer, 2006.

[34] W. H. Greene, *Econometric analysis*. Upper Saddle River, NJ, USA: Prentice Hall, 2003.

[35] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE circuits and systems magazine*, vol. 9, no. 3, pp. 32–50, 2009.

[36] S. Dean, H. Mania, N. Matni, B. Recht, and S. Tu, "On the sample complexity of the linear quadratic regulator," *Foundations of Computational Mathematics*, 2019.

[37] C. D. Meyer, *Matrix analysis and applied linear algebra*. Philadelphia, PA, USA: Siam, 2000.
[38] Y. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*. Philadelphia, PA, USA: Society for Industrial Mathematics, 1995.
[39] H. Kushner and G. G. Yin, *Stochastic approximation and recursive algorithms and applications*, 2nd ed. New York, NY, USA: Springer, 2003, vol. 35.
[40] T. Yamamoto, "Historical developments in convergence analysis for Newton's and Newton-like methods," *Journal of Computational and Applied Mathematics*, vol. 124, no. 1-2, pp. 1–23, 2000.
[41] D. W. Marquardt, "An algorithm for least-squares estimation of non-linear parameters," *Journal of the society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.
[42] A. Calanca, L. M. Capisani, A. Ferrara, and L. Magnani, "MIMO closed loop identification of an industrial robot," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 5, pp. 1214–1224, 2011.
[43] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics-Modelling, Planning and Control*, 3rd ed. London, UK: Springer-Verlag, 2009.