Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Data-Driven Control of Unknown Systems: A Linear Programming Approach [★]

Alexandros Tanzanakis [*] John Lygeros [*]

[*] *Department of Information Technology and Electrical Engineering, ETH Zurich, Switzerland, (e-mail: {atanzana,jlygeros}@ethz.ch).*

Abstract: We consider the problem of discounted optimal state-feedback regulation for general unknown deterministic discrete-time systems. It is well known that open-loop instability of systems, non-quadratic cost functions and complex nonlinear dynamics, as well as the on-policy behavior of many reinforcement learning (RL) algorithms, make the design of model-free optimal adaptive controllers a challenging task. We depart from commonly used least-squares and neural network approximation methods in conventional model-free control theory, and propose a novel family of data-driven optimization algorithms based on linear programming, off-policy Q-learning and randomized experience replay. We develop both policy iteration (PI) and value iteration (VI) methods to compute an approximate optimal feedback controller with high precision and without the knowledge of a system model and stage cost function. Simulation studies confirm the effectiveness of the proposed methods.

Keywords: linear programming, Q-learning, approximate dynamic programming, data-driven control.

1. INTRODUCTION

Reinforcement Learning (RL) bridges the gap between model-based and model-free control. This is accomplished by optimizing policies of (possibly) unknown dynamical systems with the goal of maximizing or minimizing a long-term reward function. The derivation of the Q-learning algorithm (Watkins, 1989), (Bradtke et al., 1994), along with Approximate Dynamic Programming (ADP) methods such as neurodynamic programming (Bertsekas and Tsitsiklis, 1996) were among the first approaches that dealt effectively with the problem of model-free optimal adaptive control.

To address the challenges associated with solving the Bellman and Hamilton-Jacobi-Bellman equations in model-based and model-free optimal control (Lewis et al., 2012b), reliable ADP methods have been developed (Powell, 2011). These equations can be approximately solved by utilizing a family of iterative methods known as Policy Iteration (PI) and Value Iteration (VI) (Bertsekas, 2017), (Lewis et al., 2012a).

In the Q-learning setting, an Actor-Critic framework (Kiumarsi et al., 2018), (Konda and Tsitsiklis, 2010) is commonly employed to approximate the Q-function and control policy with appropriate parametric function models (i.e. models with a priori fixed number of basis elements). For the computation of an approximate control policy, two fundamental learning schemes are used. In on-policy learning (Wei et al., 2015), (Kiumarsi et al., 2014), a single control policy is used for both generation of training data from the system and for online policy evaluation and improvement. In off-policy learning (Li et al., 2018), (Li et al., 2019), a behavior control policy is applied to the system and is responsible for the generation of training data samples, while a target control policy is iteratively evaluated and updated online. In the off-policy setting, a highly promising approach called Experience Replay (Adam et al., 2012), (Liu et al., 2014), (Zha et al., 2019) is often employed to resolve the critical sample inefficiency issue of many learning algorithms, where real data samples are gathered and used only in a specific learning iteration, after the end of which they are discarded. PI and VI can be implemented in both schemes by applying either least-squares or neural network approximation methods.

The Linear Programming (LP) approach to ADP is an alternative, model-based optimization paradigm to approximate the Value function (Hernandez-Lerma and Lasserre, 1996), (Wang et al., 2015) or the Q-function (Beuchat et al., 2020), (Cogill et al., 2006) and control policy, with rigorous theoretical guarantees on the approximation quality and online performance. For the model-free setting, (Banjac and Lygeros, 2019) proposed an on-policy Q-learning-based PI LP algorithm for discounted state-feedback regulation of deterministic linear time-invariant (LTI) systems, utilizing a heuristic based on support constraints.

Here, we propose a novel family of off-policy Q-learning-based LP algorithms for reliable discounted state-feedback regulation of general unknown deterministic discrete-time systems. The PI and VI methods are reformulated as data-driven, finite-dimensional linear programs, which proceed with the computation of an approximate optimal feedback controller without the knowledge of a system model and stage cost function. As a consequence, unlike the method proposed in (Banjac and Lygeros, 2019), our methods inherit the convergence guarantees of the standard PI

[★] This research work was supported by the European Research Council (ERC) under the project OCAL, grant number 787845.

and VI algorithms. To cope with the sample exploitation problem, we utilize a simple yet highly effective off-policy learning scheme called Randomized Experience Replay. To the best of our knowledge, this the first work in the related LP literature that provides a unified approach for effective model-free control of both deterministic discrete-time LTI and nonlinear systems.

The rest of this paper is organized as follows. The technical preliminaries are derived in Section 2. The proposed family of data-driven optimization algorithms is presented and discussed in Section 3. Simulation studies are carried on Section 4 and conclusions are given in Section 5.

Notation. $\mathbb{N}$ denotes the set of natural numbers excluding 0. $\mathbb{R}$ defines the set of real numbers, while $\mathbb{R}_+$ the set of real numbers which are greater or equal than 0. $\mathbb{S}^n$ denotes the set of $n \times n$ symmetric matrices, while $\mathbb{S}^n_{++}$ the set of $n \times n$ symmetric positive definite matrices. $I_{n\times n}$ denotes an $n\times n$ identity matrix. $tr(A)$ defines the trace of a matrix $A \in \mathbb{R}^{n\times n}$.

2. TECHNICAL PRELIMINARIES

2.1 The problem of discounted state-feedback regulation

Consider a deterministic discrete-time system of the form

$$x_{k+1} = f(x_k, u_k), \tag{1}$$

where $x_k \in \mathcal{X} \subseteq \mathbb{R}^n$, $u_k \in \mathcal{U} \subseteq \mathbb{R}^m$ and $f : \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ the model dynamics at time step k. The main objective of discounted state-feedback regulation is the computation of a control policy $\mu : \mathcal{X} \to \mathcal{U}$ that minimizes the cost

$$J^\mu(x_0) = \sum_{k=0}^{\infty} \gamma^k l\big(x_k, \mu(x_k)\big), \tag{2}$$

where $\gamma \in (0, 1)$ is the discount factor and $l : \mathcal{X} \times \mathcal{U} \to \mathbb{R}_+$ is the stage cost. To ensure the problem is well-posed, we assume that there exists a policy μ such that $J^\mu(x) < +\infty$ for all $x \in \mathcal{X}$. We are interested in the case where the functions l and f are not known, but their values can be observed for specific instances of x and u, for example by sampling, simulating, or experimenting with the system. This is the typical setting in Q-learning.

2.2 The optimal Q-function

The optimal policy that minimizes J^μ can be computed if one has access to the optimal Q-function

$$Q^\star(x, u) = l(x, u) + \inf_{\mu} \sum_{k=1}^{\infty} \gamma^k l(x_k, \mu(x_k)). \tag{3}$$

It can be shown that $Q^\star$ satisfies the Bellman equation

$$Q^*(x, u) = \underbrace{l(x, u) + \gamma \min_{v} Q^*\big(x', v\big)}_{FQ^*(x,u)}, \tag{4}$$

where $x' = f(x, u)$ and F is the Bellman operator for Q-functions. F can be shown to be monotone and contractive (Cogill et al., 2006). Once $Q^\star$ is available, the optimal policy can then be computed by

$$\mu^*(x) = \underset{v}{\operatorname{argmin}} Q^*(x, v). \tag{5}$$

To obtain the linear programming formulation, one starts by relaxing the Bellman equation (4) to the Bellman inequality (Beuchat et al., 2020), (Cogill et al., 2006). An exact LP-based reformulation of (4) is then given by

$$\begin{aligned} \max_{Q\in\mathcal{F}(\mathcal{X},\mathcal{U})} \quad & \int_{\mathcal{X}\times\mathcal{U}} Q(x, u)c\big(d(x, u)\big) \\ \text{s.t.} \quad & Q(x, u) \le l(x, u) + \gamma Q\big(x', v\big) \\ & \forall (x, u, v) \in \mathcal{X} \times \mathcal{U}^2, \end{aligned} \tag{6}$$

where $\mathcal{F}$ is the space of measurable functions bounded in an appropriate norm.

Lemma 1 (Beuchat et al., 2020), (Cogill et al., 2006). If $Q^\star \in \mathcal{F}(\mathcal{X},\mathcal{U})$, then the maximizer of (6) is identical to the solution of (4), for c almost all $(x, u) \in \mathcal{X} \times \mathcal{U}$.

An important condition for the equivalence between (4) and (6) is that $\mathcal{F}(\mathcal{X},\mathcal{U})$ contains the optimal Q-function, i.e. the inequalities of problem (6) can be satisfied with equality. The quantity $c(\cdot,\cdot)$ is called state-action relevance weight. State-action relevance weights are finite measures and typically allocate a positive mass to all open subsets of $\mathcal{X} \times \mathcal{U}$ (Beuchat et al., 2020).

The solution of (6) is intractable in general. The main difficulties are summarized as follows (Beuchat et al., 2020):

i) The dimension of the space $\mathcal{F}(\mathcal{X},\mathcal{U})$ can be high (if $\mathcal{X}$ and $\mathcal{U}$ are finite) or even infinite (if they are not, the case of interest here).
ii) The number of inequality constraints of optimization problem (6) can be high or infinite.
iii) For an arbitrary $Q^* \in \mathcal{F}(\mathcal{X},\mathcal{U})$, the control policy calculation (5) may be intractable.

These difficulties describe the curse of dimensionality issue in ADP for our problem setting. To tackle difficulty i), we construct a restricted function space $\hat{\mathcal{F}}(\mathcal{X} \times \mathcal{U})$ (Beuchat et al., 2020), (Wang et al., 2015) of linear combinations of basis functions $\hat{Q}_j(\cdot,\cdot), j = 1, \ldots, K$,

$$\hat{\mathcal{F}}(\mathcal{X} \times \mathcal{U}) = \{Q(\cdot,\cdot)|Q(x, u) = \alpha^T \hat{Q}(x, u)\}, \tag{7}$$

where $\alpha \in \mathbb{R}^K$ and $\hat{Q}(x, u) = \big[\hat{Q}_1(x, u), \ldots, \hat{Q}_K(x, u)\big] : \mathcal{X} \times \mathcal{U} \to \mathbb{R}^K$. An approximate solution of (4) can then be computed by solving

$$\begin{aligned} \max_{Q\in\hat{\mathcal{F}}(\mathcal{X}\times\mathcal{U})} \quad & \int_{\mathcal{X}\times\mathcal{U}} Q(x, u)c\big(d(x, u)\big) \\ \text{s.t.} \quad & Q(x, u) \le l(x, u) + \gamma Q\big(x', v\big) \\ & \forall (x, u, v) \in \mathcal{X} \times \mathcal{U}^2. \end{aligned} \tag{8}$$

The optimizer $\hat{Q}^\star$ of (8) defines a policy

$$\hat{\mu}^*(x) = \underset{v}{\operatorname{argmin}} \hat{Q}^*(x, v). \tag{9}$$

The hope is that, if we choose the space $\hat{\mathcal{F}}$ well enough, $\hat{Q}^\star$ will be a good approximation of $Q^\star$, so hopefully the performance of $\hat{\mu}^\star$ will be similar to that of $\mu^\star$.

To tackle difficulty (ii), $\hat{\mathcal{F}}(\mathcal{X},\mathcal{U})$ must be constructed based on suitable basis function elements, according to a particular application setting. For example, one can use quadratic functions and the S-procedure or polynomial optimization methods to derive a tight approximation of the infinite inequality constraints in (6). Likewise, one may want to restrict attention to basis functions that are convex in u to deal with difficulty iii).

In general, the solution of (8) will also depend on the choice of the state-action relevance weight $c(\cdot,\cdot)$. The work of (Beuchat et al., 2016) proposed solving the approximate LP problem (8) for multiple realizations of $c(\cdot,\cdot)$ and then computing a pointwise maximum over all derived solutions.

We note that in the case that $Q^\star \in \hat{\mathcal{F}}(\mathcal{X}\times\mathcal{U})$, then the solution of the approximate LP problem (8) is $Q^\star$, as long as $c(\cdot,\cdot)$ allocates positive mass to all open subsets of $\mathcal{X}\times\mathcal{U}$. However, the standard LP approach to ADP requires both the system dynamics (1) and stage cost function to be known. In the following sections we will show how to bypass this requirement by deriving novel, data-driven LP variants of the well-known PI and VI methods.

2.3 PI and VI in Q-learning

In RL/ADP, the optimal Q-function and control policy are approximated online. In Q-learning, the Bellman equation (4) can be approximately solved using an iterative family of methods called PI and VI (Bertsekas, 2017), (Lewis et al., 2012a). PI requires an initial control policy $\mu^0(x)$ such that $J^{\mu^0}(x) < \infty$ for all $x \in \mathcal{X}$. At iteration $i \geq 0$, it proceeds with the following two successive steps until convergence of the Q-function:

i) **PI - Policy Evaluation Step:** Solve for Q^i,

$$Q^i(x,u) = l(x,u) + \gamma Q^i\big(x', \mu^i(x')\big). \tag{10}$$

ii) **PI - Policy Improvement/Update Step:**

$$\mu^{i+1}(x) = \underset{v}{\operatorname{argmin}} Q^i(x,v). \tag{11}$$

It can be shown that PI provides non-increasing, monotone convergence to the optimal Q-function and control policy for both undiscounted and discounted optimal control problems. PI provides fast convergence to the optimal Q-function and control policy, although it requires an initial stabilizing policy, which is challenging in many cases to compute (Wei et al., 2018, Chapter 4), (Luo et al., 2016), (Heydari, 2016).

VI can be initialized with an arbitrary $Q^0(x,u) \geq 0$ and control policy $\mu^0(x)$. At iteration $i \geq 0$, it proceeds with the following two steps until convergence of the Q-function:

i) **VI - Policy Evaluation Step:** Solve for Q^{i+1},

$$Q^{i+1}(x,u) = l(x,u) + \gamma Q^i\big(x', \mu^i(x')\big). \tag{12}$$

ii) **VI - Policy Improvement/Update Step:**

$$\mu^{i+1}(x) = \underset{v}{\operatorname{argmin}} Q^{i+1}(x,v). \tag{13}$$

VI also provides theoretical guarantees related to monotonicity and convergence to the optimal Q-function and control policy. However, the convergence speed of VI is much slower than the one of PI (Luo et al., 2018), (Heydari, 2016). Finally, we note that the policy evaluation step for VI (12) is just a simple recursion and not an equation as in PI (10) .

3. THE PROPOSED Q-PI-LP AND Q-VI-LP ALGORITHMS

In this section, we present a novel family of data-driven, off-policy Q-learning based optimization algorithms, called Q-PI-LP and Q-VI-LP. Algorithms 1 and 2 show the proposed methods.

In the off-policy setting, two types of control policies are defined: *i)* a behavior policy, a, which is used for training data generation and *ii)* a target policy, $\mu(x)$, which is successively evaluated and improved. To avoid confusion, we refer to the policy evaluation and improvement steps for PI and VI as target policy evaluation and improvement.

Algorithm 1 Q-PI-LP algorithm.

```
1: Select threshold parameter $\epsilon > 0$ and buffer size $N \in \mathbb{N}$.
2: Experience Replay Buffer: Construct buffer $\mathcal{B} = \{(x_b, a_b, y_b, l_b)\}_{b=1}^N$, where $(x_b, a_b)$ are random state-behavior policy sample pairs, $y_b$ is computed by applying $(x_b, a_b)$ to the unknown system (1) and $l_b$ is the measurement of the resulting stage cost.
3: Pick $\hat{\mu}^0(x)$ such that $J^{\hat{\mu}^0}(x) < \infty$ for all $x \in \mathcal{X}$.
4: Set $i = 0$.
5: Solve the LP problem,
       $\max_{Q^i \in \hat{\mathcal{F}}(\mathcal{X}\times\mathcal{U})} \int_{\mathcal{X}\times\mathcal{U}} Q^i(x,a) c\big(d(x,a)\big)$
       s.t. $Q^i(x_b,a_b) \leq l_b + \gamma Q^i\big(y_b, \hat{\mu}^i(y_b)\big)$
            for $b = 1,\ldots,N$.                                        (14)
6: Update $\hat{\mu}^{i+1}(x) = \underset{v}{\operatorname{argmin}} Q^i(x,v)$.
7: If $i \geq 1$ and $|Q^i(x_b,a_b) - Q^{i-1}(x_b,a_b)| \leq \epsilon$ for all $b$, then terminate; else set $i = i+1$, go to Step 5 and continue.
```

Algorithm 2 Q-VI-LP algorithm.

```
1: Select threshold parameter $\epsilon > 0$ and buffer size $N \in \mathbb{N}$.
2: Experience Replay Buffer: Construct Experience Replay Buffer as in Algorithm 1.
3: Choose $Q^0(x,a) \geq 0$ arbitrary.
4: Compute $\hat{\mu}^0(x) = \underset{v}{\operatorname{argmin}} Q^0(x,v)$.
5: Set $i = 0$.
6: Solve the LP problem,
       $\max_{Q^{i+1} \in \hat{\mathcal{F}}(\mathcal{X}\times\mathcal{U})} \int_{\mathcal{X}\times\mathcal{U}} Q^{i+1}(x,a) c\big(d(x,a)\big)$
       s.t. $Q^{i+1}(x_b,a_b) \leq l_b + \gamma Q^i\big(y_b, \hat{\mu}^i(y_b)\big)$
            for $b = 1,\ldots,N$.                                        (15)
7: Update $\hat{\mu}^{i+1}(x) = \underset{v}{\operatorname{argmin}} Q^{i+1}(x,v)$.
8: If $|Q^{i+1}(x_b,a_b) - Q^i(x_b,a_b)| \leq \epsilon$ for all $b$, then terminate; else set $i = i+1$, go to Step 6 and continue.
```

3.1 Randomized Experience Replay (RER)

An important challenge in model-free optimal adaptive control is the satisfaction of the persistance of excitation

(PoE) condition (Tao, 2003). PoE is required to ensure optimal parameter convergence and is generally guaranteed by designing appropriate probing noise, which is added to the control policy. However, in on-policy learning, the use of such probing noise can bias solutions (Li et al., 2019).

Off-policy learning provides three convenient ways to reduce or even eliminate any bias of solutions under PoE (Luo et al., 2017):

i) Construction of an offline batch of training data of a sufficiently large size N. This batch of data will be repeatedly used in the learning phase of an underlying off-policy algorithm. This family of off-policy learning methods, called Experience Replay (ER) (Adam et al., 2012), (Liu et al., 2014), (Zha et al., 2019), has been shown to improve the convergence rate, computational efficiency and stability of a learning algorithm.
ii) Utilization of arbitrary pairs of states and behavior policies to be applied to the unknown system (1), enabling rich data exploration.
iii) Derivation of behavior policies which are pure PoE signals and are not superimposed on any policy, e.g. probabilistic noise or sinusoidal signals with random frequencies.

Here, we utilize a simple yet highly effective off-policy learning scheme which exploits these important properties of off-policy learning, called Randomized Experience Replay (RER). RER proceeds with the construction of a rich offline batch of data tuples (x_b, a_b, y_b, l_b), with $b = 1, \ldots, N$, and $y_b = f(x_b, a_b)$, called Experience Replay Buffer. This buffer remains fixed and is repeatedly used during every iteration i for the target control policy evaluation step of Q-PI-LP and Q-VI-LP.

For each tuple in the buffer, arbitrary states x_b and behavior policies a_b are sampled from appropriate probability distributions. Related work involving a randomized selection of states and policies was proposed in (Falsone and Prandini, 2015), (Petretti and Prandini, 2014) and (Esfahani et al., 2018), but only for model-based ADP. To replicate a realistic learning scenario, we assume that the functional form of the dynamics, f, and stage cost, l, are unknown but can be sampled for particular states and inputs (using, for example, a simulator or experiment). We then set $y_b = f(x_b, a_b)$ and $l_b = l(x_b, a_b)$ for $b = 1, \ldots, N$. We note that the algorithms do have access to the data in the buffer, including y_b and l_b, but not to the functions f and l.

3.2 PI and VI as data-driven, finite dimensional LPs

Based on Sections 2.2 and 2.3, as well as the discussion on off-policy learning in the current section, we can similarly reformulate the target policy evaluation steps of PI and VI as the data-driven LP problems (14) and (15) respectively. As shown in Section 2.3, the target policy evaluation steps of PI (10) and VI (12) involve a set of equations and recursions respectively. Therefore, the associated finite-dimensional LPs (14), (15) involve equation inequality constraints and recursion inequality constraints respectively.

4. SIMULATION STUDIES

4.1 A four-dimensional open-loop unstable LTI system

Consider a four-dimensional discrete-time LTI system,

$$x_{k+1} = Ax_k + Bu_k,$$

where

$$A = \begin{bmatrix} 1.8 & -0.77 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \mathcal{X} = \mathbb{R}^4, \mathcal{U} = \mathbb{R}.$$

The open-loop eigenvalues of the system are

$$[-0.4236 \pm 0.6048i, 0.7902, 1.8569],$$

and therefore the system is unstable. We use a discount factor $\gamma = 0.9$ and a quadratic stage cost function defined as $l(x_k, u_k) = x_k^T E x_k + u_k^T F u_k$, where $E = I_{4\times4}$ and $F = 1$. In this case, the optimal Q-function is quadratic in the states and inputs,

$$Q^\star(x_k, u_k) = \begin{bmatrix} x_k \\ u_k \end{bmatrix}^T P^\star \begin{bmatrix} x_k \\ u_k \end{bmatrix},$$

where $P^\star \in \mathbb{S}^5_{++}$. Moreover, $P^\star$ can be partitioned into submatrices

$$P^\star \equiv \begin{bmatrix} P^\star_{xx} & P^\star_{xu} \\ P^\star_{ux} & P^\star_{uu} \end{bmatrix},$$

in the obvious way (Bradtke et al., 1994). Therefore, the optimal target policy is given by (5),

$$\mu^*(x) = \underset{v}{\text{argmin}} Q^*(x, v) = -(P^\star_{uu})^{-1}(P^\star_{ux} x).$$

To test the ability of the proposed algorithms to compute the optimal Q-function, we will consider the larger family of extended quadratic functions (Barratt and Boyd, 2018),

$$\hat{\mathcal{F}}(\mathcal{X} \times \mathcal{U}) = \{Q(\cdot, \cdot) | Q(x, u) = \begin{bmatrix} x \\ u \end{bmatrix}^T \hat{P} \begin{bmatrix} x \\ u \end{bmatrix} + \hat{p} \begin{bmatrix} x \\ u \end{bmatrix} + \hat{s}\}$$

which includes $Q^\star$. Here, $\hat{P} \in \mathbb{R}^{5\times5}$, $\hat{p} = [\hat{p}_x \ \hat{p}_u] \in \mathbb{R}^5$ with $\hat{p}_x \in \mathbb{R}^4, \hat{p}_u \in \mathbb{R}$ and $\hat{s} \in \mathbb{R}$.

Of course for the deterministic LQR problem at hand we know a priori that $p^\star = 0_{1\times5}$ and $s^\star = 0$. The algorithms, however, do not know that the underlying system is linear and the cost quadratic, and we want to check whether they can guess so. Note that non-zero p and s may be needed for other classes of systems (Wang et al., 2015). In addition, the state-action relevance weight $c(\cdot, \cdot)$ is considered a probability measure with first and second moments $\mu_c = 0_{5\times1}$ and $\Sigma_c = I_{5\times5}$ respectively. Therefore, the objective function of the LP problems for both Q-PI-LP and Q-VI-LP reduces to (Beuchat et al., 2020),

$$\int_{\mathcal{X}\times\mathcal{U}} Q(x, a) c\big(d(x, a)\big) = tr(\hat{P}\Sigma_c) + \hat{s}.$$

Randomized Experience Replay is implemented by constructing a buffer of $N = 7000$ tuples of $(x_b, a_b, y_b, l_b), b = 1 \ldots, 7000$, where $x_b \sim Uni(-5, 5)$ and $a_b \sim \mathcal{N}(0, 9)$. We initialize Q-PI-LP with the stabilizing target policy $\hat{\mu}^0(x) = [-0.9 \ -0.7 \ -0.5 \ -0.1]\, x$, which ensures that $J^{\hat{\mu}^0}(x) < \infty$ for all $x \in \mathbb{R}^4$. For Q-VI-LP, on the other hand, we consider the following subcases: A) $\hat{P}^0 = I_{5\times5}$, $\hat{p}^0 = 0_{1\times5}$ and $\hat{s}^0 = 0$, for which we get the non-stabilizing target policy $\hat{\mu}^0(x) = 0$ for all $x \in \mathbb{R}^4$, and B) $\hat{P}^0 = I_{5\times5}$,

$\hat{p}^0 = 0_{1\times 5}$ and $\hat{s}^0 = 0$, although we apply the initial stabilizing target policy of Q-PI-LP.

Figure 1 shows the performance of the proposed Q-PI-LP and Q-VI-LP algorithms for this LTI example. The performance is assessed based on the error between two successive Q-function estimates, specifically by computing $||\hat{P}^i - \hat{P}^{i-1}||_\infty$, $||\hat{p}^i - \hat{p}^{i-1}||_\infty$ and $|\hat{s}^i - \hat{s}^{i-1}|$. The threshold parameter is set to $\epsilon = 10^{-13}$. Q-PI-LP requires only 8 iterations to converge, while subcases A and B of Q-VI-LP require 71 iterations and 35 iterations to converge respectively. Therefore, the initialization of Q-VI-LP with a stabilizing target policy as in Q-PI-LP boosts convergence speed compared to the initialization with a non-stabilizing target policy. We then compute the error between the converged elements and the optimal ones which are computed by solving the related discrete-time algebraic Riccati equation, i.e. $||\hat{P}^\star - P^\star||_\infty$, $||\hat{p}^\star - p^\star||_\infty$ and $|\hat{s}^\star - s^\star|$, which in all cases are less or equal than 10^{-14}. Hence, the proposed algorithms provide reliable convergence to $Q^\star$ for discrete-time LTI systems.

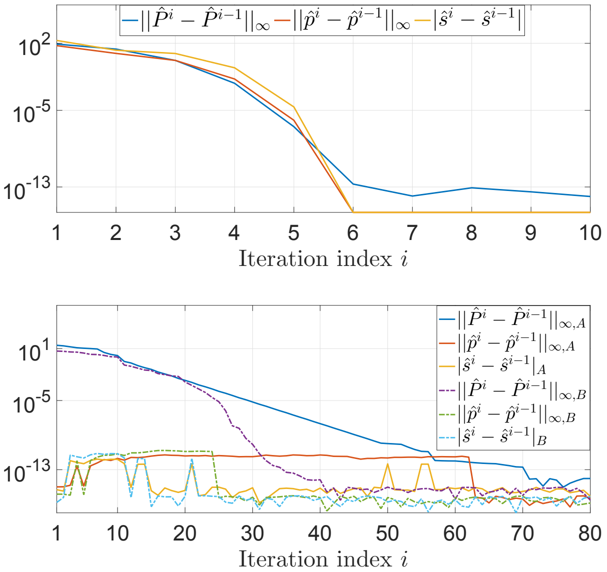

Fig. 1. Performance of Q-PI-LP (top) and Q-VI-LP (bottom) on the four-dimensional LTI example.

4.2 A two-dimensional nonlinear system

Consider the following two-dimensional discrete-time nonlinear system (Luo et al., 2017),

$$x_{k+1} = \begin{bmatrix} (x_{1,k} + x_{2,k}^2 + u_k)\cos(x_{2,k}) \\ 0.5(x_{1,k}^2 + x_{2,k} + u_k)\sin(x_{2,k}) \end{bmatrix},$$

where $\mathcal{X} = \mathbb{R}^2, \mathcal{U} = \mathbb{R}$. The following two cases are considered: i) The case of a quadratic cost function $l(x_k, u_k) = x_k^T E x_k + u_k^T F u_k$, where $E = I_{2\times 2}$ and $F = 1$, as considered in (Luo et al., 2017), and ii) the case of a nonquadratic cost function $l(x_k, u_k) = \ln\left(x_k^T E x_k + \exp(x_k^T E x_k) u_k^T F u_k + 1\right)$, where $E = I_{2\times 2}$ and $F = 1$. In both cases, the discount factor $\gamma = 0.95$. Randomized Experience Replay is implemented by constructing a buffer of $N = 3000$ tuples of $(x_b, a_b, y_b, l_b), b = 1\ldots, 3000$, where $x_b \sim Uni(-5, 5)$ and $a_b \sim \mathcal{N}(0, 1)$. Furthermore, the following family of quartic Q-functions is considered,

$$\hat{\mathcal{F}}(\mathcal{X}\times\mathcal{U}) = \{Q(\cdot,\cdot)|Q(x,u) = \begin{bmatrix} x \\ x^2 \\ u \end{bmatrix}^T \hat{P} \begin{bmatrix} x \\ x^2 \\ u \end{bmatrix} \},$$

where

$$\hat{P} \equiv \begin{bmatrix} \hat{P}_{xx} & \hat{P}_{xu} \\ \hat{P}_{ux} & \hat{P}_{uu} \end{bmatrix} \in \mathbb{R}^{5\times 5},$$

and all submatrices of $\hat{P}$ have identical dimensions with the ones from the simulation study in Section 4.1. The state-action relevance weight $c(\cdot,\cdot)$ is considered a probability measure. The additional term x^2 in the representation of the Q-function requires the first, second, third and fourth moments of $c(\cdot,\cdot)$ to appear in the objective function of the LP problems for both Q-PI-LP and Q-VI-LP. Considering the first moment as $\mu_c = 0_{3\times 1}$, the objective function for both algorithms reduces to (Beuchat et al., 2020),

$$\int_{\mathcal{X}\times\mathcal{U}} Q(x,a)c\big(d(x,a)\big) = tr(\hat{P}_1\Sigma_c) + \hat{p}_2^T\phi_c + \hat{p}_3^T\psi_c,$$

where $\hat{P}_1 \in \mathbb{R}^{3\times 3}$, $\hat{p}_2 \in \mathbb{R}^{12}$ and $\hat{p}_3 \in \mathbb{R}^4$ are elements of the $\hat{P}$ matrix with second, third and fourth moments given by $\Sigma_c \in \mathbb{S}^3$, $\phi_c \in \mathbb{R}^{12}$ and $\psi_c \in \mathbb{R}^4$ respectively. For this simulation study, $\Sigma_c = I_{3\times 3}$, $\phi_c = 1_{12\times 1}$ and $\psi_c = 1_{4\times 1}$. Q-PI-LP is initialized with the stabilizing target policy $\hat{\mu}^0(x) = [-1.5\ 0.5\ 0\ 0]\begin{bmatrix} x \\ x^2 \end{bmatrix}$ as in (Luo et al., 2017), which ensures that $J^{\hat{\mu}^0}(x) < \infty$ for all $x \in \mathbb{R}^2$. In Q-VI-LP, on the other hand, we consider the following subcases: A) $\hat{P}^0 = 0_{5\times 5}$, where we select the initial target policy $\hat{\mu}^0(x) = 0$ for all $x \in \mathbb{R}^2$, and B) $\hat{P}^0 = 0_{5\times 5}$, although we apply the above stabilizing target policy of (Luo et al., 2017).

Figures 2 and 3 show the performance of the proposed Q-PI-LP and Q-VI-LP algorithms for the cases of the quadratic and nonquadratic cost functions respectively. The performance is assessed based on the error between two successive Q-function estimates, specifically by computing the element-wise infinity norm $||\hat{P}^i - \hat{P}^{i-1}||_\infty$. The threshold parameter is set to $\epsilon = 10^{-17}$. Since $Q^\star$ cannot be computed analytically for this simulation study, the state and control trajectories under the $\hat{Q}^\star$ upon convergence are also shown.

For the case of the quadratic cost function (Figure 2), Q-PI-LP requires 8 iterations to converge, while cases A and B of Q-VI-LP require 104 iterations and 62 iterations to converge respectively. Therefore, the initialization of Q-VI-LP with a stabilizing target policy as in Q-PI-LP boosts convergence speed compared to the simple choice of an initial zero Q-function and target policy. Both algorithms converge to the following matrix,

$$\hat{P}^\star = \begin{bmatrix} 1.1154 & -0.0101 & 0.0288 & 0.0097 & 0.6390 \\ -0.0101 & 1.1195 & 0.0667 & 0.0209 & 0.0617 \\ 0.0288 & 0.0667 & 0.0023 & 0.0045 & -0.0305 \\ 0.0097 & 0.0209 & 0.0045 & -4\cdot 10^{-4} & -0.2880 \\ 0.6390 & 0.0617 & -0.0305 & -0.2880 & 1.0157 \end{bmatrix},$$

with the associated target policy,

$$\hat{\mu}^\star(x) = [-0.6292\ -0.0608\ 0.0301\ 0.2836]\begin{bmatrix} x \\ x^2 \end{bmatrix},$$

which is identical to the one computed by the state-of-art neural network-based policy gradient ADP method in (Luo et al., 2017). Furthermore, the state and control trajectories with $x_0 = \begin{bmatrix} x_{1,0} \\ x_{2,0} \end{bmatrix} = \begin{bmatrix} 1.8 \\ 1 \end{bmatrix}$ confirm that the proposed algorithms provide reliable state-feedback regulation to the origin.

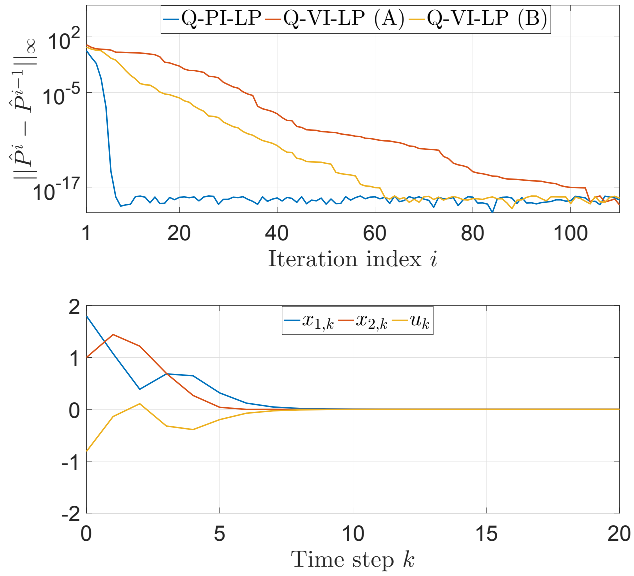

Fig. 2. Performance of Q-PI-LP and Q-VI-LP (top) and state-control trajectories under the $\hat{Q}^\star$ upon convergence (bottom) on the 2D nonlinear example with a quadratic cost function.

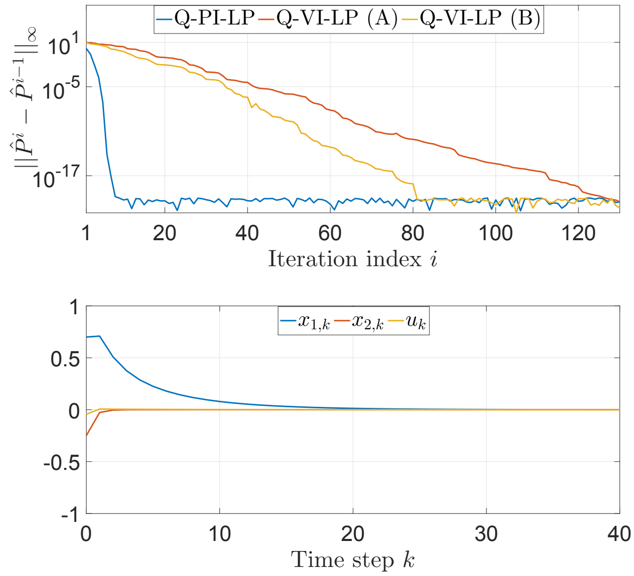

Fig. 3. Performance of Q-PI-LP and Q-VI-LP (top) and state-control trajectories under the $\hat{Q}^\star$ upon convergence (bottom) on the 2D nonlinear example with a nonquadratic cost function.

Finally, for the case of the nonquadratic cost function (Figure 3), Q-PI-LP requires 9 iterations to converge, while cases A and B of Q-VI-LP require 113 iterations and 76 iterations to converge respectively. Similarly, we observe that the initialization of Q-VI-LP with a stabilizing target policy as in Q-PI-LP boosts convergence speed compared to the choice of an initial zero Q-function and target policy. Both algorithms converge to the following matrix,

$$\hat{P}^\star = \begin{bmatrix} 0.6435 & 0.0682 & 0.0259 & -0.0131 & 0.0329 \\ 0.0682 & 0.6310 & 0.1173 & 0.0190 & 0.1450 \\ 0.0259 & 0.1173 & 0.0146 & 0.0044 & 0.0451 \\ -0.0131 & 0.0190 & 0.0044 & 0.0034 & 0.0051 \\ 0.0329 & 0.1450 & 0.0451 & 0.0051 & 0.2107 \end{bmatrix},$$

with the associated target policy,

$$\hat{\mu}^\star(x) = [-0.1561 \ -0.6881 \ -0.2140 \ -0.0242] \begin{bmatrix} x \\ x^2 \end{bmatrix}.$$

The state and control trajectories with $x_0 = \begin{bmatrix} x_{1,0} \\ x_{2,0} \end{bmatrix} = \begin{bmatrix} 0.7 \\ -0.25 \end{bmatrix}$ similarly show the superior capabilities of Q-PI-LP and Q-VI-LP in providing effective state-feedback regulation for nonlinear systems. Based on the theroretical discussions in Sections 2 and 3 and the conducted simulation studies in the current section, we observe that Q-PI-LP and Q-VI-LP inherit the monotonicity and convergence guarantees of the conventional PI and VI algorithms respectively.

We finally note that, in the data-driven on-policy PI method of (Banjac and Lygeros, 2019), an additional constraint, $\hat{P}_{uu} \succ \tau I$, with $\tau \in \mathbb{R}_+$ a sufficiently small constant and I an identity matrix of appropriate dimensions, is added to the related LP problem, to ensure invertability of $\hat{P}_{uu}$ required for the computation of the target policy $\hat{\mu}(x)$. In our family of algorithms, this additional constraint is not necessary, since the objective function of the LP problems on Q-PI-LP and Q-VI-LP is observed to enforce the derived $\hat{P}$ to be positive definite when required, at every iteration i, as long as $\Sigma_c \succ 0$.

5. CONCLUSIONS

In this paper, we have successfully extended the well-established model-based LP approach to ADP to the critical model-free setting. By utilizing off-policy Q-learning, RER and data-driven LP variants of PI and VI methods, we have derived novel, high-performance optimization algorithms which provide effective discounted state-feedback regulation of general unknown deterministic discrete-time systems. These successful results lead us to explore extensions of the proposed algorithms to other challenging domains, e.g. the problem of model-free optimal control with state and control constraints, robust control of unknown systems using novel data-driven H_∞ control methods etc. An important open problem of interest is how to combine non-parametric function models (e.g. Gaussian Processes) with the proposed methods for reliable and efficient model-free optimal control of general discrete-time systems.

REFERENCES

Adam, S., Busoniu, L., and Babuska, R. (2012). Experience replay for real-time reinforcement learning control. *IEEE Transactions on Systems, Man and Cybernetics–Part C: Applications and Reviews*, 42(2), 201–212.

Banjac, G. and Lygeros, J. (2019). A data-driven policy iteration scheme based on linear programming. In *Proceedings of the 2019 IEEE Conference on Decision and Control*, 816–821.

Barratt, S. and Boyd, S. (2018). Stochastic control with affine dynamics and extended quadratic costs. In *arXiv:1811.00168.*

Bertsekas, D.P. (2017). Value and policy iterations in optimal control and adaptive dynamic programming. *IEEE Transactions on Neural Networks and Learning Systems*, 28(3), 500–509.

Bertsekas, D.P. and Tsitsiklis, J.N. (1996). *Neuro-Dynamic Programming.* Athena Scientific.

Beuchat, P., Georghiou, A., and Lygeros, J. (2016). Alleviating tuning sensitivity in approximate dynamic programming. In *Proceedings of the 2016 IEEE European Control Conference*, 1616–1622.

Beuchat, P., Georghiou, A., and Lygeros, J. (2020). Performance guarantees for model-based approximate dynamic programming in continuous spaces. *IEEE Transactions on Automatic Control*, 65(1), 143–158.

Bradtke, S.J., Ydstie, B.E., and Barto, A.G. (1994). Adaptive linear quadratic quadratic control using policy iteration. In *Proceedings of the 1994 IEEE American Control Conference*, 3475–3479.

Cogill, R., Rotkowitz, M., Roy, B.V., and Lall, S. (2006). An approximate dynamic programming approach to decentralized control of stochastic systems. In *Control of Uncertain Systems: Modelling, Approximation, and Design*, 243–256.

Esfahani, P.M., Sutter, T., Kuhn, D., and Lygeros, J. (2018). From infinite to finite programs: explicit error bounds with applications to approximate dynamic programming. *SIAM Journal on Optimization*, 28(3), 1968–1998.

Falsone, A. and Prandini, M. (2015). An iterative scheme for the approximate linear programming solution to the optimal control of a markov decision process. In *2015 IEEE European Control Conference.*

Hernandez-Lerma, O. and Lasserre, J.B. (1996). *Discrete-Time Markov Control Processes: Basic Optimality Criteria.* Springer-Verlag.

Heydari, A. (2016). Analyzing policy iteration in optimal control. In *Proceedings of the 2016 IEEE American Control Conference*, 5728–5733.

Kiumarsi, B., Lewis, F.L., Modares, H., Karimpour, A., and Naghibi-Sistani, M.B. (2014). Reinforcement q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics. *Automatica*, 50(4), 1167–1175.

Kiumarsi, B., Vamvoudakis, K.G., Modares, H., and Lewis, F.L. (2018). Optimal and autonomous control using reinforcement learning: a survey. *IEEE Transactions on Neural Networks and Learning Systems*, 29(6), 2042–2062.

Konda, V.R. and Tsitsiklis, J.N. (2010). On actor-critic algorithms. *SIAM Journal on Optimization*, 20(6), 2959–2977.

Lewis, F.L., Vrabie, D., and Vamvoudakis, K.G. (2012a). Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers. *IEEE Control Systems Magazine*, 32(6), 76–105.

Lewis, F.L., Vrabie, D., and Syrmos, V.L. (2012b). *Optimal Control.* John Wiley & Sons, New Jersey.

Li, J., Chai, T., Lewis, F.L., Ding, Z., and Jiang, Y. (2019). Off-policy interleaved q-learning: Optimal control for affine nonlinear discrete-time systems. *IEEE Transactions on Neural Networks and Learning Systems*, 30(5), 1308–1320.

Li, J., Chai, T., Lewis, F.L., Fan, J., Ding, Z., and Ding, J. (2018). Off-policy q-learning: Set-point design for optimizing dual-rate rougher flotation operational processes. *IEEE Transactions on Industrial Electronics*, 65(5), 4092–4102.

Liu, Q., Zhou, X., Zhu, F., Fu, Q., and Fu, Y. (2014). Experience replay for least-squares policy iteration. *IEEE/CAA Journal of Automatica Sinica*, 1(3), 274–281.

Luo, B., Liu, D., Huang, T., and Wang, D. (2016). Model-free optimal tracking control via critic-only q-learning. *IEEE Transactions on Neural Networks and Learning Systems*, 27(10), 2134–2144.

Luo, B., Liu, D., and Wu, H.N. (2018). Adaptive constrained optimal control design for data-based nonlinear discrete-time systems with critic-only structure. *IEEE Transactions on Neural Networks and Learning Systems*, 29(8), 2099–2111.

Luo, B., Liu, D., Wu, H.N., Wang, D., and Lewis, F.L. (2017). Policy gradient adaptive dynamic programming for data-based optimal control. *IEEE Transactions on Cybernetics*, 47(10), 3341–3354.

Petretti, A. and Prandini, M. (2014). An approximate linear programming solution to the probabilistic invariance problem for stochastic hybrid systems. In *2014 IEEE Conference on Decision and Control.*

Powell, W.P. (2011). *Approximate Dynamic Programming: Solving the Curses of Dimensionality.* John Wiley & Sons, New Jersey.

Tao, G. (2003). *Adaptive Control Design and Analysis.* John Wiley & Sons, New Jersey.

Wang, Y., O'Donoghue, B., and Boyd, S. (2015). Approximate dynamic programming via iterated bellman inequalities. *International Journal of Robust and Nonlinear Control*, 25(10), 1472–1496.

Watkins, C. (1989). *Learning from delayed rewards.* Ph.D. thesis, Cambridge University, U.K.

Wei, Q., Liu, D., and Shi, G. (2015). A novel dual iterative q-learning method for optimal battery management in smart residential environments. *IEEE Transactions on Industrial Electronics*, 62(4), 2509–2518.

Wei, Q., Song, R., Li, B., and Lin, X. (2018). *Self-Learning Optimal Control of Nonlinear Systems: Adaptive Dynamic Programming Approach.* Science Press Beijing and Springer Nature Singapore.

Zha, D., Lai, K.H., Zhou, K., and Hu, X. (2019). Experience replay optimization. In *Proceedings of the 2019 International Joint Conference on Artificial Intelligence*, 4243–4249.