Learning Optimal Control Policies for Stochastic Systems
with a Relaxed Bellman Operator

Andrea Martinelli and John Lygeros

Abstract—We introduce a relaxed version of the Bellman operator for \(q\)-functions and prove that it is still a monotone contraction mapping with a unique fixed point. In the spirit of the linear programming approach to approximate dynamic programming, we exploit the new operator to build a simplified linear program (LP) for \(q\)-functions. In the case of discrete-time stochastic linear systems with infinite state and action spaces, the solution of the LP preserves the minimizers of the optimal \(q\)-function. Therefore, even though the solution of the LP does not coincide with the optimal \(q\)-function, the policy we retrieve is the optimal one. The LP has fewer decision variables than existing programs, and we show how it can be employed together with reinforcement learning approaches when the dynamics is unknown.

I. INTRODUCTION

The term optimal control came into use in the 1950s to describe the problem of designing a controller to minimize a measure of a dynamical system’s behavior over time [1]. The problem formulation is widely applicable and it arises in many disciplines, from robotics to bioengineering to finance [2], [3], [4], to name a few. In the same years, R. Bellman developed a method, based on the principle of optimality [5], that uses the concept of value function to define a functional equation – the Bellman equation [6]. The class of methods for solving optimal control problems by solving this equation came to be known as dynamic programming (DP). Most of these methods are based on three fundamental approaches: value iteration (VI), policy iteration (PI) and linear programming (LP) [5]. The mathematical foundations of these approaches lie in the monotonicity and contractivity properties shown by a functional Bellman operator, implicitly defined in the Bellman equation [7]. DP methods suffer from the same monotonicity and contractivity properties as the standard operator. We show that, in the case of stochastic linear systems in continuous spaces, the unique fixed point of the new operator does not coincide with the optimal \(q\)-function but, nevertheless, preserves the minimizers with respect to the control variable, hence recovers the optimal policy. Moreover, our LP has half the decision variables of existing programs for stochastic systems, thanks to the relaxation induced by the modified operator. Finally, we demonstrate how to build and solve a sampled version of the LP with RL techniques.

A. Functional Analysis

Let \(X\) be a finite dimensional vector space. We introduce a weight function \(r : X \rightarrow \mathbb{R}\) such that \(r(x) > 0 \ \forall x \in X\), and denote with \(S(X)\) the vector space of all real-valued measurable functions \(v : X \rightarrow \mathbb{R}\) that have a finite weighted sup-norm [7, §2.1]

\[
\|v(x)\|_{\infty,r} = \sup_{x \in X} \frac{|v(x)|}{r(x)} < \infty. \tag{1}
\]

The following definitions and theorem can be found in [23, §1] and [24, Def. 5.1-1 and Thm. 5.1-2], respectively.

Definition 1 (Monotonicity): A map \(T : S(X) \rightarrow S(X)\) is monotone if

\[
\langle Tv_1 - Tv_2, v_1 - v_2 \rangle \geq 0 \ \forall v_1, v_2 \in S(X).
\]

Next, consider a metric \(d\) on the space \(S(X)\), making \((S(X), d)\) a metric space.
Definition 2 (Contraction): A map $\mathcal{T} : \mathcal{S}(\mathbf{X}) \to \mathcal{S}(\mathbf{X})$ is a contraction with respect to the metric $d$ if there exists a constant $\gamma \in [0,1)$ such that
\[
d(Tv_1, Tv_2) \leq \gamma d(v_1, v_2) \quad \forall v_1, v_2 \in \mathcal{S}(\mathbf{X}).
\]

Theorem 1 (Banach's Theorem): Let $(\mathcal{S}(\mathbf{X}), d)$ be a complete metric space with a contraction $\mathcal{T} : \mathcal{S}(\mathbf{X}) \to \mathcal{S}(\mathbf{X})$. Then, $\mathcal{T}$ has a unique fixed point.

B. Stochastic Optimal Control

Consider a discrete-time stochastic dynamical system
\[
x_{k+1} = f(x_k, u_k, \xi_k),
\] with (possibly infinite) state and action spaces $x_k \in \mathbf{X} \subseteq \mathbb{R}^{n_x}$ and $u_k \in \mathbf{U} \subseteq \mathbb{R}^{n_u}$. Here, $\xi_k \in \Xi \subseteq \mathbb{R}^{n_\xi}$ denotes the realizations of independent identically distributed random variables with zero mean and covariance matrix $\Sigma = \mathbb{E}[\xi_k \xi_k^\top]$, and $f : \mathbf{K} \times \Xi \to \mathbf{X}$, with $\mathbf{K} = \mathbf{X} \times \mathbf{U}$, is the map encoding the dynamics. We consider stationary feedback policies, given by functions $\pi : \mathbf{X} \to \mathbf{U}$; for more general classes of policies, see [12]. A nonnegative cost is associated to each state-action pair through the stage cost function $l : \mathbf{K} \to \mathbb{R}_+$. We introduce a discount factor $\gamma \in (0,1)$ and consider the infinite-horizon cost associated to policy $\pi$
\[
v_\pi(x) = \mathbb{E}_\xi \left[ \sum_{k=0}^{\infty} \gamma^k l(x_k, \pi(x_k)) \mid x_0 = x \right].
\]

The function $v : \mathbf{X} \to \mathbb{R}_+$ is the value function. The objective of the optimal control problem is to find an optimal policy $\pi^*$ such that $v_{\pi^*}(x) = v^*(x) = \inf_\pi v_\pi(x)$.

II. CONTRACTION MAPPINGS AND LINEAR PROGRAMMING

The optimal policy $\pi^*$ is generally difficult to compute since, amongst other issues, it involves the minimization of an infinite sum of costs. However, the value function associated to $\pi$ can be recursively defined as [6]
\[
v_\pi(x) = l(x, \pi(x)) + \gamma \mathbb{E}_\xi [v_\pi(f(x, \pi(x), \xi))],
\]
for all $x \in \mathbf{X}$. From now on, whenever possible we will denote $f(x, \pi(x), \xi)$ as $x_{\pi}^\pi$. Equation (4), of course, also holds for an optimal policy
\[
v^*(x) = l(x, \pi^*(x)) + \gamma \mathbb{E}_\xi [v^*(x_{\pi}^\pi)]
= \inf_{u \in \mathbf{U}} \left\{ l(x, u) + \gamma \mathbb{E}_\xi [v^*(x_{\pi}^u)] \right\}
= T v^*(x) \quad \forall x \in \mathbf{X}.
\]

The operator $T$ is the Bellman operator, it maps from $\mathcal{S}(\mathbf{X})$ to itself and can be shown to possess the two fundamental properties of monotonicity (Definition 1) and $\gamma$-contractivity with respect to the sup-norm (Definition 2) [7], [25]. Thanks to Banach’s Theorem we are then guaranteed that $T$ has a unique fixed point $\bar{v}$ given by
\[
\bar{v} = T \bar{v} = v^* = \lim_{n \to \infty} T^n v \quad \forall v \in \mathcal{S}(\mathbf{X}).
\]

Therefore, we know that $v^*$ is unique and can be obtained by iteratively applying $T$ starting from any function $v \in \mathcal{S}(\mathbf{X})$, a process known as value iteration. By exploiting both the monotonicity and contractivity property, we observe that
\[
v \leq T v \Rightarrow v \leq \lim_{n \to \infty} T^n v = v^*,
\]
meaning that any $v$ satisfying the Bellman inequality (7) is a pointwise lower bound to $v^*$. It is then natural to look for the greatest $v \in \mathcal{S}(\mathbf{X})$ that satisfies (7):

\[
\sup_{v \in \mathcal{S}(\mathbf{X})} \int_\mathbf{X} v(x) c(dx)
\]
\[
\text{s.t. } v(x) \leq T v(x) \quad \forall x \in \mathbf{X},
\]
where $c$ is a finite measure that assigns positive mass to all open subsets of $\mathbf{X}$. Notice that $T$ is a nonlinear operator, and therefore the optimization problem (8) is not a linear program. However, it is possible to reformulate (8) as an equivalent linear program [13] by dropping the infimum operator

\[
\sup_{v \in \mathcal{S}(\mathbf{X})} \int_\mathbf{X} v(x) c(dx)
\]
\[
\text{s.t. } v(x) \leq T_L v(x, u) \quad \forall (x, u) \in \mathbf{K},
\]
where $T_L v(x, u) = l(x, u) + \gamma \mathbb{E}_\xi [v^*(x^u_{\pi}^*)]$. Problem (9) is in general an infinite dimensional linear program, and it is not solvable due to several sources of intractability, which are collectively known as curse of dimensionality. See, e.g., [19] and [26]. If one is able to obtain $v$, they can in principle compute the corresponding policy by

\[
\pi(x) = \arg \min_{u \in \mathbf{U}} \left\{ l(x, u) + \gamma \mathbb{E}_\xi [v^*(x^u_{\pi}^*)] \right\}.
\]

However, if the dynamics $f$ or the stage cost $l$ are not known, this calculation is also impossible. In this regard, we introduce the $q$-function [17] associated to a policy $\pi$ as

\[
q_{\pi}(x, u) = l(x, u) + \gamma \mathbb{E}_\xi [v_{\pi^*}(x^u_{\pi}^*)] = l(x, u) + \gamma \mathbb{E}_\xi [q_{\pi^*}(x^u_{\pi}^*, \pi(x^u_{\pi}^*))],
\]

for all $(x, u) \in \mathbf{K}$. This can be interpreted as the cost of applying control input $u$ at state $x$, and following policy $\pi$ thereafter. The optimal $q$-function is expressed by

\[
q^*(x, u) = l(x, u) + \gamma \mathbb{E}_\xi \left\{ \inf_{u \in \mathbf{U}} q_{\pi^*}(x^u_{\pi^*}, u) \right\} = F q^*(x, u) \quad \forall (x, u) \in \mathbf{K}.
\]

The link between $v_\pi$ and $q_{\pi}$ is given by
\[
v_\pi(x) = \inf_{u \in \mathbf{U}} q_{\pi^*}(x, u),
\]
and the advantage of the $q$-function reformulation is that the policy extraction does not require knowledge of $f$ and $l$:
\[
\pi(x) = \arg \min_{u \in \mathbf{U}} q(x, u).
\]
Since the operator $F$ shares the same monotonicity and contractivity properties of $T$ [27], we can write again a (nonlinear) exact program for the $q$-function

$$
\sup_{q \in S(K)} \int_{K} q(x, u) c(dx, du) \\
\text{s.t. } q(x, u) \leq Fq(x, u) \quad \forall (x, u) \in K.
$$

(15)

This time it is not straightforward to replace the nonlinear constraints in (15) with linear ones due to the nesting of the $E$ and $\inf$ operators in (12). A linear reformulation of (15) can be obtained, as shown in [18] for finite state and action spaces and in [19] for infinite ones, by introducing additional decision variables

$$
\sup_{v \in S(K)} \int_{K} q(x, u) c(dx, du) \\
\text{s.t. } q(x, u) \leq Tlv(x, u) \quad \forall (x, u) \in K \\
v(x) \leq q(x, u) \quad \forall (x, u) \in K.
$$

(16)

Following [20], in the case of deterministic systems the lack of expectation makes possible to compute $q^*$ by solving the simpler LP

$$
\sup_{q \in S(K)} \int_{K} q(x, u) c(dx, du) \\
\text{s.t. } q(x, u) \leq l(x, u) + \gamma q(f(x, u), w),
$$

for all $(x, u, w) \in H = K \times U$.

LP (17), besides having half the decision variables of (16), also allows one to exploit an iterative algorithm [20, Alg. 1] to approximate $q^*$ even if the dynamics and stage cost are not known. The rationale is to explore the state-space with some control policy and collect associated costs, in a reinforcement learning fashion [1], and build a sampled version of (17). At each iteration step, a greedy policy is extracted from (17) and is used to drive the exploration process, in the spirit of policy iteration [5]. Our objective here is to derive a linear program for stochastic systems but with the same structure of (17), i.e., referred to $q$ only and with the additional degree of freedom $w$. To achieve this, we introduce a new functional operator.

### III. THE RELAXED BELLMAN OPERATOR

Consider the relaxed Bellman operator $\hat{F} : S(K) \rightarrow S(K)$

$$
\hat{F}q(x, u) = l(x, u) + \gamma \inf_{w} \mathbb{E}_{\xi} \left[ q(x^+, w) \right].
$$

(18)

Note that the operator (18) retains the same structure as the standard Bellman operator (12), but the expectation and infimum are exchanged. In the following, we show several fundamental properties of (18).

#### A. Properties of the Relaxed Bellman Operator

The aim of this section is to show that $\hat{F}$ is a monotone contraction mapping with a unique fixed point $\hat{q} \in S(K)$. Moreover, $\hat{q}$ is an upper bound to the fixed point of $F$.

**Proposition 1:** The operator $\hat{F}$ is monotone.

**Proof:** Consider two functions $q_1, q_2 \in S(K)$,

$$
q_1(x, w) \leq q_2(x, w) \quad \forall (x, w)
\Rightarrow
\begin{align*}
q_1(x^+, w) & \leq q_2(x^+, w) \quad \forall (x, u, w, \xi) \\
\mathbb{E}_{\xi} \left[ q_1(x^+, w) \right] & \leq \mathbb{E}_{\xi} \left[ q_2(x^+, w) \right] \quad \forall (x, u, w) \\
\inf_{w} \mathbb{E}_{\xi} \left[ q_1(x^+, w) \right] & \leq \inf_{w} \mathbb{E}_{\xi} \left[ q_2(x^+, w) \right] \quad \forall (x, u) \\
\hat{F}q_1(x, w) & \leq \hat{F}q_2(x, w) \quad \forall (x, w),
\end{align*}
$$

hence

$$
\langle \hat{F}q_1 - \hat{F}q_2, q_1 - q_2 \rangle \geq 0 \quad \forall q_1, q_2 \in S(K),
$$

and the operator is monotone.

**Proposition 2:** The operator $\hat{F}$ is a $\gamma$-contraction with respect to the sup-norm.

**Proof:** Given $q_1, q_2 \in S(K)$, we have that

$$
\hat{F}q_1(x, w) - \hat{F}q_2(x, w) =
\begin{align*}
\gamma \text{sup}_{w} \mathbb{E}_{\xi} \left[ q_1(x^+, w) - q_2(x^+, w) \right] \\
= \gamma \left( \inf_{w} \mathbb{E}_{\xi} \left[ q_1(x^+, w) \right] - \inf_{w} \mathbb{E}_{\xi} \left[ q_2(x^+, w) \right] \right) \\
\leq \gamma \text{sup}_{w} \left( \inf_{w} \mathbb{E}_{\xi} \left[ q_1(x^+, w) \right] - \inf_{w} \mathbb{E}_{\xi} \left[ q_2(x^+, w) \right] \right) \\
= \gamma \sup_{w} \mathbb{E}_{\xi} \left[ q_1(x^+, w) - q_2(x^+, w) \right] \\
\leq \sup_{w} \mathbb{E}_{\xi} \left[ q_1(x^+, w) - q_2(x^+, w) \right] \\
\leq \sup_{x, w} \left( q_1(x, w) - q_2(x, w) \right).
\end{align*}
$$

The latter implies

$$
||\hat{F}q_1 - \hat{F}q_2||_{\infty} \leq \gamma ||q_1 - q_2||_{\infty} \quad \forall q_1, q_2 \in S(K),
$$

hence the operator is $\gamma$-contractive.

**Proposition 3:** The operator $\hat{F}$ has a unique fixed point in $S(K)$.

**Proof:** As the vector space $S(K)$ is complete under the sup-norm [7, B.2], the result follows from Proposition 2 and Theorem 1.

**Proposition 4:** The fixed point of $\hat{F}$ is an upper bound to the fixed point of $F$.

**Proof:** Since

$$
\inf_{w} q(x^+, w) \leq q(x^+, w) \quad \forall (x, u, w, \xi),
\Rightarrow
\mathbb{E}_{\xi} \left[ \inf_{w} q(x^+, w) \right] \leq \mathbb{E}_{\xi} \left[ q(x^+, w) \right] \quad \forall (x, u, w) \\
\mathbb{E}_{\xi} \left[ \inf_{w} q(x^+, w) \right] \leq \inf_{w} \mathbb{E}_{\xi} \left[ q(x^+, w) \right] \quad \forall (x, u) \\
\hat{F}q(x, u) \leq \hat{F}q(x, u) \quad \forall q \in S(K),
$$

(19)

which implies that $q^* \leq \hat{q}$.

#### B. The Relaxed Linear Program

Consider the (nonlinear) program

$$
\sup_{q \in S(K)} \int_{K} q(x, u) c(dx, du) \\
\text{s.t. } q(x, u) \leq \hat{F}q(x, u) \quad \forall (x, u) \in K.
$$

(20)

**Proposition 5:** If $q$ is feasible for (15), then is feasible for (20). Moreover, the unique optimal solution to (20) is the fixed point of $\hat{F}$. 


Proof: According to inequality (19), if $q$ is feasible for (15) then $q \leq \mathcal{F}q \leq \hat{\mathcal{F}}q$ $\forall (x, u) \in K$. The second statement follows from Propositions 1-3.

On the same line of the linearizations in (8)-(9) and (15)-(16), one can replace the nonlinear constraints in (20) with linear ones and obtain the relaxed linear program (RLP)

$$\sup_{q \in \mathcal{S}(K)} \int q(x, u)c(dx, du)$$

s.t. $q(x, u) \leq \hat{\mathcal{F}}Lq(x, u, w) \forall (x, u, w) \in H,$

where $\hat{\mathcal{F}}Lq(x, u, w) = l(x, u) + \gamma E_q[q(x^+ u, w)].$

Theorem 2: If $(q, v)$ is feasible for (16), then $q$ is feasible for (21). Moreover, the unique optimal solution to the RLP (21) is the fixed point of $\hat{\mathcal{F}}$.

Proof: If $(q, v)$ is a feasible pair for (16), then $q \leq l + \gamma E_q v \leq l + \gamma E_q = \hat{\mathcal{F}}Lq$ $\forall (x, u, w) \in H$. Furthermore, thanks to Proposition 5, we know that $\hat{q}$ is the unique optimal solution to (20). As the RLP (21) is a relaxation of (20), $\hat{q}$ is feasible for (21). On the other hand, any feasible solution $q'$ to (21) satisfies $q \leq \hat{\mathcal{F}}Lq$ for all $(x, u, w) \in H$ and, in particular, for the $w$ that minimizes $E_q[q(x^+ u, w)]$. That is, $q'$ satisfies $q \leq \hat{q}$ and therefore it is a lower bound to $q$. As a consequence, $\hat{q}$ is the unique optimal solution to (21).

Note that, according to Propositions 5 and 2, the programs (20)-(21) are relaxations of (15)-(16), respectively. Hence, the name of relaxed Bellman operator for $\mathcal{F}$. Indeed, according to Proposition 4, the optimal solution to the RLP is an upper bound to $q^*$. The RLP can be considered as the stochastic counterpart to (17). In contrast to (16), the RLP requires $q$ only and involves half the decision variables. We are interested in the relation between the optimisers of (16) and (21) and the corresponding optimal policies.

IV. FIXED POINT ANALYSIS FOR LINEAR DYNAMICAL SYSTEMS

A special case of the infinite-horizon optimal control problem arises when the dynamical map is linear

$$f(x, u, \xi) = Ax + Bu + \xi,$$

with $X = \mathbb{R}^n_x$, $U = \mathbb{R}^n_u$, $\Xi = \mathbb{R}^n_c$, $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_u \times n_x}$, and the cost function is quadratic

$$l(x, u) = \begin{bmatrix} x^\top \\ u^\top \end{bmatrix} \begin{bmatrix} L_{xx} & L_{xz} \\ L_{zu}^\top & L_{uu} \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} \geq 0, \quad L_{uu} > 0. \quad (23)$$

Assumption 1: The pair $(\gamma \frac{1}{6} A, \gamma \frac{1}{2} B)$ is stabilizable.

The optimal $q$-function for such a problem is [28]

$$q^*(x, u) = \begin{bmatrix} x^\top \\ u^\top \end{bmatrix} \begin{bmatrix} q_{xx}^* \\ q_{zu}^* \\ q_{uu}^* \end{bmatrix} + \frac{\gamma \text{Tr}(P\Sigma)}{1 - \gamma}, \quad (24)$$

with $q_{xx}^* = L_{xx} + \gamma A^\top PA$, $q_{zu}^* = L_{zu} + \gamma A^\top PB$, and $q_{uu}^* = L_{uu} + \gamma B^\top PB$.

The matrix $P \in \mathbb{R}^{n_x \times n_x}$ is the solution of the discrete-time algebraic Riccati equation (DARE)

$$P = Q^* / q_{uu}^*.$$

and $Q^* / q_{uu}^* = q_{xx}^* - q_{zu}^* q_{uu}^* - q_{uu}^*$ indicates the Schur complement of block $q_{uu}^*$ of matrix $Q^*$. Under Assumption 1, $P$ can be shown to be positive definite and unique [29, Cond. 6.1.32]. The optimal policy can then be found by extracting the minimizers with respect to $u$ of (24), resulting in

$$\pi^*(x) = -q_{uu}^{-1} q_{zu}^* x.$$  

Theorem 3: In the linear quadratic case (22)-(23) under Assumption 1, the unique fixed point of the operator $\hat{\mathcal{F}}$ is

$$\hat{q}(x, u) = q^*(x, u) + \frac{\gamma \text{Tr}(q_{xx}^* q_{uu}^* - q_{zu}^* x) + 1}{1 - \gamma}, \quad (27)$$

and the greedy policy $\hat{\pi}$ associated with $\hat{q}$ coincides with the optimal policy $\pi^*$ in (26).

Proof: Consider a positive definite quadratic $q$-function

$$q(x, u) = \begin{bmatrix} x^\top \\ u^\top \end{bmatrix} \begin{bmatrix} q_{xx} & q_{zu} \\ q_{zu}^\top & q_{uu} \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} + e.$$ We want to characterize the fixed point $\hat{q}$ of

$$q(x, u) = \hat{\mathcal{F}}q(x, u).$$

Note that $q(x, u) \in \mathcal{S}(K)$, hence the solution to (28) is unique according to Proposition 3. Moreover, $\hat{\mathcal{F}}q(x, u) = l(x, u) + \gamma \inf_w (#),$ (29)

where

$$# = E_q \left[ \begin{bmatrix} Ax + Bu + \xi \end{bmatrix}^\top Q \begin{bmatrix} Ax + Bu + \xi \end{bmatrix} + e \right]$$

$$= \begin{bmatrix} Ax + Bu \end{bmatrix}^\top Q \begin{bmatrix} Ax + Bu \end{bmatrix} + \text{Tr}(q_{xx} \Sigma) + e.$$

Since $Q$ is positive definite by assumption, we have that $(#)$ is minimized by

$$w = -q_{uu}^{-1} q_{zu} (Ax + Bu).$$

If we substitute (30) into $(#)$, we can write the fixed point equation (28) as

$$\begin{bmatrix} x^\top \\ u^\top \end{bmatrix} Q \begin{bmatrix} x \\ u \end{bmatrix} + e = \begin{bmatrix} x^\top \\ u^\top \end{bmatrix} L \begin{bmatrix} x \\ u \end{bmatrix} +$$

$$+ \gamma \begin{bmatrix} x^\top \\ u^\top \end{bmatrix} \begin{bmatrix} A^\top & B^\top \end{bmatrix} \begin{bmatrix} Q & q_{uu} \end{bmatrix} \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} + \gamma(\text{Tr}(q_{xx} \Sigma) + e).$$

Since (28) has to hold for all $(x, u) \in K$, we impose

$$\begin{cases} Q = L + \gamma \begin{bmatrix} A^\top & B^\top \end{bmatrix} \begin{bmatrix} Q & q_{uu} \end{bmatrix} \begin{bmatrix} A & B \end{bmatrix} \\ e = \gamma(\text{Tr}(q_{xx} \Sigma) + e). \end{cases}$$

(31a) (31b)
Notice that \( Q^* \) defined in (24) is a solution of (31a). In fact, we can decompose it as
\[
Q^* = L + \gamma \left[ A^T PA + A^T PB \right]
\]
\[
= L + \gamma \left[ A^T B^T \right] P \left[ A + B \right]
\]
where we exploited equivalence (25) in the last step. Moreover, equation (31b) is satisfied by
\[
e = \gamma \text{Tr}(q_x \Sigma)/(1 - \gamma).
\]
Therefore, the unique solution of (28) is
\[
\hat{q}(x, u) = \left[ \begin{array}{c} x \\ u \end{array} \right]^T Q^* \left[ \begin{array}{c} x \\ u \end{array} \right] + \frac{\gamma \text{Tr}(q_x \Sigma)}{1 - \gamma}
\]
\[
= q^*(x, u) + \frac{\gamma \text{Tr}(q_x q_u^{-1} q_x^T \Sigma)}{1 - \gamma}.
\]
That is, the fixed points of \( F \) and \( \hat{F} \), in the linear quadratic (LQ) case, only differ by a constant term and have the same minimisers with respect to the control variable. As a consequence, the policy associated with \( \hat{q} \) is
\[
\hat{\pi}(x) = \arg \min_{\pi \in \mathcal{U}} \hat{q}(x, u) = -q_u^{-1} q_x^T x = \pi^*(x),
\]
which is the optimal policy of the LQ problem.

Note that the solution to the RLP (21) preserves the shape of the optimal \( q \)-function, in the sense that both \( \hat{q} \) and \( q^* \) attain the same minimizers with respect to \( u \). The function \( \hat{q} \) is shifted with respect to \( q^* \) by a constant positive offset (27), which agrees with Proposition 4. The constant offset depends on the discount factor and the covariance matrix, which means we can retrieve the optimal policy \( \pi^* \) independently of the magnitude of the noise covariance.

A major breach between our method and the classical LP approach is that we do not try to recover the optimal \( q \)-function but, instead, we are interested in preserving its minimizers only. This opens the doors to the potential employment of different operators in the theory of the LP approach to ADP.

V. LEARNING EXAMPLE

Consider the problem of learning an optimal control policy for the following system
\[
x_{k+1} = \left[ \begin{array}{ccc} 1 & 0.1 \\ 0.5 & -0.5 \end{array} \right] x_k + \left[ \begin{array}{c} 1 \\ 0.5 \end{array} \right] u_k + \xi_k,
\]
when its dynamics is not known. We have that \( x_k \in \mathbb{R}^2 \), \( u_k \in \mathbb{R} \), and \( \xi_k \in \mathbb{R}^2 \) is the realization of two independent Gaussian processes with zero mean and covariance matrix \( \Sigma = \text{diag}(0.1, 0.1) \). The discount factor is \( \gamma = 0.95 \) and the stage cost is \( l(x, u) = x^T \text{diag}(1, 1) x + 0.1 u^2 \).

The objective is to solve the RLP (21) and find the optimal control policy. We stress again that all the LPs introduced in this paper are not directly solvable, in general, due to the curse of dimensionality (see Section II). First, \( q \) is an optimization variable in the infinite dimensional space \( S(K) \). As suggested in [11] and [13], a first approximation can be to restrict \( q \) in the span of a finite family of basis functions. Since \( (\hat{\gamma} A, \hat{\gamma} B) \) form a stabilizable pair, we know that \( q^* \) lies in the following subspace of quadratics
\[
\hat{S}(K) = \left\{ \left[ \begin{array}{c} x \\ u \end{array} \right]^T Q \left[ \begin{array}{c} x \\ u \end{array} \right] + e, \quad Q \in \mathbb{R}^{3 \times 3}, e \in \mathbb{R} \right\} \subseteq S(K).
\]
In particular, as pointed out in [19], we can rewrite the objective of (21) as
\[
\int_K q(x, u) c(dx, du) = \mathbb{E}_c q = \text{Tr}(QC) + e, \quad (33)
\]
where \( C = \text{diag}(1, 1, 0.1) \) is the covariance of the measure \( c \). As highlighted in [13], the choice of \( C \) is not relevant when \( \hat{S}(K) \subseteq S(K) \).

A second source of intractability is that the RLP (21) has an infinite number of constraints. A possible approach is then to sample only a finite subset of the constraints, as argued in [30]. In a reinforcement learning fashion, we run a sequence of state-space explorations (roll-outs), starting from random initial conditions \( x_0 \sim \mathcal{N}(0, \sqrt{5}) \). For each collected data tuple \( (x, \pi(x), x^T, w, l(x, u)) \) we construct one of the constraints in (21). After roll-out \( i \), we solve a sampled version of (21) with objective (33) and we extract the corresponding greedy policy \( \pi_i \). At the next iteration step, we can use \( \pi_i \) to drive the exploration process, in a policy iteration spirit. As shown in [20] and [21] for deterministic systems, we can also exploit the additional degree of freedom \( w \) (e.g. by setting \( w = \pi_i \)) to build new constraints from those we computed previously, without the need to run new roll-outs every time.

Notice that the constraints we sample depend on the realization of the stochastic process \( \xi_k \), while the constraints we need to implement has to hold with expectation (see (21)). Therefore, we re-initialize the dynamics at the same state-action pairs many times (\( \sim 750 \)), as suggested in [22], and we compute the average constraint.

Fig. 1 displays the result of 10 identical experiments, initialized with random stabilizing gains, conducted on system (32) with the reinforcement learning method described above. After less than 10 policy updates and for all 10 experiments, the learned policy \( \hat{\pi} \) converges to \( \pi^*(x) = [k_1^* \ k_2^*] x \approx [0.93 \ -0.14] x \). On the right-hand side of Fig. 1 we show how the performance measure \( \mathbb{E}_c \hat{q} \) does not tend to \( \mathbb{E} q^* \), as theoretically anticipated in (27), but it is shifted by a positive offset \( \hat{q} - q^* \approx 2.41 \).

VI. CONCLUSIONS

We introduced a relaxed Bellman operator \( \hat{F} \) for \( q \)-functions, and we proved that it retains the key properties of monotonicity and contractivity with respect the the sup-norm. We characterized its fixed point \( \hat{q} \) in the LQ case with infinite dimensional spaces, and we showed that \( \hat{q} \) preserves the minimizers of the optimal function \( q^* \). Then, we illustrated how to exploit \( \hat{F} \) to build an LP that retrieves
LP provides significant simplification with respect to already fixed point of dynamical map and stage cost, and characterize again the to optimal control problems, and many promising research existing LPs, since it involves fewer decision variables. We look for functions that preserve the minimizers of \( q \) but we are not approximating the optimal \( q^* \). A major difference with the classical ADP methods is that it would be appropriate to derive a different method that does not need to re-initialize the dynamics at the same state-action pairs to build the constraints for the LP. In fact, the re-initialization might not always be practically possible for stochastic systems. In addition, probabilistic error bounds could be provided to link to amount of data collected with the performance measure.

**ACKNOWLEDGEMENTS**

Thanks to my colleagues and friends P. Beuchat and G. Banjac for the fruitful discussions on ADP.

**REFERENCES**

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. The MIT Press, 2017.

[2] J. Bobrow, S. Dubowsky, and J. Gibson, “Time-optimal control of robotic manipulators along specified paths,” *The International Journal of Robotics Research*, vol. 4, no. 3, pp. 3–17, 1985.

[3] A. D. Kuo, “An optimal control model for analyzing human postural balance,” *IEEE Transactions on Biomedical Engineering*, vol. 42, no. 1, pp. 87–101, 1995.

[4] D. Bertsimas and A. W. Lo, “Optimal control of execution costs,” *Journal of Financial Markets*, vol. 1, no. 1, pp. 1–50, 1998.

[5] D. P. Bertsekas, *Dynamic Programming and Optimal Control, Vol. I*, 3rd ed. Athena Scientific, 2005.

[6] R. Bellman, “On the theory of dynamic programming,” *Proceedings of the National Academy of Sciences*, vol. 38, no. 8, pp. 716–719, 1952.

[7] D. P. Bertsekas, *Abstract Dynamic Programming*. Athena Scientific, 2013.

[8] ——, *Dynamic Programming and Optimal Control, Vol. II*, 3rd ed. Athena Scientific, 2007.

[9] W. B. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. John Wiley & Sons, 2011.

[10] A. S. Manne, “Linear programming and sequential decisions,” *Management Science*, vol. 6, no. 3, pp. 259–267, 1960.

[11] P. J. Schweitzer and A. Seidmann, “Generalized polynomial approximations in markovian decision processes,” *Journal of Mathematical Analysis and Applications*, vol. 110, no. 2, pp. 568–582, 1985.

[12] O. Hernandez-Lerma and J. B. Lasserre, *Discrete-Time Markov Control Processes: Basic Optimality Criteria*. Springer-Verlag NY, 1996.

[13] D. P. de Farias and B. Van Roy, “The linear programming approach to approximate dynamic programming,” *Operations Research*, vol. 51, no. 6, pp. 850–865, 2003.

[14] P. Mohajerin Esfahani, T. Sutter, D. Kuhn, and J. Lygeros, “From infinite to finite programs: Explicit error bounds with applications to approximate dynamic programming,” *SIAM Journal on Optimization*, vol. 28, no. 3, pp. 1968–1998, 2018.

[15] R. S. Sutton, “Learning to predict by the methods of temporal differences,” *Machine Learning*, vol. 3, pp. 9–44, 1988.

[16] L. Bouni, T. de Bruin, D. Toli, J. Kober, and I. Palunko, “Re-inforcement learning for control: Performance, stability, and deep approximators,” *Annual Reviews in Control*, vol. 46, pp. 8–28, 2018.

[17] C. J. C. H. Watkins and P. Dayan, “Q-learning,” *Machine Learning*, vol. 8, no. 3, pp. 279–292, May 1992.

[18] R. Cogill, M. Rotkowitz, B. Van Roy, and S. Lall, “An approximate dynamic programming approach to decentralized control of stochastic systems,” in *Control of Uncertain Systems: Modelling, Approximation, and Design*. Springer-Verlag Berlin Heidelberg, 2006, pp. 243–256.

[19] P. N. Beuchat, A. Georghiou, and J. Lygeros, “Performance guarantees for model-based approximate dynamic programming in continuous spaces,” *IEEE Transactions on Automatic Control*, vol. 65, no. 1, pp. 143–158, 2020.

[20] G. Banjac and J. Lygeros, “A data-driven policy iteration scheme based on linear programming,” in *58th IEEE Conference on Decision and Control (CDC)*, 2019, in press.

[21] A. Tanzanakis and J. Lygeros, “Data-driven control of unknown systems: A linear programming approach,” 2020, arXiv:2003.00779.

[22] T. Sutter, A. Kamoutsi, P. Mohajerin Esfahani, and J. Lygeros, “Data-driven approximate dynamic programming: A linear programming approach,” in *56th IEEE Conference on Decision and Control (CDC)*, 2017, pp. 5174–5179.

[23] R. T. Rockafellar, “Monotone operators and the proximal point algorithm,” *SIAM Journal on Control and Optimization*, vol. 14, no. 5, pp. 877–898, 1976.

[24] E. Kreyszig, *Introductory functional analysis with applications*. John Wiley & Sons, 1978.

[25] E. V. Denardo, “Contraction mappings in the theory underlying dynamic programming,” *SIAM Review*, vol. 9, no. 2, pp. 165–177, 1967.

[26] Y. Wang, B. O’Donoghue, and S. Boyd, “Approximate dynamic programming via iterated Bellman inequalities,” *International Journal of Robust and Nonlinear Control*, vol. 25, no. 10, pp. 1472–1496, 2015.

[27] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*, 1st ed. Athena Scientific, 1996.

[28] S. J. Bradke, B. E. Ydstie, and A. G. Barto, “Adaptive linear quadratic control using policy iteration,” in *American Control Conference (ACC)*, 1994, pp. 3475–3479.

[29] M. H. A. Davis and R. B. Vinter, *Stochastic Modelling and Control*. Chapman and Hall, 1985.

[30] D. P. de Farias and B. Van Roy, “On constraint sampling in the linear programming approach to approximate dynamic programming,” *Mathematics of Operations Research*, vol. 29, no. 3, pp. 462–478, 2004.

**Fig. 1.** Feedback gains (left) and performance measure (right) results for 10 identical reinforcement learning experiments on system (32). In each iteration, 200 roll-outs of length 1 are performed.