Performance guarantees for model-based Approximate Dynamic Programming in continuous spaces

Journal Article

Author(s):

Beuchat, Paul N. ; Georghiou, Angelos; Lygeros, John

Publication date:

2020-01

Permanent link:

<https://doi.org/10.3929/ethz-b-000388076>

Rights / license:

In Copyright - Non-Commercial Use Permitted

Originally published in:

IEEE Transactions on Automatic Control 65(1), <https://doi.org/10.1109/TAC.2019.2906423>

Funding acknowledgement:

787845 - Optimal control at large (EC)

Performance Guarantees for Model-Based Approximate Dynamic Programming in Continuous Spaces

Paul N. Beuchat¹, Angelos Georghiou², and John Lygeros¹, *Fellow, IEEE*

Abstract—We study both the value function and Q-function formulation of the Linear Programming approach to Approximate Dynamic Programming. The approach is model-based and optimizes over a restricted function space to approximate the value function or Q-function. Working in the discrete time, continuous space setting, we provide guarantees for the fitting error and online performance of the policy. In particular, the *online performance guarantee* is obtained by analyzing an iterated version of the greedy policy, and the *fitting error guarantee* by analyzing an iterated version of the Bellman inequality. These guarantees complement the existing bounds that appear in the literature. The Q-function formulation offers benefits, for example, in decentralized controller design, however it can lead to computationally demanding optimization problems. To alleviate this drawback, we provide a condition that simplifies the formulation, resulting in improved computational times.

I. INTRODUCTION

In 1952, Bellman proposed Dynamic Programming (DP) as a solution method for discrete time stochastic optimal control problems [1]. The solution of the Bellman equation is the optimal cost-to-go function, also called the value function, which characterizes the performance of the optimal control policy. For continuous state and action spaces, the value function takes values in an infinite dimensional function space and the policy involves computation of a multi-variate expectation. The continuous state, input, and disturbance spaces could be discretized to leverage the extensive literature on solving the Bellman equation for finite spaces [2], [3], however this is computationally infeasible even for small problems.

As such, an extensive body of literature has proposed and studied various approximation methods for continuous space problems, including a range of model-free methods that learn the value function from interactions with an un-modelled system, [4], [5], [6], [7]. In this paper we focus on the model-based method named the Linear Programming (LP) approach to Approximate Dynamic Programming (ADP) [8] that takes advantage of model knowledge for the system dynamics, cost function, and exogenous disturbance as part of the algorithm for approximating the value function.

This research was partially funded by the European Commission under the project Local4Global, and by the European Research Council under the project OCAL, grant number 787845.

¹ Automatic Control Laboratory at ETH Zürich, Switzerland, {beuchatp, jlygeros}@ethz.ch

² Desautels Faculty of Management, McGill University, Montreal, Canada, angelos.georghiou@mcgill.ca

The LP approach to ADP has been formulated for value functions [9] and Q-functions [10], though much of the model-based LP approach literature focuses on value function approximation. The Q-function has the property that the optimal control policy can be expressed without involving any of the terms that describe the model. This property was exploited in [10], [11] in a model-based setting. In particular, the work of [11] provides numerical evidence that the Q-function approximation can provide better online performance compared to the value function approximation, while the work of [10] uses the Q-function approximation for decentralized control design in finite spaces. A wide range of model-free, data-driven ADP methods also use the Q-function and exploit the fact that the resulting optimal policy does not directly involve any of the terms that describe the model. There are many success stories from these model-free methods, for example [12], [13], [14]. In this paper we provide some discussion to compare model-based and model-free approaches to ADP.

Motivated by the empirical success of the model-based LP approach, [16], [17], a key challenge is to provide theoretical guarantees on the quality of the approximation and the online performance. In [9], the authors presented a variant of the LP approach with theoretical guarantees for finite space problems. They provided three guarantees for the value function formulation: (i) a bound on the *online performance* of the control policy, (ii) a bound on how close the approximate value function is to the optimal in an *infinity norm* sense, and (iii) a bound on how close the approximate value function is to the optimal using a *Lyapunov-based* analysis. A number of works use [9] as a basis for deriving additional performance guarantees. An online performance bound for Q-functions was developed in [10] for the finite space setting. An infinity norm bound for value function approximation was provided in [15] for the continuous space setting by considering an iterated version of the Bellman inequality. An iterated version of the infinity norm bound for Q-functions was given in continuous spaces by [11]. A Lyapunov-based bound was presented in [18] for finite space by analyzing a smoothed version of the LP approach. As many practical control problems involve continuous state, action, and disturbance spaces, it is valuable to derive online performance and Lyapunov-based bounds for the continuous space setting. In this paper we address this gap in the literature for value functions and Q-functions.

The first contribution of this paper derives novel theoretical guarantees for the value function and Q-function approximation using the model-based LP approach in continuous spaces,

TABLE I: Road map to performance guarantees for the Linear Programming approach to Approximate Dynamic Programming. The bold entries represent contributions of this paper.

Spaces	Bound Type	Non-iterated		Iterated	
		Value functions	Q-functions	Value functions	Q-functions
Finite	Online performance	[9, Theorem 1]	[10, Theorem 1]	—	—
	Infinity norm	[9, Theorem 2]	—	—	—
	Lyapunov-based	[9, Theorem 3]	—	—	—
Continuous	Online performance	Theorem 4.2	Theorem 4.1	Theorem 4.2	Theorem 4.1
	Infinity norm	[15, §4.2]	[11, Theorem 4.1]	[15, §4.2]	[11, Theorem 4.1]
	Lyapunov-based	Theorem 4.7	Theorem 4.8	Theorem 4.7	Theorem 4.8

Note: Entries marked with “—” are bounds that do not exist in the literature for the finite space setting.

while the second contribution improves the scalability of the formulation. In particular, the contributions of the paper are:

- We prove a continuous space online performance bound by analyzing an iterated version of the greedy policy. When using the non-iterated greedy policy, our bounds form the counterpart to the bounds derived in [9, Theorem 1] and [10, Theorem 1] for finite spaces.
- We prove a continuous space Lyapunov-based bound by analyzing the iterated Bellman inequality. When using the non-iterated Bellman inequality, our bounds forms the counterpart to the bound derived in [9, Theorem 3] for finite spaces. Additionally, our bounds contain [15, §4.2] and [11, Theorem 4.1] as a special case with a Lyapunov function that is constant for all states and inputs.
- Approximating the Q-function using the LP formulation can be computationally demanding. We provide a condition that substantially decreases the optimization problem size for the Q-function formulation, making the method suitable for practical applications.

The existing results and contributions of this paper are summarized in Table I for the performance guarantees. In support of the contributions, we provide numerical results to demonstrate the bounds, the performance of the iterated policy, and the potential of Q-functions for continuous space distributed control applications. Section II presents the Dynamic Programming formulation. Section III introduces the approximation methods and the iterated policy. Section IV provides the theoretical guarantees for both the value function and Q-function formulations and contrasts with theoretical results from the model-free literature. Section V provides conditions under which the Q-function formulation can be simplified. Section VI uses numerical examples to demonstrate the theory. *Notation:* $\mathbb{R}_+$ ($\mathbb{R}_{++}$) is the space of non-negative (positive) scalars; $\mathbb{S}_n$ is the space of symmetric matrices of size n ; $\mathbb{N}$ is the space of positive integers; I_n is the $n \times n$ identity matrix; given $f: \mathcal{X} \rightarrow \mathbb{R}$, the infinity norm is $\|f\|_\infty = \sup_{x \in \mathcal{X}} |f(x)|$, and the weighted 1-norm is $\|f\|_{1,c} = \int_{\mathcal{X}} |f(x)|c(x)dx$.

II. DYNAMIC PROGRAMMING (DP) FORMULATION

A. Problem Formulation and Assumptions

We consider infinite horizon, stochastic optimal control problems with a discounted cost objective. The state of the system at time t is denoted by $x_t \in \mathcal{X} \subseteq \mathbb{R}^{n_x}$. The state is influenced by control decisions $u_t \in \mathcal{U} \subseteq \mathbb{R}^{n_u}$, and stochastic disturbances $\xi_t \in \Xi \subseteq \mathbb{R}^{n_\xi}$ distributed according to probability measure μ_ξ that is used in all expectations. The state evolves according to $x_{t+1} = g(x_t, u_t, \xi_t)$, where $g: \mathcal{X} \times \mathcal{U} \times \Xi \rightarrow \mathcal{X}$.

At time t , the system incurs the stage cost $\gamma^t l(x_t, u_t)$, where $\gamma \in [0, 1)$ is the discount factor. By Π we denote the set of all feasible policies, i.e., $\{\pi(\cdot) : \pi(x) \in \mathcal{U}, \forall x \in \mathcal{X}\}$, with $\pi(\cdot)$ measurable, see [19, Definition 2.2.3]. We restrict our attention to deterministic stationary policies and define the online performance for a fixed policy and initial state x as,

$$V_\pi(x) = \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t l(x_t, \pi(x_t)) \mid x_0 = x \right] \quad (1)$$

The objective is to find the policy that minimizes (1).

To pose this problem in the DP formulation, we work in the same setting as [19, Section 6.3], specifically under [19, Assumptions 4.2.1(a), 4.2.1(b), 4.2.2]. The assumptions ensure that from the class of time-varying stochastic policies, the minimum is attained by a stationary deterministic policy, see [19, Theorem 4.2.3]. Additionally, under the assumptions it can be seen that an initial state distribution ν and an admissible policy π define a Markov chain. Let $P_\nu^\pi[\cdot]$ denote the probability distribution of the state at time t , given that the initial states are distributed according to ν that is concentrated on $\mathcal{X}$ and the system evolves autonomously under the fixed policy π . Finally, $\mathcal{F}(\mathcal{X} \times \mathcal{U})$ and $\mathcal{F}(\mathcal{X})$ are defined as the vector spaces of bounded, real-valued, Borel-measurable functions on $\mathcal{X} \times \mathcal{U}$ and $\mathcal{X}$ respectively, where [19, Definition 6.3.2, 6.3.4] provides the definitions of boundedness.

B. Bellman Equation and Operator

We now re-cast the stochastic optimal control problem in the dynamic programming formulation. The value function $V^*: \mathcal{X} \rightarrow \mathbb{R}$ represents the optimal cost-to-go from any state of the system if the optimal control policy is played, and is the solution of the Bellman equation [1],

$$V^*(x) = \inf_{u \in \mathcal{U}} \underbrace{l(x, u) + \gamma \mathbb{E} [V^*(g(x, u, \xi))]}_{(\mathcal{T}V^*)(x)}, \quad \forall x \in \mathcal{X}. \quad (2)$$

$\mathcal{T}$ is known as the Bellman operator, and the $\mathcal{T}_u$ operator is used to define an auxiliary function $Q^*: \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}$ that represents the cost of making decision u now and then playing optimally. The Bellman equation in terms of Q^* is thus,

$$Q^*(x, u) = l(x, u) + \gamma \mathbb{E} \left[\underbrace{\inf_{v \in \mathcal{U}} Q^*(g(x, u, \xi), v)}_{(FQ^*)(x, u)} \right], \quad (3)$$

for all $x \in \mathcal{X}$ and all $u \in \mathcal{U}$. The F -operator is the equivalent of $\mathcal{T}$ for the so-called Q-functions. The Q-function is an example of a *post-decision value function* [20, §4.6].

The optimal policy can be defined using V^* or Q^* by,

$$\pi^*(x) = \arg \min_{u \in \mathcal{U}} l(x, u) + \gamma \mathbb{E} [V^*(g(x, u, \xi))], \quad (4a)$$

$$= \arg \min_{u \in \mathcal{U}} Q^*(x, u). \quad (4b)$$

Note that evaluating (4a) requires use of the dynamics, stage cost, and expectation with respect to ξ , whereas (4b) involves only Q^* . The existence of a V^* , Q^* , and π^* that are Borel-measurable and attain the infimum is ensured by [19, Assumptions 4.2.1(a), 4.2.1(b), 4.2.2].

C. LP Reformulation for Q -functions

Inspired by the LP reformulation of (2) [19], we derive an LP whose optimal solution Q^* solves equation (3). Equation (3) can be relaxed to an inequality,

$$Q(x, u) \leq FQ(x, u), \quad \forall x \in \mathcal{X}, u \in \mathcal{U}, \quad (5)$$

called the *F-operator inequality*. One can show that operator F is monotone, and satisfies value iteration convergence [10]. Therefore any Q satisfying (5) will be a point-wise under-estimator of Q^* . Hence a solution of the following program,

$$\begin{aligned} \max_{Q \in \mathcal{F}(\mathcal{X} \times \mathcal{U})} \quad & \int_{\mathcal{X} \times \mathcal{U}} Q(x, u) c(x, u) dx du \\ \text{s.t.} \quad & Q(x, u) \leq FQ(x, u), \quad \forall x \in \mathcal{X}, u \in \mathcal{U}, \end{aligned} \quad (6)$$

coincides with the solution of (3) for c -almost all (c -a.a) $(x, u) \in \mathcal{X} \times \mathcal{U}$, where $c(\cdot, \cdot)$ is a finite measure on $\mathcal{X} \times \mathcal{U}$ that assigns positive mass to all open subsets of $\mathcal{X} \times \mathcal{U}$; see Appendix A for details. The equivalence between (3) and (6) requires that $\mathcal{F}(\mathcal{X} \times \mathcal{U})$ is the function space over which the decision variable Q is optimized, see [19, §6.3]. Intuitively speaking, $\mathcal{F}(\mathcal{X} \times \mathcal{U})$ is rich enough to satisfy $Q \leq FQ$ with equality, point-wise for all $x \in \mathcal{X}$ and all $u \in \mathcal{U}$.

The feasible region of (6) can be increased by using an iterated F -operator inequality. A Q -function satisfying $Q \leq F^M Q$, with $M \in \mathbb{N}$, will be a point-wise under-estimator of Q^* . By F^M we denote M applications of the F operator, and under [19, Assumptions 4.2.1(a), 4.2.1(b), 4.2.2] we have that $F^M Q \xrightarrow{M \rightarrow \infty} Q^*$. The same reasoning as with (6) also establishes that a solution of the following program:

$$\begin{aligned} \max_{Q \in \mathcal{F}(\mathcal{X} \times \mathcal{U})} \quad & \int_{\mathcal{X} \times \mathcal{U}} Q(x, u) c(x, u) dx du \\ \text{s.t.} \quad & Q(x, u) \leq F^M Q(x, u), \quad \forall x \in \mathcal{X}, u \in \mathcal{U}, \end{aligned} \quad (7)$$

coincides with the solution of (3) for c -a.a $(x, u) \in \mathcal{X} \times \mathcal{U}$.

The constraint in (7) is non-linear in Q due to nested minimizations and expectations. A linear reformulation is obtained by introducing additional decision variables and constraints. Following the lines of [10, Theorem 2] and [15, §3.4], problem (7) is equivalent to the infinite dimensional linear program:

$$\begin{aligned} \max_{Q_j, V_j} \quad & \int_{\mathcal{X} \times \mathcal{U}} Q_0(x, u) c(x, u) dx du \\ \text{s.t.} \quad & Q_j \in \mathcal{F}(\mathcal{X} \times \mathcal{U}), V_j \in \mathcal{F}(\mathcal{X}), \quad j=0, \dots, M-1, \\ & Q_j(x, u) \leq \mathcal{T}_u V_j(x, u), \quad j=0, \dots, M-1, \\ & V_j(x) \leq Q_{j+1}(x, u), \quad j=0, \dots, M-2, \\ & V_{M-1}(x) \leq Q_0(x, u), \end{aligned} \quad (8)$$

where the inequality constraints hold for all $x \in \mathcal{X}$ and $u \in \mathcal{U}$. The propositions necessary to show the equivalence between (7) and (8) are given in Appendix B.

The introduction of the iterated F -operator inequality is seemingly unnecessary as it does not change the solution of problem (6), however, it can improve the approximation quality in Section III where the decision variables are restricted to a finite dimensional space. The LP reformulation (8) is necessary for applying the approximation techniques of Section III.

D. Sources of Intractability

Solving (8) for Q^* , and implementing (4b), is in general intractable. The difficulties can be categorized as:

- (D1) $\mathcal{F}(\mathcal{X})$ and $\mathcal{F}(\mathcal{X} \times \mathcal{U})$ are infinite dimensional spaces;
- (D2) Problem (8) has infinite number of constraints;
- (D3) Objective of (8) involves a multidimensional integral;
- (D4) The $\mathcal{T}_u$ -operator involves an infinite dimensional integral over ξ ;
- (D5) Since Q^* can be any element of $\mathcal{F}(\mathcal{X} \times \mathcal{U})$, the policy (4b) may be intractable;

Difficulties (D1-D5) relate to the so-called *curse of dimensionality* [21], and apply also to the iterated value function formulation in the continuous space setting [15].

III. APPROXIMATE DYNAMIC PROGRAMMING (ADP)

A. The Approximate LP

As suggested in [8], we restrict the value functions and Q -functions to take values in the span of a finite family of basis functions $\hat{V}_j^{(i)}: \mathcal{X} \rightarrow \mathbb{R}$ and $\hat{Q}_j^{(i)}: \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}$. We parameterize the restricted function spaces as

$$\begin{aligned} \hat{\mathcal{F}}_j(\mathcal{X}) &= \left\{ \sum_{i=1}^K \alpha_j^{(i)} \hat{V}_j^{(i)}(x) \mid \alpha_j^{(i)} \in \mathbb{R}, \right\}, \\ \hat{\mathcal{F}}_j(\mathcal{X} \times \mathcal{U}) &= \left\{ \sum_{i=1}^K \beta_j^{(i)} \hat{Q}_j^{(i)}(x, u) \mid \beta_j^{(i)} \in \mathbb{R}, \right\}, \end{aligned} \quad (9)$$

for $j = 0, \dots, M$. The subscript j is used to highlight that the restricted function space can be different for each of the value functions and Q -functions. If desired, all of the restricted spaces can be taken to be the same.

An approximate solution of (7) is obtained by the program:

$$\begin{aligned} \max_{\hat{Q} \in \hat{\mathcal{F}}_0(\mathcal{X} \times \mathcal{U})} \quad & \int_{\mathcal{X} \times \mathcal{U}} \hat{Q}(x, u) c(x, u) dx du \\ \text{s.t.} \quad & \hat{Q}(x, u) \leq F^M \hat{Q}(x, u), \quad \forall x \in \mathcal{X}, u \in \mathcal{U}, \end{aligned} \quad (10)$$

where the only change from (7) was to replace $\mathcal{F}(\mathcal{X} \times \mathcal{U})$ by $\hat{\mathcal{F}}_0(\mathcal{X} \times \mathcal{U})$. The optimization variables are now the $\beta_0^{(i)}$'s in the definition of $\hat{\mathcal{F}}_0(\mathcal{X} \times \mathcal{U})$. To apply existing methods for the LP approach to ADP, we make the constraint in (10) linear by applying Proposition B.1 and B.2 with all the additional value functions and Q -functions restricted to $\hat{\mathcal{F}}_j(\mathcal{X})$ and $\hat{\mathcal{F}}_j(\mathcal{X} \times \mathcal{U})$ respectively. The additional decision variables and constraints introduced by this linear reformulation are a drawback that we address in Section V.

In general, a solution of (10), denoted $\hat{Q}^*$, will not solve the Bellman equation (3). The following lemma, which follows

TABLE II: Examples of overcoming (D2-D5)

Ref.	Problem instance studied:	Class of basis functions	Overcome (D2) by:
[15], [22], [23]	Linear-quadratic problems	Quadratic	S-procedure
[24], [15], [25]	Polynomial problems	Polynomial	Sum-of-squares
[26], [27], [28]	Finite, linear-quadratic, non-linear	Finite, quadratic, non-linear	Sampling
[29], [30]	Stochastic reachability	Radial basis functions	Sampling
[31] [32]	Perimeter surveillance	Piecewise-constant	Exact Reformulation

from [9, Lemma 1], provides the intuition that $\hat{Q}^*$ is the closest under-estimator of Q^* weighted by $c(\cdot, \cdot)$.

Lemma 3.1: $\hat{Q}$ is an optimal solution of (10) if and only if it is an optimal solution of the following program

$$\begin{aligned} \min_{\hat{Q} \in \hat{\mathcal{F}}_0(\mathcal{X} \times \mathcal{U})} \quad & \left\| Q^* - \hat{Q} \right\|_{1, c(x, u)} \\ \text{s.t.} \quad & \text{same as (10)}. \end{aligned} \quad (11)$$

A natural choice for the online policy is to replace Q^* in equation (4) with the solution of (10),

$$\hat{\pi}(x) = \arg \min_{u \in \mathcal{U}} \hat{Q}^*(x, u), \quad (12)$$

often referred to as the *greedy policy*. A good approximation of the optimal Q-function is one for which the online performance of (12) is near optimal. Although Lemma 3.1 shows that $\hat{Q}^*$ is the closest approximate Q-function for a given set of basis functions, it reveals nothing about the sub-optimality of policy (12). In Section IV we show that the online performance of (12) can be bounded by how well $\hat{Q}^*$ approximates Q^* .

Problem (10) overcomes difficulty (D1) as $\hat{\mathcal{F}}_j(\mathcal{X})$ and $\hat{\mathcal{F}}_j(\mathcal{X} \times \mathcal{U})$ are parameterized by a finite dimensional decision variable. There are a number of choices of $\hat{\mathcal{F}}_j(\mathcal{X})$ and $\hat{\mathcal{F}}_j(\mathcal{X} \times \mathcal{U})$ that address (D2-D5). The possible choices depend on the class of the stage cost and dynamics, the description of $\mathcal{X}$ and $\mathcal{U}$, and the distribution of the exogenous disturbance. Table II summarizes examples found in the literature, where the applicability, approximation quality, and computational burden depends on the problem data and design choices made when a practitioner implements the chosen algorithm.

In (8) the specific choice of $c(\cdot, \cdot)$ does not affect the optimal solution. This is no longer the case in (10) where the choice of $c(\cdot, \cdot)$ plays a central role in determining the quality of $\hat{Q}^*$. Lemma 3.1 suggests that one can influence the approximation quality by an appropriate choice of $c(\cdot, \cdot)$, which is commonly referred to as the *relevance weighting*. To partly alleviate the dependency on the choice of relevance weighting, [11] suggests solving (10) for multiple choices of $c(\cdot, \cdot)$, and using the point-wise maximum from the family of approximations in the greedy policy. They argue that improved online performance can be achieved with this approach. Note that if the restricted function space is chosen such that $Q^* \in \hat{\mathcal{F}}_0(\mathcal{X} \times \mathcal{U})$, then the optimal solution of (10) is Q^* as long as $c(\cdot, \cdot)$ assigns positive mass to all open subsets of $\mathcal{X} \times \mathcal{U}$.

For completeness and comparison, we state without derivation that approximate iterated LP for the value function formulation of ADP, as introduced in [15],

$$\begin{aligned} \max_{\hat{V} \in \hat{\mathcal{F}}_0(\mathcal{X})} \quad & \int_{\mathcal{X}} \hat{V}(x) c(x) dx \\ \text{s.t.} \quad & \hat{V}(x) \leq \mathcal{T}^M \hat{V}(x), \quad \forall x \in \mathcal{X}. \end{aligned} \quad (13)$$

Weighting $c(\cdot)$ here is the counterpart of the relevance weighting in the objective of (10) and similarly it plays a central role in determining the quality of $\hat{V}^*$. The constraint is called the iterated Bellman inequality and the LP reformulation of the non-linear operator $\mathcal{T}^M$ is given in [15, §3.4].

We note that, under the assumptions of Section II-A, programs (10) and (13) are always feasible. Specifically, under [19, Assumption 4.2.1(a)] that the stage cost is non-negative, the choice $\alpha_j^{(i)} = \beta_j^{(i)} = 0$ for all i, j , is feasible for both the iterated F -operator and Bellman inequality constraints.

B. Iterated Greedy Policy

The following policy attempts to bridge the gap between finite horizon and two stage problems. Given $D \in \mathbb{N} \cup \{0\}$ and an approximate Q-function, we define the iterated greedy policy by

$$\hat{\pi}(x) = \arg \min_{u \in \mathcal{U}} F^D \hat{Q}(x, u). \quad (14)$$

The policy may improve upon (12) for any $D \geq 1$, where we use the convention that $F^0 Q = Q$ and hence (12) and (14) coincide when $D=0$. However, computing this iterated policy is complicated by the nested expectations and minimizations arising from the $F^D \hat{Q}$ term. Using similar arguments, an iterated greedy policy using an approximate value function, $\hat{V}$, would be,

$$\hat{\pi}(x) = \arg \min_{u \in \mathcal{U}} l(x, u) + \gamma \mathbb{E} \left[\left(\mathcal{T}^D \hat{V} \right) (g(x, u, \xi)) \right], \quad (15)$$

which also involves nested expectations and minimizations, and coincides with the usual greedy policy for $D=0$.

Writing out the iterations of the F or $\mathcal{T}$ operator, it can be seen that the iterated greedy policy is exactly the generic form of a D -stage stochastic programming problem [33, section 3.1]. Popular approximate solution methods for such stochastic programs are Model Predictive Control (MPC) [34], [35] and Affine Decision Rules (ADR) [36], [37]. In [38] the authors analyze and provide algorithms for computing an MPC policy parametric in the current state x , referred to as *explicit MPC*. For example, when the dynamics are linear and the stage cost quadratic, as is the case in the Section VI examples, the explicit MPC policy is shown to be piecewise linear [38, §6.3]. Solving (14) or (15) with an MPC approach would be equivalent to a finite horizon MPC formulation, with a time horizon of D steps, and $\hat{Q}$ or $\hat{V}$ as the terminal cost. Further details on the connection between ADP and MPC policies are given in [39].

In Section IV, we give a bound on the sub-optimality of the online performance achieved by (14) or (15). This indicates that a tighter performance bound can be achieved through the iterated greedy policy. In Section VI we use a numerical example to demonstrate the potential of this interpretation.

IV. PERFORMANCE BOUNDS FOR ADP

In this section, we present performance guarantees for the continuous space setting. The *online performance* bounds in Section IV-A and *Lyapunov-based* bounds in Section IV-C are novel for the continuous space setting and represent a contribution of this paper. To assist the reader, Table I summarizes the proposed bounds and those found in the literature, [9], [10], [11], and [15].

Note that the bounds in Section IV-B and IV-C require that the restricted functions spaces (9) are all the same, which we denote as $\hat{\mathcal{F}}(\mathcal{X})$ and $\hat{\mathcal{F}}(\mathcal{X} \times \mathcal{U})$ throughout this section.

A. Online Performance Bound

We present first a bound on the online performance of playing the iterated greedy policy (14) or (15). These bounds only require the approximate value function or Q-function to be a point-wise under-estimator of V^* or Q^* respectively. To this end, we introduce two measures: the *expected state-action frequency*, μ defined on $\mathcal{X} \times \mathcal{U}$, and its marginal on the state space, $\tilde{\mu}$ defined on $\mathcal{X}$, called the *expected state frequency*. For any Borel sets $\Gamma \in \mathcal{B}(\mathcal{X} \times \mathcal{U})$ and $B \in \mathcal{B}(\mathcal{X})$ the measures are defined as:

$$\begin{aligned} \mu(\Gamma) &:= \sum_{t=0}^{\infty} \gamma^t P_{\nu}^{\pi} [(x_t, \pi(x_t)) \in \Gamma], \\ \tilde{\mu}(B) &:= \mu(B \times \mathcal{U}) = \sum_{t=0}^{\infty} \gamma^t P_{\nu}^{\pi} [x_t \in B], \end{aligned} \quad (16)$$

see [19, 6.3.6] for further details. One can show that $(1-\gamma)\tilde{\mu}$ is a probability measure. From Section II-A it is clear that V_{π} in (1) is a point-wise over-estimator of V^* . Given a function $Q: \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}$, define the following: $Q|_{\pi}(x) := Q(x, \pi(x))$.

Theorem 4.1: Let $\hat{Q}: \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}$ be such that $\hat{Q}(x, u) \leq Q^*(x, u)$ for all $x \in \mathcal{X}$ and all $u \in \mathcal{U}$, and let $\hat{\pi}: \mathcal{X} \rightarrow \mathcal{U}$ be a D -iterated policy defined in (14). Then the sub-optimality of the online performance is bounded as,

$$\|V_{\hat{\pi}} - V^*\|_{1,\nu} \leq \frac{1}{1-\gamma} \left\| Q^*|_{\hat{\pi}} - \left(F^D \hat{Q} \right)|_{\hat{\pi}} \right\|_{1,(1-\gamma)\tilde{\mu}}.$$

The proof is given in Appendix C.

Theorem 4.2: Let $\hat{V}: \mathcal{X} \rightarrow \mathbb{R}$ be such that $\hat{V}(x) \leq V^*(x)$ for all $x \in \mathcal{X}$, and let $\hat{\pi}: \mathcal{X} \rightarrow \mathcal{U}$ be a D -iterated policy defined in (15). Then the sub-optimality of the online performance is bounded as,

$$\|V_{\hat{\pi}} - V^*\|_{1,\nu} \leq \frac{1}{1-\gamma} \left\| V^* - \left(\mathcal{T}^D \hat{V} \right) \right\|_{1,(1-\gamma)\tilde{\mu}}.$$

The proof is a minor adaptation of the proof of Theorem 4.1. Notice that for $D=0$, Theorems 4.1 and 4.2 are reminiscent of the finite space versions, [10, Theorem 1] and [9, Theorem 1] respectively. The proofs, however, require a different analysis due to the consideration of continuous spaces. Fig. 1 visualizes the quantities involved.

The following insights apply to Theorem 4.1 and 4.2:

- They provide the reassurance for continuous space problems that when policy (14) or (15) uses an under-estimator, the sub-optimality of the online performance is bounded by how closely $\hat{Q}$ or $\hat{V}$ fits Q^* or V^* respectively.

Fig. 1: The upper plot shows the online performance of policy $\hat{\pi}$ is a point-wise over-estimator of V^* , and that by assumption of Theorem 4.2 $\hat{V}$ is a point-wise under-estimator. The lower plot highlights that the 1-norm weightings in Theorems 4.1 and 4.2, ν and $\tilde{\mu}$, can differ significantly.

- They motivate the potential benefit of considering a D -iterated policy based on an under-estimator. Although F and $\mathcal{T}$ are not contractive with respect to the weighted 1-norm, it is expected that the right hand side gets smaller as D increases, and hence the online sub-optimality is more tightly bounded.

B. Infinity-norm Bound

We present now a result that bounds the fitting of $\hat{Q}^*$ or $\hat{V}^*$ relative to Q^* or V^* , by how close Q^* or V^* is to the span of the basis functions. These bound were reported in [11] and [15] and are included here for completeness.

Theorem 4.3: Let Q^* be the solution of (3) and $\hat{Q}^*$ be the solution of (10) for a given choice $\hat{\mathcal{F}}(\mathcal{X} \times \mathcal{U})$ and $c(\cdot, \cdot)$ then,

$$\|Q^* - \hat{Q}^*\|_{1,c(x,u)} \leq \frac{2}{1-\gamma^M} \inf_{\hat{Q} \in \hat{\mathcal{F}}(\mathcal{X} \times \mathcal{U})} \|Q^* - \hat{Q}\|_{\infty}. \quad (17)$$

The proof was first reported in [11, Theorem 4.1]. It is included in Appendix D in the interest of completeness.

Theorem 4.4: Let V^* be the solution of (2) and $\hat{V}^*$ be the solution of (13) for a given choice $\hat{\mathcal{F}}(\mathcal{X})$ and $c(\cdot)$ then,

$$\|V^* - \hat{V}^*\|_{1,c(x)} \leq \frac{2}{1-\gamma^M} \inf_{\hat{V} \in \hat{\mathcal{F}}(\mathcal{X})} \|V^* - \hat{V}\|_{\infty}.$$

The proof is given in [15, §4.3].

Comparing the left-hand-side in Theorem 4.4 to the right-hand-side in Theorem 4.2, the choice $c(\cdot) = (1-\gamma)\tilde{\mu}(\cdot)$ means that the online performance is bounded by Theorem 4.4. For Theorems 4.3 and 4.1 to be combined in a similar way, the relevance weighting $c(\cdot, \cdot)$ should satisfy,

$$\left\| Q^*|_{\hat{\pi}} - \left(F^D \hat{Q} \right)|_{\hat{\pi}} \right\|_{1,\tilde{\mu}} = \|Q^* - \hat{Q}^*\|_{1,c(x,u)}.$$

In both cases, choosing c as described is a difficult task since it is a circular requirement: the choice of c affects the solution of the approximate LP, which in turn affects the approximate policy, which affects the expected state frequency $\tilde{\mu}$, which in turn affects the desired relevance weighting c .

The following insights apply to Theorem 4.3 and 4.4:

- As $\|Q^* - \hat{Q}^*\|_{\infty}$ and $\|V^* - \hat{V}^*\|_{\infty}$ may be very large, the bounds may be too conservative for practical use. We investigate this limitation, which affects all similar bounds in literature, through numerical examples in Section VI.

- The right-hand-side of the bounds hold for any choice of the relevance weightings. Thus, the bounds do not provide any intuition for how to choose $c(\cdot, \cdot)$ or $c(\cdot)$.
- A large M tightens the bound via the γ^M term. The upper bound on M is dictated by the size of the approximate LP that can be solved in the time frame available.
- The right-hand-side of the bounds may be infinite in some cases. Consider for example a linear-quadratic problem on unbounded spaces where V^* is known to be quadratic. If $\hat{\mathcal{F}}(\mathcal{X})$ is the space of affine functions, then $\|V^* - \hat{V}\|_\infty$ is infinite for all elements from $\hat{\mathcal{F}}(\mathcal{X})$.

C. Lyapunov-based Bound

Finally, we derive Lyapunov-based bounds that are novel for the continuous space setting. To this end, for functions $V : \mathcal{X} \rightarrow \mathbb{R}$ we define an operator H_V ,

$$(H_V V)(x) = \max_{u \in \mathcal{U}} \mathbb{E}[V(f(x, u, \xi))],$$

and for functions $Q : \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}$ we define an operator H_Q ,

$$(H_Q Q)(x, u) = \max_{v \in \mathcal{U}} \mathbb{E}[Q(f(x, u, \xi), v)].$$

Given that the system is in state x , the function $(H_V V)(x)$ represents the worst case expected value of V at the next state. For Q-functions, given further that action u will be applied, the function $(H_Q Q)(x, u)$ represents the worst case expected value two times steps into the future. It is readily shown that both H_V and H_Q are monotone operators.

Given function V and Q , let,

$$\begin{aligned} \beta_V &= \max_{x \in \mathcal{X}} \gamma (H_V V)(x) / V(x), \\ \beta_Q &= \max_{(x, u) \in (\mathcal{X} \times \mathcal{U})} \gamma (H_Q Q)(x, u) / Q(x, u), \end{aligned}$$

be the maximum ratio of the worse case expected value at a future time step, to the value in the current state(-by-input).

Definition 4.5: A function $V : \mathcal{X} \rightarrow \mathbb{R}_{++}$ is called a *Lyapunov function* if $\beta_V < 1$.

Definition 4.6: A function $Q : \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}_{++}$ is called a *Lyapunov Q-function* if $\beta_Q < 1$.

For any positive function, $V : \mathcal{X} \rightarrow \mathbb{R}_{++}$, let $1/V$ denote the map $x \mapsto 1/V(x)$, and similarly for a strictly positive $Q : \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}_{++}$. Now we can state the bounds.

Theorem 4.7: Let V^* be the solution of (2) and $\hat{V}^*$ be the solution of (13) for a given choice $\hat{\mathcal{F}}(\mathcal{X})$ and $c(\cdot)$. Then, for any Lyapunov function $\hat{V}^+(x) \in \hat{\mathcal{F}}(\mathcal{X})$,

$$\|V^* - \hat{V}^*\|_{1, c(x)} \leq \frac{2 \|\hat{V}^+\|_{1, c(x)}}{1 - \beta_{\hat{V}^+}^M} \inf_{\hat{V} \in \hat{\mathcal{F}}} \|V^* - \hat{V}\|_{\infty, 1/\hat{V}^+}.$$

The proof is given in Appendix E.

Theorem 4.8: Let Q^* be the solution of (3) and $\hat{Q}^*$ be the solution of (10) for a given choice $\hat{\mathcal{F}}(\mathcal{X} \times \mathcal{U})$ and $c(\cdot, \cdot)$. Then, for any Lyapunov Q-function $\hat{Q}^+(x, u) \in \hat{\mathcal{F}}(\mathcal{X} \times \mathcal{U})$:

$$\|Q^* - \hat{Q}^*\|_{1, c(x, u)} \leq \frac{2 \|\hat{Q}^+\|_{1, c(x, u)}}{1 - \beta_{\hat{Q}^+}^M} \inf_{\hat{Q} \in \hat{\mathcal{F}}} \|Q^* - \hat{Q}\|_{\infty, 1/\hat{Q}^+}$$

The proof follows by modifying the proof for Theorem 4.7. When $M=1$, Theorem 4.7 is reminiscent of the finite space version [9, Theorem 3]. The proof requires an adapted analysis due to the consideration of the iterated Bellman inequality.

The following insights apply to Theorem 4.7 and 4.8:

- Theorems 4.3 and 4.4 are a special case of 4.8 and 4.7 because the function that returns a constant value for all $x \in \mathcal{X}$, $u \in \mathcal{U}$ is a Lyapunov function with $\beta_V = \beta_Q = \gamma$.
- As the inverse of the Lyapunov function weights the infinity norm term, the bounds may be tighter than Theorems 4.3 and 4.4. To see this, consider that in regions where V^* or Q^* are large, the Lyapunov function may also be chosen to be large and hence reduce the worst case error in those regions. Section VI-A provides an example where, for larger M , the bound tightening is significant.
- The relevance weighting now appears on the right-hand side of the bound. This indicates that an appropriate choice of relevance weighting is the one which gives the tightest bound. However, finding the combination of a relevance weighting and Lyapunov function that yields the tightest bound is, in general, a difficult problem.

We refer to [9, §5] for some discussion on the choice of Lyapunov functions for finite space problems.

D. Comparison to Temporal Difference and Q-learning

Model-free approaches, such as Temporal Difference (TD) learning [4] and Q-learning [12], aim to optimize the control policy based only on data collected through interactions with the system. Recent results demonstrate many successes and great potential of these methods, see for example [6], [7], [13], [14], [40], [41]. By contrast, model-based approaches assume complete and accurate knowledge of the underlying system model when synthesizing a control policy, and through this provide theoretical analysis and performance guarantees. In particular, the LP approach to ADP ensures that the approximate value functions and Q-functions are point wise under-estimators of V^* and Q^* , thus facilitating the theoretical guarantees presented in Sections IV-A, IV-B, IV-C. In this section we compare aspects of the model-based LP approach to ADP with examples from the TD and Q-learning literature. We refer the reader to [42, Chapter 8] for further discussion on comparing model-based and model-free approaches.

Similar to the LP approach, many variants of TD learning also use a linearly parameterized function space for the approximation architecture, for example, [4], [43], [44]. In [45] the authors provide theoretical guarantees on the approximation quality of the solution from TD learning. In particular, [45, Theorem 1] provides an approximation quality bound that is reminiscent of Theorem 4.4. However, as TD learning is designed for autonomous systems, there is no notion of an online performance guarantee. For a controlled system, the Actor-Critic algorithm in [46] uses TD learning in the critic step, and in the actor step it makes gradient updates in the control policy space. The authors show convergence of their Actor-Critic algorithm to a local minimum with respect to the parametrization of the control policy. However, they do not provide any bound on the sub-optimality of the resulting

policy. By contrast, the model-based LP approach to ADP allows one to compute online performance guarantees such as those offered by Theorems 4.1 and 4.2.

Recent Q-learning methods utilize Neural Networks for the restricted function space [2, Section 6.3.1], [40], and demonstrate many successes, for example playing games [7], [41] and regulating a two-degree-of-freedom helicopter [47]. This suggests that Neural Networks can be an interesting choice of restricted function space for the LP approach to ADP presented in this paper. However, the non-linear nature of Neural Networks will complicate the analysis of the LP approach and makes for an interesting future research direction, potentially providing guarantees for a fixed Neural Network architecture.

V. PARTICULAR Q-FUNCTION FORMULATIONS

In this section, we consider cases for which the Q-function formulation can be simplified. We first present the condition which facilitates this simplification, thus making the formulation computationally efficient. We then provide two problem classes for which the condition is satisfied. In particular, this formulation can be beneficial for the decentralized control designs that we discuss in Section V-C.

A. Condition for equivalence

Applying Propositions B.1 and B.2 to (10), the approximate LP for the Q-function formulation is,

$$\begin{aligned} \max_{\hat{Q}_j, \hat{V}_j} \quad & \int_{\mathcal{X} \times \mathcal{U}} \hat{Q}_0(x, u) c(x, u) dx du \\ \text{s.t.} \quad & \hat{Q}_j \in \hat{\mathcal{F}}_j(\mathcal{X} \times \mathcal{U}), \hat{V}_j \in \hat{\mathcal{F}}_j(\mathcal{X}), \quad j = 0, \dots, M-1, \\ & \hat{Q}_j(x, u) \leq \mathcal{T}_u \hat{V}_j(x, u), \quad j = 0, \dots, M-1, \quad (18a) \\ & \hat{V}_j(x) \leq \hat{Q}_{j+1}(x, u), \quad j = 0, \dots, M-2, \quad (18b) \\ & \hat{V}_{M-1}(x) \leq \hat{Q}_0(x, u), \quad (18c) \end{aligned}$$

where the inequality constraints hold for all $x \in \mathcal{X}$ and all $u \in \mathcal{U}$. Now consider the following formulation with $M-1$ fewer Q-functions and $M-1$ fewer infinite constraints:

$$\begin{aligned} \max_{\hat{Q}_0, \hat{V}_j} \quad & \int_{\mathcal{X} \times \mathcal{U}} \hat{Q}_0(x, u) c(x, u) dx du \\ \text{s.t.} \quad & \hat{Q}_0 \in \hat{\mathcal{F}}_0(\mathcal{X} \times \mathcal{U}), \hat{V}_j \in \hat{\mathcal{F}}_j(\mathcal{X}), \quad j = 0, \dots, M-1, \\ & \hat{Q}_0(x, u) \leq \mathcal{T}_u \hat{V}_0(x, u), \quad (19a) \\ & \hat{V}_{j-1}(x) \leq \mathcal{T}_u \hat{V}_j(x, u), \quad j = 1, \dots, M-1, \quad (19b) \\ & \hat{V}_{M-1}(x) \leq \hat{Q}_0(x, u), \quad (19c) \end{aligned}$$

where the inequality constraints hold for all $x \in \mathcal{X}$ and all $u \in \mathcal{U}$. In Lemma 5.1 below, we provide a condition for when (18) and (19) are equivalent.

Lemma 5.1: If the sets $\hat{\mathcal{F}}_j(\mathcal{X})$ and $\hat{\mathcal{F}}_j(\mathcal{X} \times \mathcal{U})$ are chosen such that for all $\hat{V}_j \in \hat{\mathcal{F}}_j(\mathcal{X})$ there exists a $\hat{Q}_j \in \hat{\mathcal{F}}_j(\mathcal{X} \times \mathcal{U})$ with

$$\hat{Q}_j(x, u) = \mathcal{T}_u \hat{V}_j(x, u), \quad \forall x \in \mathcal{X}, u \in \mathcal{U},$$

for $j = 1, \dots, M-1$, then the approximate LP (18) and (19) have the same optimal value and there is a mapping between feasible and optimal solutions in both problems.

The proof is given in Appendix F.

B. Input constrained, Linear-Quadratic control

In the case of linear dynamics, quadratic cost function, and control actions constrained to lie in a polytopic feasible set, then the value function and Q-function is known to be piece-wise quadratic [38, Theorem 6.7]. Hence, quadratic basis functions defined as,

$$\begin{aligned} \hat{\mathcal{F}}_j(\mathcal{X}) &= \left\{ \hat{V}_j(x) \mid V_j(x) = x^\top P_j x + p_j^\top x + s_j \right. \\ &\quad \left. P_j \in \mathbb{S}^{n_x}, p_j \in \mathbb{R}^{n_x}, s_j \in \mathbb{R} \right\} \\ \hat{\mathcal{F}}_j(\mathcal{X} \times \mathcal{U}) &= \left\{ \hat{Q}_j(x, u) \text{ such that:} \right. \\ &\quad \left. Q_j(x, u) = \begin{bmatrix} x \\ u \end{bmatrix}^\top P_j^Q \begin{bmatrix} x \\ u \end{bmatrix} + p_j^Q \begin{bmatrix} x \\ u \end{bmatrix} + s_j^Q \right. \\ &\quad \left. P_j^Q \in \mathbb{S}^{n_x+n_u}, p_j^Q \in \mathbb{R}^{n_x+n_u}, s_j^Q \in \mathbb{R} \right\} \quad (20) \end{aligned}$$

are reasonable choices, see [15, §6]. The $\alpha_j^{(i)}$'s and $\beta_j^{(i)}$'s from (9) are the coefficients of the monomials. In this setting, for any quadratic value function, the term

$$\mathcal{T}_u \hat{V}(x, u) = l(x, u) + \mathbb{E}[\hat{V}(g(x, u, \xi))],$$

is be quadratic in (x, u) , and requires knowledge of the first and second moments of the exogenous disturbance. As $\hat{\mathcal{F}}_j(\mathcal{X} \times \mathcal{U})$ is taken to be the space of all quadratic functions in $[x^\top, u^\top]^\top$, the condition of Lemma 5.1 is satisfied.

C. Structured Q-functions for decentralized control

Consider a decentralized control problem with N agents. The input for each agent, $u = [u_1^\top, \dots, u_N^\top]^\top$, can only depend on a locally available portion of the state vector, $x_1, \dots, x_N$, i.e., a decentralized policy is of the form,

$$u = \pi_{\text{Decent}}(x) = [\pi_1(x_1)^\top \quad \dots \quad \pi_N(x_N)^\top]^\top.$$

This framework is not readily addressed by traditional DP formulations.

As the greedy policy (12) is a constrained optimization problem, decentralized control is realized if both the objective and constraint set have the required separable structure. For the constraint $u \in \mathcal{U}$ we assume that the set is separable, i.e., $\mathcal{U} = \mathcal{U}_1 \times \dots \times \mathcal{U}_N$. As The Q-function is the objective of (12), it will be separable if the Q-function is a sum of per-agent Q-functions that only depend on u_i and x_i . Let $\mathcal{S}$ denote the set of functions with the separable structure:

$$\mathcal{S} = \left\{ \hat{Q}(\cdot, \cdot) \mid \hat{Q}(x, u) = \sum_{i=1}^N \hat{Q}_i(u_i, x_i) + q(x) \right\} \quad (21)$$

where $q(x)$ can be any function of the full state vector. The term $q(x)$ is allowed because it does not affect the decision made by evaluating the greedy policy. This separable Q-function structure is as suggested in [10].

It is necessary to enforce the structural constraint (21) on $\hat{Q}_0$ in both (18) and (19). The remaining Q-functions and value functions need not have the decentralized structure enforced. Lemma 5.1 allows a different restricted function space for $\hat{Q}_0$, and hence can be applied to the decentralized control formulation.

The value function formulation can also be used to approximate a solution to the decentralized control problem. It requires the assumption that $l(x, u) \in \mathcal{S}$, and the restriction on the approximate value function that $\mathbb{E}[\hat{V}(g(x, u, \xi))] \in \mathcal{S}$.

VI. NUMERICAL RESULTS

In this section we present three numerical examples to highlight various aspects of the theory presented above. The first example numerically evaluates the performance bounds from Section IV, the second assesses the potential of the iterated approximate policy presented in Section III, whereas the third demonstrates using Q-function for a distributed control setting as per Section V-C. The second example also provides empirical evidence that the Q-function formulation can achieve tighter lower bounds.

In all numerical examples we use a Linear Quadratic Regulator (LQR) as a point of comparison. This is a linear state feedback controller synthesized via the Riccati equation for a system with linear dynamics and quadratic stage cost. The code to generate the results is found at [48].

A. Evaluation of Performance Bounds

We use a one dimensional example from [15] with $n_x = n_u = n_\xi = 1$ to highlight that although the iterated value function gives an tighter lower bound of the optimal cost-to-go, it can have both worse online performance, and a worse online performance bound. The dynamics, costs, and constraints are given by,

$$\begin{aligned} x_{t+1} &= x_t - 0.5u_t + \xi_t, & l(x, u) &= x^2 + 0.1u^2, \\ \gamma &= 0.95, & |u| &\leq 1 \end{aligned}$$

with the exogenous disturbance and initial condition distributed as $\xi_t \sim \mathcal{N}(0, 0.1)$ and $x_0 \sim \mathcal{N}(0, \sigma_\nu^2 = 10) = \nu$, respectively. The benefit of using a 1-dimensional example is that the value function and optimal policy (V^* and π^*) can be effectively approximated by using a discretization method and used to directly assess the quality of approximation.

We use the space of univariate quadratics as $\hat{\mathcal{F}}(\mathcal{X})$, without a linear term due to the problem symmetry, and we choose the state-relevance weighting as the initial state distribution, i.e.,

$$\hat{\mathcal{F}}(\mathcal{X}) = \{px^2 + s \mid p, s \in \mathbb{R}\}, \quad c(\cdot) = \nu(\cdot).$$

We compare approximate value functions, solved via the iterated approximate LP with $M = \{1, 10, 200\}$ and their respective approximate policies.

Table IV shows the bound of Theorem 4.2 for this example, and Table III shows the bounds of Theorems 4.4 and 4.7. For completeness, details on the computation of V^* , $\hat{V}$, $V_{\hat{\pi}}$, μ , and the Lyapunov functions are given in Appendix G.

TABLE III: Bounds for example VI-A. The last column is the percentage decrease from the right-hand-side of Theorem 4.4 to the right-hand-side of Theorem 4.7, and β_{V+} is for the Lyapunov function that gives the smallest value for right-hand-side of Theorem 4.7.

M	LHS of Thm. 4.4 & 4.7	RHS of Thm. 4.4	RHS of Thm. 4.7	β_{V+}	% ↓
1	22.2	28158	27831	0.970	1.2
10	16.4	3509	3161	0.972	9.9
200	10.1	1408	541	0.988	61.6

Table III shows that, for this example, the bounds of Theorems 4.4 and 4.7 are conservative, but the Lyapunov-based

TABLE IV: Online performance bound of Theorem 4.2 for example VI-A. Evaluated numerically using 10^8 Monte Carlo simulations per controller.

M	D	$\ V_{\hat{\pi}} - V^*\ _{1, \nu}$	$\leq$	$\frac{1}{1-\gamma} \ V^* - (\mathcal{T}^D \hat{V})\ _{1, (1-\gamma)\hat{\mu}}$
LQR		0.061	$\leq$	73.2
1	0	0.061	$\leq$	113.7
10	0	0.069	$\leq$	138.0
200	0	0.079	$\leq$	168.5
200	1	0.061	$\leq$	158.5
200	2	0.061	$\leq$	150.1
200	3	0.061	$\leq$	142.6
200	4	0.061	$\leq$	135.8
200	5	0.061	$\leq$	129.4

approach tightens the bound for all values of M . Interestingly, the benefit of the Lyapunov-based bound is more pronounced for larger M . To understand the reasoning for this example, see that that a β_{V+} closer to 1 coincides with a Lyapunov function that minimizes the term:

$$\left\| \hat{V}^+ \right\|_{1, c(x)} \min_{\hat{V} \in \hat{\mathcal{F}}} \left\| V^* - \hat{V} \right\|_{\infty, 1/\hat{V}^+}.$$

However, a larger M is required to ensure that the denominator term $1 - \beta_{V+}^M$ does not dominate the bound. For each value of M , there is a sweet spot that gives the tightest bound, see Appendix G. This highlights the benefit of deriving the Lyapunov-based bound using the iterated Bellman formulation.

As indicated by the dependence on M , the right-hand-side of Theorems 4.4 and 4.7 are improved by more than an order of magnitude in going from $M = 1$ to $M = 200$. However, as the bounds are anyway conservative, it is not clear that in general the left-hand-side of Theorem 4.4 and 4.7 should decrease as M increases. For this example, the choice $c(\cdot) = \nu(\cdot)$ means the left-hand-side is the under-estimation error of the optimal cost-to-go, $J^* = \int V^* d\nu$, and numerically agree with [15].

Table IV shows that, for this example, the online performance bounds of Theorem 4.2 is also conservative, and that the iterated policy tightens the bound. The difficulty in choosing the state-relevance weighting is highlighted by the fact that the approximate value function with $M = 200$ gives a better lower-bound of V^* but has worse online performance. For the $M = 200$ approximate LP, it would be possible to choose a $c(x)$ different from ν that yields a value function similar to $\hat{V}^*$ with $M = 1$. Thus there is an inherent discrepancy between choosing a $c(x)$ that maximizes the lower-bound of V^* , useful for assessing sub-optimality, and choosing a $c(x)$ that achieves the best online performance.

The bottom five rows of Table IV show that, as expected, the iterated policy improves both the online performance and the online performance bound. A nice feature of this bound is that it theoretically converges to 0 as D increases. However, for higher dimensional systems the Bellman operator can only be approximated for a low number of iterations. We study this in more detail in the next section.

TABLE V: Results averaged over 20 randomly generated 50-dimensional examples for each γ . For the online performance, the expectation over $x_0 \sim \nu$ is computed using 500 samples, and expectation with respect to ξ is computed from 500 Monte Carlo simulation each of length 2000 time steps. For the lower bounds the expectation over x_0 is computed from the same 500 samples. In order to aggregate results across different systems, the costs and computation times are normalized with respect to the average performance of the MPC controller with horizon $T=10$. The column “Controller computation” relates to the average computation time in milliseconds to compute the control action at each time step, using a single thread on a 3.00Ghz Xeon processor. The ratio to the controller with the highest computation load is shown in the “speed-up” column.

Description		Normalized cost, $\gamma = 0.95$				Normalized cost, $\gamma = 0.99$				Controller computation	
		avg.	σ	min.	max.	avg.	σ	min.	max.	time (ms)	speed-up
Online Performance	LQR	1.1475	0.446	0.400	8.249	1.1805	0.266	0.652	4.439	—	—
	$\hat{Q}^*$ with $M = 1, D = 0$	1.0090	0.325	0.368	3.322	1.0257	0.187	0.602	2.611	0.014	80.6
	$\hat{V}^*$ with $M = 1, D = 0$	1.0085	0.325	0.368	3.322	1.0254	0.186	0.602	2.605	0.014	80.8
	$\hat{V}^*$ with $M = 50, D = 0$	1.0044	0.318	0.374	3.249	1.0110	0.176	0.607	2.515	0.014	81.4
	$\hat{Q}^*$ with $M = 50, D = 0$	1.0043	0.318	0.375	3.248	1.0109	0.176	0.608	2.510	0.014	81.4
	MPC: $T = 5, l_T = \text{lqr}$	1.0018	0.319	0.367	3.239	1.0027	0.174	0.598	2.478	0.225	5.2
	$\hat{Q}^*$ with $M = 1, D = 4$	1.0002	0.317	0.367	3.226	1.0025	0.174	0.598	2.472	0.225	5.2
	$\hat{V}^*$ with $M = 1, D = 4$	1.0002	0.317	0.367	3.226	1.0025	0.174	0.598	2.470	0.225	5.2
	$\hat{V}^*$ with $M = 50, D = 4$	1.0001	0.317	0.367	3.223	1.0008	0.173	0.598	2.459	0.225	5.2
	$\hat{Q}^*$ with $M = 50, D = 4$	1.0001	0.317	0.367	3.223	1.0008	0.173	0.598	2.459	0.225	5.2
	MPC: $T = 10, l_T = \text{lqr}$	1.0000	0.317	0.367	3.222	1.0000	0.172	0.598	2.458	1.175	1.0
Optimal, V^* , and Q^*		not available				not available				—	—
Lower Bound	$\hat{Q}^*$ with $M = 50$	0.924	0.287	0.274	2.511	0.892	0.139	0.510	1.924	—	—
	$\hat{V}^*$ with $M = 50$	0.907	0.275	0.280	2.342	0.872	0.132	0.512	1.826	—	—
	$\hat{Q}^*$ with $M = 1$	0.853	0.246	0.274	2.122	0.864	0.121	0.522	1.673	—	—
	$\hat{V}^*$ with $M = 1$	0.812	0.230	0.268	1.947	0.833	0.113	0.518	1.516	—	—

B. High-dimensional example

To highlight the potential of the iterated greedy policy, proposed in Section III-B, on a system of higher dimension we consider an input constrained Linear Quadratic Regulator (LQR) problem. The system dynamics are $x_{t+1} = Ax_t + B_u u + B_\xi \xi$, with $x_t \in \mathbb{R}^{50}$, $u_t \in \mathbb{R}^6$, $\xi_t \in \mathbb{R}^{50}$, and the matrices A , B_u , B_ξ , of compatible size, describe the linear dynamics. The A and B_u matrices are randomly generated with A scaled to be marginally stable, and the results are averaged over the performance on 20 separate extractions for each $\gamma = \{0.95, 0.99\}$. In all cases B_ξ is an identity matrix, and the exogenous disturbance and initial condition are distributed as $\xi_t \sim \mathcal{N}(0, 0.1I_{50})$ and $x_0 \sim \mathcal{N}(0, 9I_{50}) = \nu$, respectively. The $\mathcal{X}$ space is unconstrained, while the $\mathcal{U}$ space is a hyper-rectangle with the lower and upper bounds chosen to make the constraints relevant for the whole horizon.

Table V presents the online performance results of using quadratic approximate value functions and Q-functions, parameterized as in (20). We solve (19) with $M=1$ and $M=50$ and simulate both the greedy policy (12), and the iterated greedy policy (14) and (15) with $D=4$. As discussed in Section III-B, the iterated greedy policy is approximated with an MPC-reformulation, where $D=4$ corresponds to a prediction horizon of $T=5$. For a comparison controller, we use MPC with prediction horizon $T = \{5, 10\}$ and the Riccati equation solution as the terminal cost function, implemented with the batch approach as detailed in [38, §8.2]. The online performance is computed as $\mathbb{E}_\nu \left[\mathbb{E}_\xi \left[\sum_{t=0}^{2000} \gamma^t l(x_t, u_t) \mid x_0, \hat{\pi}(\cdot) \right] \right]$ using 500 Monte Carlo samples from ν , and the expectation with respect to ξ computed from 500 Monte Carlo simulations each of length 2000 time steps. The table also presents lower-bounds on the value function implied by each

approximation. The lower-bound is computed as $\mathbb{E}_\nu [\hat{V}(x)]$ and $\mathbb{E}_\nu [\min_u \hat{Q}(x, u)]$ respectively, with the expectations computed using the same 500 Monte Carlo samples from ν .

For this example, $\hat{Q}^*$ with $M=50$ gives a tighter lower bound when compared to $\hat{Q}^*$ with $M=1$, the same trend as for the 1 dimensional example of Section VI-A. By contrast, the online performance of the greedy policy using $\hat{Q}^*$ with $M=50$ is improved compared to using $\hat{Q}^*$ with $M=1$. The iterated greedy policy achieves a noticeable improvement in the online performance, compared to the standard greedy policy, both for the value function and Q-function formulation.

The most striking feature of this numerical example is the similarity between the iterated greedy policy and the MPC controller used for comparison. They only differ in the time horizon and choice of terminal cost function. The results in Table V highlight that the $\hat{V}^*$ and $\hat{Q}^*$ encode a sufficient approximation of the cost-to-go function to allow for a shorter horizon to be used; for example, using $D = 4$ for the iterated greedy policy results in comparable performance to an MPC controller with horizon $T=10$ and an LQR based terminal cost, but at a fraction of the computational cost. In all cases, computing the policy involves solving a Quadratic Program with the number of decision variables and constraints proportional to the prediction horizon.

This numerical example also indicates that for a system where the input constraints are active at the end of the MPC prediction horizon, choosing an approximate value function or approximate Q-function for the terminal cost can lead to improved online performance. This comes at the expense of solving the approximate LP (10) or (13), which for larger M is more computationally demanding than solving the Riccati equation. The computation time was 20 seconds for $M=1$ and

Fig. 2: (a) Schematic of coupled oscillator model used to demonstrate using Q-functions for distributed control. The constituent sub-systems are the masses m_i , that respectively make decision u_i based on state measurements x_i and $\dot{x}_i$ in the decentralized setting, and may also have access to the state measurements of neighbouring masses in the distributed setting. (b) Quadratic approximate Q-function structure used for the P^Q matrix from (20), i.e., each square represents the coefficient of an order 2 monomial. When only the dark shaded elements are non-zero the greedy policy is decentralized, and when additionally the dotted elements are non-zero the greedy policy is distributed with nearest neighbour communication. (c) Online performance (solid) and lower bounds (dashed) for the coupled oscillator example versus the communication connections. The centralized optimal (dotted) is 757.4 for this example. The horizontal axis is the number of neighbouring oscillators, in each direction, from which state measurements are available for making control decisions. Thus, 0 represents a decentralized controller, and 1 represents a controller with nearest neighbour communication.

24 minutes for $M=50$ on a 4.0Ghz Intel Core i7 processor, with Appendix H providing details of how the problem was reformulated for a commercial solver. As the approximate LP only needs to be solved once for a particular system, this computation can be performed off-line, and the result offers improvements for the online performance and computation as demonstrated by this example.

C. Coupled Oscillator Example

To demonstrate the application of Q-functions for distributed control we use a string of coupled oscillators, visualized as a spring-mass-dampener system in Fig. 2(a). Each mass is considered as a separate system, and needs to make its control decision based on the measurement of its own state, and possibly that of its nearest neighbours.

The coupled oscillator can be modelled by a linear system readily derived by writing the equations of motion for each mass. The state vector is the position and velocity of each mass, denoted as x_i and $\dot{x}_i$ respectively. Each mass can be controlled by a driving force $u_i \in \mathbb{R}$ applied to the mass. The exogenous driving force is $\xi \in \mathbb{R}$ and the factor $\alpha_i \in \mathbb{R}$ represents an external influence. The spring constant and dampening ratio of the elements connecting mass i to mass j are denoted by k_{ij} and c_{ij} respectively. The fixed wall is represented as $i = 0$.

The online performance of the distributed control policies is compared to the optimal centralized policy. To be able to compute the centralized optimal, we use a quadratic stage and unconstrained state and action spaces. The stage cost for each mass is $l_i(x_i, \dot{x}_i, u_i) = 0.5x_i^2 + \dot{x}_i^2 + 0.2u_i^2$, with a discount factor of $\gamma = 0.99$, and the dynamics is converted to discrete time with a 0.05 second sampling time. Fig. 2(a) shows a system with 3 masses for clarity, but for the numerical results in Fig. 2(c) we simulate a system with 20 masses.

As described in Section V-C, an approximate Q-function can lead to a decentralized greedy policy if given an appropriate structure. The Q^* is quadratic for this coupled oscillator example and due to the dynamic coupling the optimal greedy policy does not have a separable structure. Thus, for the restricted function space $\hat{\mathcal{F}}(\mathcal{X} \times \mathcal{U})$ we use quadratic functions parameterized as in (20) with the P^Q matrix restricted to have the structure shown in Fig. 2(b). The structure is shown for a three mass example and is readily extended for a longer string of masses. The approximate greedy policy is decentralized if the shaded structure is used, and distributed with nearest neighbour communication if additionally the dotted elements are non-zero. Note that $Q^* \notin \hat{\mathcal{F}}(\mathcal{X} \times \mathcal{U})$.

Fig. 2(c) presents the online performance results of using structured approximate Q-functions for decentralized and distributed control of the coupled oscillator system with 20 masses and the parameters randomly drawn from a uniform distribution on the following ranges: $m_i \in [0.5, 1.5]$, $k_{ij} \in [3.0, 4.0]$, $c_{ij} \in [0.01, 0.05]$, $\alpha_i \in [0.04, 0.08]$. The exogenous disturbance and initial condition are assumed to be distributed according to $\xi_t \sim \mathcal{N}(0, 1)$, $x_i \sim \mathcal{N}(0, 0.5)$, and $\dot{x}_i \sim \mathcal{N}(0, 1)$ respectively.

We solve (19) with $M = \{1, 10, 20\}$ and simulate greedy policy (12). The online performance is computed using 5000 Monte Carlo samples from the initial state distribution, and the expectation with respect to ξ is computed from 500 Monte Carlo simulations each of length 2000 time steps. As a datum, the online performance of the centralized LQR controller is 757.4 which lies between the upper and lower bound curves in Fig. 2(c). For each approximate Q-function, the lower-bound is computed from the same initial condition samples. Note that these are all lower bounds on the centralized LQR performance because problem (19) is formulated to approximate the centralized problem.

The results in Fig. 2(c) show that, for this example, the de-

centralized/distributed ADP approach using Q-functions, can produce near centralized optimal performance: within 6.3% in the decentralized case, within 1.4% in the distributed, nearest neighbour communication, case. The online performance is significantly influenced by the choice of M .

VII. CONCLUSIONS

In this paper we derived theoretical performance guarantees for the Linear Programming Approach to Approximate Dynamic Programming in continuous spaces. We analyzed an iterated version of the greedy policy to provide a guarantee that the online performance of the policy is bounded. We provided a Lyapunov-based bound on the approximation quality of a solution using the LP approach with the iterated Bellman inequality. This bound demonstrates a 61% tightening, compared to the bound presented in [15], on the numerical example for which the bounds were evaluated.

We proposed a condition that allows for a more efficient iterated Q-function formulation. A numerical case study on linear-quadratic examples with a 50-dimensional state vector demonstrates the potential for large-scale systems. Using an approximate value function or Q-function as a terminal cost for an MPC type controller achieves for these examples comparable online performance with one fifth of the online computational load. The proposed condition applies also when using Q-functions in a decentralized control framework. The online performance using decentralized Q-functions is within 6.3% of the optimal centralized performance for the coupled oscillator example.

As future work, we aim to adapt the LP approach to ADP to address the challenge of tuning the relevance weighting parameter, and through this reduce the conservativeness of the theoretical guarantees. The numerical results demonstrate potential benefits for application to physical systems, particularly for systems with fast dynamics where the online policy is implemented on an embedded micro-controller with limited computation resources.

APPENDIX A EQUIVALENCE OF Q-FUNCTION LP

The theorem below provides the condition on $c(\cdot, \cdot)$ under which the solution of (3), the Q-function variant of the Bellman equation, is feasible and optimal for the LP (6). Let $K(\cdot|\cdot, \cdot)$ denote the discrete-time transition kernel describing the state evolution under the dynamics and the exogenous and control inputs, i.e., given a Borel set $B \in \mathcal{B}(\mathcal{X})$,

$$K(B | x_t, u_t) = \mathbb{P}[g(x_t, u_t, \xi_t) \in B]$$

represents the probability that state x_{t+1} will be in B given that the system is currently in state x_t and input u_t is played. Furthermore, let $\mathcal{M}(\mathcal{X} \times \mathcal{U})$ and $\mathcal{M}(\mathcal{X})$ be defined as the vector spaces of finite signed measures on $\mathcal{X} \times \mathcal{U}$ and $\mathcal{X}$ respectively, bounded as per [19, Definition 6.3.2, 6.3.4]. We use $\mu(d\cdot)$ throughout as short-hand notation for $\mu(\cdot)d\cdot$.

Given $\rho \in \mathcal{M}(\mathcal{X} \times \mathcal{U})$ we define an operator T as,

$$(T\rho)(B) = \rho(B) - \gamma \int_{\mathcal{X} \times \mathcal{U}} K(B|x, u) \rho(d(x, u)).$$

Thus, the dual LP of (6) is,

$$\begin{aligned} \min_{\mu} \quad & \int_{\mathcal{X} \times \mathcal{U}} l(x, u) \mu(d(x, u)) + \int_{\mathcal{X} \times \mathcal{U}} l(x, u) c(d(x, u)) \\ \text{s.t.} \quad & (T\mu)(B) = \gamma \int_{\mathcal{X} \times \mathcal{U}} K(B|x, u) c(d(x, u)), \forall B \in \mathcal{B}(\mathcal{X}) \\ & \mu \in \mathcal{M}^+(\mathcal{X} \times \mathcal{U}), (\mu + c) \in \mathcal{M}^+(\mathcal{X} \times \mathcal{U}). \end{aligned} \quad (22)$$

where $\mathcal{M}^+(\mathcal{X} \times \mathcal{U})$ is the non-negative variant of $\mathcal{M}(\mathcal{X} \times \mathcal{U})$. Now we state the requirement on the state-by-input relevance weighting for (6) to recover Q^* for c -a.a. $x \in \mathcal{X}$.

Theorem A.1: Under [19, Assumptions 4.2.1(a), 4.2.1(b), 4.2.2], if ν and c satisfy,

$$\gamma \int_{\mathcal{X} \times \mathcal{U}} K(B|x, u) c(d(x, u)) = \nu(B), \forall B \in \mathcal{B}(\mathcal{X}) \quad (23)$$

with $c \in \mathcal{M}^+(\mathcal{X} \times \mathcal{U})$, then, the optimal values of (6) and (22) coincide with,

$$\int_{\mathcal{X} \times \mathcal{U}} Q^*(x, u) c(x, u) dx du.$$

Proof: As the term $\int l dc$ in the objective of (22) is an additive constant with respect to the decision variable μ , we have by [19, Theorem 6.3.7] that the optimal value of (22), denoted $J_{(22)}$, satisfies,

$$\begin{aligned} J_{(22)} &= \int_{\mathcal{X}} V^*(x) \nu(dx) + \int_{\mathcal{X} \times \mathcal{U}} l(x, u) c(d(x, u)) \\ &= \int_{\mathcal{X} \times \mathcal{U}} \left(\gamma \int_{y \in \mathcal{X}} V^*(y) K(dy|x, u) + l(x, u) \right) c(d(x, u)) \\ &= \int_{\mathcal{X} \times \mathcal{U}} \left(l(x, u) + \gamma \mathbb{E}_{\xi} [V^*(g(x, u, \xi))] \right) c(d(x, u)) \\ &= \int_{\mathcal{X} \times \mathcal{U}} \left(Q^*(x, u) \right) c(d(x, u)). \end{aligned}$$

The first equality follows from [19, Theorem 6.3.7]. The second equality substitutes (23) for ν and uses Fubini's theorem to switch the order of integration. The third equality is the definition of integration with respect to the transition kernel, and the final equality is the definition of Q^* .

The strong duality between (22) and (6) follows from [19, Theorem 6.3.8]. They use the sequence of value functions: $v_0(\cdot) = 0$ and for all $n \geq 1$ and $x \in \mathcal{X}$

$$v_n(x) = \min_{u \in \mathcal{U}} \left[l(x, u) + \gamma \int_{y \in \mathcal{X}} v_{n-1}(y) K(dy|x, \pi(x)) \right],$$

see [19, equation 6.3.38], to show that both the primal and dual programs converger to V^* . By defining a Q-function to match each value function for all $n \geq 0$ and $(x, u) \in \mathcal{X} \times \mathcal{U}$

$$q_n(x, u) := l(x, u) + \gamma \int_{y \in \mathcal{X}} v_n(y) K(dy|x, u),$$

it follows that problems (6) and (22) have the same optimal value, and that problem (6) attains the supremum. ■

APPENDIX B

PROPOSITIONS FOR LP REFORMULATION

This appendix states the propositions necessary for the reformulation of the non-linear iterated F -operator inequality constraint as a set of linear constraints.

Proposition B.1: For an arbitrary $Q : \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}$ the following statements are equivalent:

- (i) $Q(x, u) \leq FQ(x, u)$ for all $x \in \mathcal{X}$ and all $u \in \mathcal{U}$;
- (ii) There exists $V \in \mathcal{F}(\mathcal{X})$ such that $Q(x, u) \leq \mathcal{T}_u V(x)$ and $V(x) \leq Q(x, u)$ for all $x \in \mathcal{X}$ and all $u \in \mathcal{U}$.

The proof is given in [10, Theorem 2]. Note, if V is in some subset of $\mathcal{F}(\mathcal{X})$, then the reformulation is only sufficient, i.e., (ii) $\Rightarrow$ (i).

Proposition B.2: For an arbitrary $Q : \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}$ the following are equivalent:

- (i) $Q(x, u) \leq F^M Q(x, u)$ for all $x \in \mathcal{X}$ and all $u \in \mathcal{U}$;
- (ii) There exists $Q_1, \dots, Q_{M-1} \in \mathcal{F}(\mathcal{X} \times \mathcal{U})$ such that:

$$\begin{aligned} Q(x, u) &\leq FQ_1(x, u), \\ Q_{j-1}(x, u) &\leq FQ_j(x, u), \quad j=2, \dots, M-1, \\ Q_{M-1}(x, u) &\leq FQ(x, u), \end{aligned}$$

where the inequalities hold for all $x \in \mathcal{X}$ and all $u \in \mathcal{U}$.

The proof follows from [15, §3.4]. Note, if for any j , Q_j is in some subset of $\mathcal{F}(\mathcal{X} \times \mathcal{U})$, then the reformulation is only sufficient, i.e., (ii) $\Rightarrow$ (i).

APPENDIX C

PROOF OF ONLINE PERFORMANCE BOUND

Given a measure $\rho \in \mathcal{M}(\mathcal{X})$ (see [19, Definition 6.3.4]), a feasible policy $\pi : \mathcal{X} \rightarrow \mathcal{U}$, and a Borel set $B \in \mathcal{B}(\mathcal{X})$, define the operator $T_\pi : \mathcal{M}(\mathcal{X}) \rightarrow \mathcal{M}(\mathcal{X})$ as,

$$(T_\pi \rho)(B) = \rho(B) - \gamma \int_{x \in \mathcal{X}} K(B|x, \pi(x)) \rho(dx).$$

Thus T_π represents the discounted difference in occupancy measure between two time steps of the stochastic process. Given a function $V \in \mathcal{F}(\mathcal{X})$, and the same feasible policy, consider also the operator $T_\pi^* : \mathcal{F}(\mathcal{X}) \rightarrow \mathcal{F}(\mathcal{X})$ defined as,

$$(T_\pi^* V)(x) = V(x) - \gamma \int_{y \in \mathcal{X}} V(y) K(dy|x, \pi(x)). \quad (24)$$

Thus T_π^* represents the expected value of discounted difference between two time steps of the stochastic process. Both operators define a continuous linear map on the corresponding spaces and are adjoints of each other, i.e.,

$$\int_{x \in \mathcal{X}} V(x) (T_\pi \rho)(dx) = \int_{x \in \mathcal{X}} (T_\pi^* V)(x) \rho(dx), \quad (25)$$

see [19, Section 6.3]. The online performance bound for finite space is proven by inverting the transition kernel matrix, see [9, Theorem 1]. The adjoint property of T_π and T_π^* can be seen as a counterpart to inverting the transition kernel.

A required identity is that the online performance can be expressed in terms of the stage cost and the frequency measure defined in Section IV-A. Given a policy, $\pi : \mathcal{X} \rightarrow \mathcal{U}$ and the

expected state frequency with respect to that policy, $\tilde{\mu}$, the online performance is expressed as:

$$\begin{aligned} V_\pi(y) &:= \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t l(x_t, \pi(x_t)) \mid x_0 = y \right] \\ &= \int_{\mathcal{X}} l(x, \pi(x)) \tilde{\mu}(dx) \end{aligned} \quad (26)$$

When the left hand side is integrated over the initial state distribution, ν , then $\tilde{\mu}$ is chosen accordingly.

A final identity relates the initial state distribution to the expected state frequency. Given any $B \in \mathcal{B}(\mathcal{X})$ the following relation holds:

$$\begin{aligned} \nu(B) &= \tilde{\mu}(B) - \gamma \int_{x \in \mathcal{X}} K(B|x, \pi(x)) \tilde{\mu}(dx) \\ &= (T_\pi \tilde{\mu})(B) \end{aligned} \quad (27)$$

This identity stems from [19, eq. (6.3.10)].

We now have all the tools required to prove Theorem 4.1.

Proof of Theorem 4.1:

For all $k \in \mathbb{N}$,

$$\hat{Q}(x, u) \leq F^k \hat{Q}(x, u) \leq Q^*(x, u) \leq Q_{\hat{\pi}}(x, u), \quad (28)$$

for all $x \in \mathcal{X}$ and $u \in \mathcal{U}$, and hence also for all $u = \hat{\pi}(x) \in \mathcal{U}$.

Recalling the notation $Q|_\pi(x) := Q(x, \pi(x))$, we have,

$$\begin{aligned} &\|V_{\hat{\pi}} - V^*\|_{1, \nu} \\ &= \int_{\mathcal{X}} (V_{\hat{\pi}}(x) - V^*(x)) \nu(dx) \\ &\leq \int_{\mathcal{X}} \left(V_{\hat{\pi}}(x) - \left(F^D \hat{Q} \right) \Big|_{\hat{\pi}}(x) \right) \nu(dx) \\ &= \int_{\mathcal{X}} l(x, \hat{\pi}(x)) \tilde{\mu}(dx) - \int_{\mathcal{X}} \left(F^D \hat{Q} \right) \Big|_{\hat{\pi}}(x) (T_\pi \tilde{\mu})(dx) \\ &= \int_{\mathcal{X}} l(x, \hat{\pi}(x)) \tilde{\mu}(dx) - \int_{\mathcal{X}} \left(T_\pi^* \left(F^D \hat{Q} \right) \Big|_{\hat{\pi}} \right)(x) \tilde{\mu}(dx) \\ &= \int_{\mathcal{X}} \left(F^{D+1} \hat{Q} \right) \Big|_{\hat{\pi}}(x) \tilde{\mu}(dx) - \int_{\mathcal{X}} \left(F^D \hat{Q} \right) \Big|_{\hat{\pi}}(x) \tilde{\mu}(dx) \\ &\leq \int_{\mathcal{X}} Q^*|_{\hat{\pi}}(x) \tilde{\mu}(dx) - \int_{\mathcal{X}} \left(F^D \hat{Q} \right) \Big|_{\hat{\pi}}(x) \tilde{\mu}(dx) \\ &= \frac{1}{1-\gamma} \left\| Q^*|_{\hat{\pi}} - \left(F^D \hat{Q} \right) \Big|_{\hat{\pi}} \right\|_{1, (1-\gamma)\tilde{\mu}} \end{aligned}$$

The first equality and first inequality hold by the point-wise ordering of (28). The second equality uses (26) for the first term and (27) for the second term. The third equality uses (25), while the fourth uses (24) to expand the T_π^* operator, and then the definition of the F -operator and the chosen policy to construct the first term. The last inequality and equality follow from the point-wise ordering of (28) and the definition of the 1-norm. The factor $(1-\gamma)$ was introduced so that the scaling in the 1-norm is a probability measure. ■

APPENDIX D

PROOF OF INFINITY-NORM BOUND

The proof of Theorem 4.3 uses the lemmas stated below without proof. These proofs can be found in [49, Appendix

D]. The proof here is an adaptation to Q-functions of the proof for value functions that is given in [15, §4.3], and requires the following auxiliary lemmas.

Lemma D.1: For any $Q : \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}$ and $M \in \mathbb{N}$ iterations,

$$(F^M Q)(x, u) \geq Q(x, u) - (1 + \gamma^M) \|Q^* - Q\|_\infty,$$

for all $x \in \mathcal{X}$ and all $u \in \mathcal{U}$.

Lemma D.2: Let $\hat{Q}(x, u) \in \hat{\mathcal{F}}(\mathcal{X} \times \mathcal{U})$ be an arbitrary element from the basis functions set, and let $\tilde{Q}(x, u)$ be defined as,

$$\tilde{Q}(x, u) = \hat{Q}(x, u) - \underbrace{\frac{1 + \gamma^M}{1 - \gamma^M} \|Q^* - \hat{Q}\|_\infty}_{\text{downwards shift term}}, \quad (29)$$

then $\tilde{Q}(x, u)$ satisfies the iterated F -operator inequality, and if $\hat{\mathcal{F}}(\mathcal{X} \times \mathcal{U})$ allows for affine combinations of the basis functions, then $\tilde{Q}$ is also an element of $\hat{\mathcal{F}}(\mathcal{X} \times \mathcal{U})$.

Proof of Theorem 4.3:

Given any $\tilde{Q} \in \hat{\mathcal{F}}(\mathcal{X} \times \mathcal{U})$, construct $\tilde{Q} \in \hat{\mathcal{F}}(\mathcal{X} \times \mathcal{U})$ following Lemma D.2 to be feasible for the approximate iterated LP. Working from the left hand side of equation (17),

$$\begin{aligned} & \|Q^* - \hat{Q}^*\|_{1, c(x, u)} \\ & \leq \|Q^* - \tilde{Q}\|_{1, c(x, u)} \\ & \leq \|Q^* - \tilde{Q}\|_\infty \\ & \leq \|Q^* - \hat{Q}\|_\infty + \|\hat{Q} - \tilde{Q}\|_\infty \\ & = \|Q^* - \hat{Q}\|_\infty + \frac{1 + \gamma^M}{1 - \gamma^M} \|Q^* - \hat{Q}\|_\infty \\ & = \frac{2}{1 - \gamma^M} \|Q^* - \hat{Q}\|_\infty \end{aligned}$$

where the first inequality holds by Lemma 3.1 because $\tilde{Q}$ is also feasible for (10), the second inequality by assuming without loss of generality that $c(x, u)$ is a probability measure, the third inequality is an application of the triangle inequality, the first equality stems directly from the definition of $\tilde{Q}$, and the final is an algebraic manipulation. As this argumentation holds for any $\tilde{Q} \in \hat{\mathcal{F}}(\mathcal{X} \times \mathcal{U})$, the result follows. ■

APPENDIX E

PROOF OF LYAPUNOV-BASED BOUND

The proof of Theorem 4.7 uses the lemmas stated below without proof. These proofs can be found in [49, Appendix E]. The proof is reminiscent of that for [9, Theorem 3], but requires an adapted analysis for consideration of the iterated Bellman inequality and continuous spaces.

Lemma E.1: For any two functions $V_1, V_2 : \mathcal{X} \rightarrow \mathbb{R}$,

$$|(\mathcal{T}^M V_1)(x) - (\mathcal{T}^M V_2)(x)| \leq \gamma^M (H_V^M(|V_1 - V_2|))(x),$$

for all $x \in \mathcal{X}$, and any $M \in \mathbb{N}$.

Lemma E.2: For any positive function $V^+ : \mathcal{X} \rightarrow \mathbb{R}_{++}$, any function $V : \mathcal{X} \rightarrow \mathbb{R}$, and any integer $M \geq 1$,

$$V(x) - (\mathcal{T}^M V)(x) \leq (V^+(x) + \gamma^M (H_V^M V^+)(x)) \epsilon$$

for all $x \in \mathcal{X}$, where $\epsilon = \|V^* - V\|_{\infty, 1/V^+}$.

Lemma E.3: Given any Lyapunov function V (Definition 4.5), and its respective Lyapunov constant β_V , then,

$$\begin{aligned} & \left(\frac{2}{1 - \beta_V^M} - 1 \right) (V(x) - \gamma^M (H_V^M V)(x)) \\ & \geq (V(x) + \gamma^M (H_V^M V)(x)) \end{aligned}$$

for all $x \in \mathcal{X}$.

Lemma E.4: Let $\hat{V}^+(x)$ be a Lyapunov function (Definition 4.5) and $\hat{V} \in \hat{\mathcal{F}}(\mathcal{X})$ arbitrary, and define $\tilde{V}$ as,

$$\tilde{V}(x) = \hat{V}(x) - \epsilon \left(\frac{2}{1 - \beta_{\hat{V}^+}^M} - 1 \right) \hat{V}^+(x) \quad (30)$$

where $\epsilon = \|V^* - \hat{V}\|_{\infty, 1/\hat{V}^+}$, then $\tilde{V}(x) \leq (\mathcal{T}^M \tilde{V})(x)$ for all $x \in \mathcal{X}$, i.e., it is feasible for the approximate iterated LP. Additionally, if $\hat{V}^+ \in \hat{\mathcal{F}}(\mathcal{X})$ then $\tilde{V}$ is an element of $\hat{\mathcal{F}}(\mathcal{X})$.

Proof of Theorem 4.7:

Given any $\tilde{V}(x) \in \hat{\mathcal{F}}(\mathcal{X})$, construct $\tilde{V} \in \hat{\mathcal{F}}(\mathcal{X})$ following Lemma E.4 to be feasible for the approximate iterated LP. Working from the left hand side of the bound,

$$\begin{aligned} & \|V^* - \hat{V}^*\|_{1, c} \\ & \leq \|V^* - \tilde{V}\|_{1, c} \\ & = \int_{\mathcal{X}} \left(\frac{\hat{V}^+(x)}{\hat{V}^+(x)} \right) |V^*(x) - \tilde{V}(x)| c(dx) \\ & \leq \left(\int_{\mathcal{X}} \hat{V}^+(x) c(dx) \right) \sup_{z \in \mathcal{X}} \frac{|V^*(z) - \tilde{V}(z)|}{\hat{V}^+(z)} \\ & = \left(\|\hat{V}^+\|_{1, c(x)} \right) \|V^* - \tilde{V}\|_{\infty, 1/\hat{V}^+} \\ & \leq \|\hat{V}^+\|_{1, c(x)} \left(\|V^* - \hat{V}\|_{\infty, 1/\hat{V}^+} + \|\hat{V} - \tilde{V}\|_{\infty, 1/\hat{V}^+} \right) \\ & = \|\hat{V}^+\|_{1, c(x)} \left(\frac{2}{1 - \beta_{\hat{V}^+}^M} \right) \|V^* - \hat{V}\|_{\infty, 1/\hat{V}^+} \end{aligned}$$

where the inequalities hold for all $x \in \mathcal{X}$. The first inequality follows from Lemma 3.1 and Lemma E.4. The first equality is the definition of the weighted 1-norm and holds as $\hat{V}^+$ is strictly positive. The second inequality holds because the objective of the supremum is non-negative for all $z \in \mathcal{X}$. The second equality is the definition of the weighted 1-norm and weighted ∞ -norm. The final inequality follows by the triangle inequality. The final equality stems from using (30) by taking the weighted ∞ -norm of $(\hat{V} - \tilde{V})$ and then some simple algebra. As the inequality established holds for any $\hat{V}(x) \in \hat{\mathcal{F}}(\mathcal{X})$, it also holds when the infimum over all $\hat{V}(x) \in \hat{\mathcal{F}}(\mathcal{X})$ is taken on the right-hand-side. ■

APPENDIX F

PROOFS OF EQUIVALENT Q-FUNCTION FORMULATION

Proof of Lemma 5.1:

We shall show that any feasible solution of (18) corresponds to a feasible solution of (19) with the same objective value,

and vice versa. Note that for the proof superscript $(\cdot)'$ indicates a decision variable of problem (19).

Suppose that $\{\hat{Q}_j\}_{j=0}^{M-1}$, $\{\hat{V}_j\}_{j=0}^{M-1}$ is a feasible solution of (18), and take the following decision variables for (19),

$$\hat{Q}'_0 = \hat{Q}_0, \quad \hat{V}'_j = \hat{V}_j, \quad j = 0, \dots, M-1.$$

We now check feasibility for the constraints of (19).

$$\hat{Q}'_0(x, u) = \hat{Q}_0(x, u) \leq \mathcal{T}_u \hat{V}_0(x, u) = \mathcal{T}_u \hat{V}'_0(x, u),$$

for all $x \in \mathcal{X}$ and $u \in \mathcal{U}$, thus (19a) is satisfied. We have that for $j = 1, \dots, M-1$,

$$\hat{V}'_{j-1}(x) = \hat{V}_{j-1}(x) \leq \hat{Q}_j(x, u) \leq \mathcal{T}_u \hat{V}_j(x, u) = \mathcal{T}_u \hat{V}'_j(x, u),$$

for all $x \in \mathcal{X}$ and $u \in \mathcal{U}$, thus (19b) are satisfied. Finally,

$$\hat{V}'_{M-1}(x) = \hat{V}_{M-1}(x) \leq \hat{Q}_0(x, u) = \hat{Q}'_0(x, u),$$

for all $x \in \mathcal{X}$ and $u \in \mathcal{U}$, thus (19c) is also satisfied, and the considered decision variables are feasible for problem (19). As $\hat{Q}'_0 = \hat{Q}_0$, the objective values are equal. This completes the equivalence in one direction.

Suppose that $\hat{Q}'_0$, $\{\hat{V}'_j\}_{j=0}^{M-1}$ is a feasible solution of (19), and take the following decision variables for (18),

$$\begin{aligned} \hat{Q}_0 &= \hat{Q}'_0, \\ \hat{V}_j &= \hat{V}'_j, \quad j = 0, \dots, M-1, \\ \hat{Q}_j &= \mathcal{T}_u \hat{V}'_j, \quad j = 1, \dots, M-1, \end{aligned}$$

where the choices of $\hat{Q}_j$ are valid by the assumption. We now check the feasibility for the constraints of (18).

$$\hat{Q}_0(x, u) = \hat{Q}'_0(x, u) \leq \mathcal{T}_u \hat{V}'_0(x, u) = \mathcal{T}_u \hat{V}_0(x, u),$$

for all $x \in \mathcal{X}$ and $u \in \mathcal{U}$, and for $j = 1, \dots, M-1$ we have,

$$\hat{Q}_j(x, u) = \mathcal{T}_u \hat{V}'_j(x, u) \leq \mathcal{T}_u \hat{V}_j(x, u),$$

for all $x \in \mathcal{X}$ and $u \in \mathcal{U}$, thus (18a) are satisfied. We have that for $j = 0, \dots, M-2$,

$$\hat{V}_j(x) = \hat{V}'_j(x) \leq \mathcal{T}_u \hat{V}'_{j+1}(x, u) = \hat{Q}_{j+1}(x, u),$$

for all $x \in \mathcal{X}$ and $u \in \mathcal{U}$, thus (18b) are also satisfied. Finally,

$$\hat{V}_{M-1}(x) = \hat{V}'_{M-1}(x) \leq \hat{Q}'_0(x, u) = \hat{Q}_0(x, u),$$

for all $x \in \mathcal{X}$ and $u \in \mathcal{U}$, thus (18c) is also satisfied, and the considered decision variables are feasible for problem (18). As $\hat{Q}_0 = \hat{Q}'_0$, the objective values are equal. ■

APPENDIX G

COMPUTING V^* , $\hat{V}$, μ , AND LYAPUNOV FUNCTIONS

This appendix provides additional details for the numerical example of Section VI-A.

The value function was computed on the interval $\mathcal{X} = [-12\sigma_\nu, 12\sigma_\nu]$ at 10^4 evenly spaced discretization points. The $V_{\hat{\pi}}$ and $\tilde{\mu}$ are computed for 10^4 x_0 samples from ν . The expectation with respect to ξ is empirically evaluated using 10^4 extractions from the disturbance process, different for each x_0 , and each is simulated for 10^3 time steps. The boundary of $\mathcal{X} = [-12\sigma_\nu, 12\sigma_\nu]$ was not reached by any sample.

Fig. 3: Set of Lyapunov functions, parametrized by the quadratic coefficient, for the one dimensional example of Section VI-A and the corresponding β_V .

Fig. 4: Right-hand-side of Theorem 4.7 for the one dimensional example of Section VI-A, evaluated for the choice $c(\cdot) = \nu(\cdot)$. The circles mark the minimum for each curve, showing that for each M , a different Lyapunov function achieves the tightest bound.

To explain the computation of Lyapunov functions, first recall that the restricted function space used for the one dimensional example is the space of univariate quadratics, with $p \in \mathbb{R}$ as the quadratic coefficient, $s \in \mathbb{R}$ as the constant offset, and the linear term omitted. From the definition of β_V and the H_V operator, it is clear that if a function V is a Lyapunov function then $\alpha V : x \mapsto \alpha V(x)$, with $\alpha \in \mathbb{R}_{++}$, is also a Lyapunov function. Moreover, the right-hand-side of Theorem 4.7 is unchanged by this positive scaling. Thus, without loss of generality we fix $s=1$ and parametrize candidate Lyapunov functions by the quadratic co-efficient.

To compute the set of Lyapunov functions and their corresponding β_V value, we take a brute force approach. As discussed in Section IV-C, a constant function, i.e., $p = 0$, is a Lyapunov function with $\beta_V = \gamma$. For this system, with stable linear dynamics, β_V increases with p . To find the set of Lyapunov functions, we increase $p > 0$ in small increments, and compute the value of β_V by discretizing the state space on a sufficiently large interval. The relationship of β_V versus p is shown in Fig. 3.

All Lyapunov functions yield a valid bound, and the Lyapunov with the tightest bound changes based on the number of Bellman inequality iterations M . To provide some insight, Fig. 4 shows the right-hand-side of Theorem 4.7 for the choice $c(\cdot) = \nu(\cdot)$ versus β_V for this example.

APPENDIX H REFORMULATION OF BELLMAN INEQUALITY

This appendix provides a sufficient reformulation of the Bellman Inequality that is used in the numerical examples for solving programs (13) and (18) to find an approximate value function and Q-function respectively. See Section VI-B for the definitions of A , B_u , and B_ξ as the linear dynamics, and (20) for the specification of the quadratic basis functions. We introduce $\underline{u}_i, \bar{u}_i \in \mathbb{R}$, $i=1, \dots, n_u$, to denote the lower and upper bounds that describe each coordinate of the $\mathcal{U} \subseteq \mathbb{R}^{n_u}$ space. To concisely represent the quadratic stage cost we introduce the matrix $L \in \mathbb{R}^{(n_x+n_u+1) \times (n_x+n_u+1)}$ that takes the form $l(x, u) = [x^\top, u^\top, 1] L [x^\top, u^\top, 1]^\top$. The notation $\text{diag}(\cdot)$ places the vector argument on the diagonal of an otherwise zero matrix, and e_i is the standard basis column vector with 1 in the i^{th} element and zeros elsewhere, with the dimension clear from context.

Using this notation, each inequality of the form $\hat{Q}_j(x, u) \leq \mathcal{T}_u \hat{V}_j(x, u)$ for all $x \in \mathbb{R}^{n_x}$, $u \in \mathcal{U}$ is sufficiently reformulated as the following LMI:

$$0 \preceq - \begin{bmatrix} P_j^Q & \frac{1}{2} p_j^Q \\ \star & s_j^Q \end{bmatrix} + L + \gamma \begin{bmatrix} A^\top P_j A & A^\top P_j B_u & \frac{1}{2} A^\top p_j + A^\top P_j B_\xi \mathbb{E}[\xi] \\ \star & B_u^\top P_j B_u & \frac{1}{2} B_u^\top p_j + B_u^\top P_j B_\xi \mathbb{E}[\xi] \\ \star & \star & s_j + \text{tr} \left(B_\xi^\top P_j B_\xi \mathbb{E}[\xi \xi^\top] \right) \end{bmatrix} - \sum_{i=1}^{n_u} \lambda_i \begin{bmatrix} 0_{n_x \times n_x} & 0 & 0 \\ \star & -\text{diag}(e_i) & \frac{1}{2} (\underline{u}_i + \bar{u}_i) e_i \\ \star & \star & -\underline{u}_i \bar{u}_i \end{bmatrix},$$

where $\star$ indicates that the matrix is symmetric, and the $\lambda_i \in \mathbb{R}_+$, $i=1, \dots, n_u$, are the auxiliary variables introduced when using the S-procedure to reformulate the for all $u \in \mathcal{U}$ part of the constraint.

The objective function of programs (13) and (18) is linear in the decision variables and evaluation of the objective requires the first and second moments of the relevance weighting parameter. For the Q-function formulation, the objective is:

$$\max_{P_0^Q, p_0^Q, s_0^Q} \text{tr} \left(P_0^Q \Sigma_c \right) + \mu_c^\top p_0^Q + s_0^Q,$$

where $\mu_c \in \mathbb{R}^{n_x+n_u}$ and $\Sigma_c \in \mathbb{S}^{n_x+n_u}$ are the first and second moments of the measure $c(x, u)$, and $\text{tr}(\cdot)$ denotes the trace of a square matrix.

REFERENCES

- [1] R. E. Bellman, "On the theory of dynamic programming," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 38, no. 8, pp. 716–719, 1952.
- [2] D. P. Bertsekas, *Dynamic programming and optimal control*, 4th ed. Athena Scientific Belmont, MA, 2017.
- [3] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, Inc., Hoboken, New Jersey, 2005.
- [4] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Machine learning*, vol. 3, no. 1, pp. 9–44, 1988.
- [5] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Athena Scientific, 1996.
- [6] B. Luo, D. Liu, T. Huang, and D. Wang, "Model-free optimal tracking control via critic-only Q-learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 10, pp. 2134–2144, 2016.

- [7] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, Feb 2015.
- [8] P. J. Schweitzer and A. Seidmann, "Generalized polynomial approximations in Markovian decision processes," *Journal of Mathematical Analysis and Applications*, vol. 110, pp. 568–582, 1985.
- [9] 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.
- [10] 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, 2006, pp. 243–256.
- [11] P. Beuchat, A. Georghiou, and J. Lygeros, "Alleviating tuning sensitivity in approximate dynamic programming," in *European Control Conference (ECC)*, Aalborg, June 2016, pp. 1616–1622.
- [12] C. J. C. H. Watkins, "Learning from delayed rewards," Ph.D. dissertation, King's College, University of Cambridge England, May 1989.
- [13] B. Luo, Y. Yang, and D. Liu, "Adaptive Q-learning for data-based optimal output regulation with experience replay," *IEEE Transactions on Cybernetics*, 2018.
- [14] A. M. Devraj and S. Meyn, "Zap Q-learning," in *Advances in Neural Information Processing Systems (NIPS)*, December 2017, pp. 2235–2244.
- [15] 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.
- [16] C. Novoa and R. Storer, "An approximate dynamic programming approach for the vehicle routing problem with stochastic demands," *European Journal of Operational Research*, vol. 196, no. 2, pp. 509–515, 2009.
- [17] B. Stellato, T. Geyer, and P. J. Goulart, "High-speed finite control set model predictive control for power electronics," *IEEE Transactions on Power Electronics*, vol. 32, no. 5, pp. 4007–4020, 2017.
- [18] V. V. Desai, V. F. Farias, and C. C. Moallemi, "Approximate dynamic programming via a smoothed linear program," *Operations Research*, vol. 60, no. 3, pp. 655–674, 2012.
- [19] O. Hernández-Lerma and J. B. Lasserre, *Discrete-time Markov control processes: basic optimality criteria*. Springer Science & Business Media, New York, 1996.
- [20] W. B. Powell, *Approximate Dynamic Programming, Solving the Curses of Dimensionality*, 2nd ed. John Wiley & Sons, Inc., Hoboken, New Jersey, 2011.
- [21] —, "What you should know about approximate dynamic programming," *Naval Research Logistics (NRL)*, vol. 56, no. 3, pp. 239–249, February 2009.
- [22] B. O'Donoghue, Y. Wang, and S. Boyd, "Min-max approximate dynamic programming," in *Computer-Aided Control System Design (CACSD), 2011 IEEE International Symposium on*. IEEE, 2011, pp. 424–431.
- [23] —, "Iterated approximate value functions," in *European Control Conference (ECC)*. Zürich, Switzerland: IEEE, July 2013, pp. 3882–3888.
- [24] C. Savorgnan, J. B. Lasserre, and M. Diehl, "Discrete-time stochastic optimal control via occupation measures and moment relaxations," in *Conference on Decision and Control (CDC), held jointly with the Chinese Control Conference (CCC)*, Shanghai, December 2009, pp. 519–524.
- [25] T. Summers, K. Kunz, N. Kariotoglou, M. Kamgarpour, S. Summers, and J. Lygeros, "Approximate dynamic programming via sum of squares programming," in *European Control Conference (ECC)*, Zürich, Switzerland, July 2013, pp. 191–197.
- [26] 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.
- [27] A. Keshavarz and S. Boyd, "Quadratic approximate dynamic programming for input-affine systems," *International Journal of Robust and Nonlinear Control*, vol. 24, no. 3, pp. 432–449, 2012.
- [28] T. Sutter, P. M. Esfahani, and J. Lygeros, "Approximation of constrained average cost Markov control processes," in *Conference on Decision and Control (CDC)*, Los Angeles, California, USA, December 2014, pp. 6597–6602.
- [29] N. Kariotoglou, S. Summers, T. Summers, M. Kamgarpour, and J. Lygeros, "Approximate dynamic programming for stochastic reachability," in *European Control Conference (ECC)*, Zürich, Switzerland, July 2013, pp. 584–589.

- [30] N. Kariotoglou, M. Kamgarpour, T. H. Summers, and J. Lygeros, "Upper bounds for the reach-avoid probability via robust optimization," *arXiv preprint arXiv:1506.03371*, 2015.
- [31] S. Darbha, K. Krishnamoorthy, M. Pachter, and P. Chandler, "State aggregation based linear programming approach to approximate dynamic programming," in *Conference on Decision and Control (CDC)*, Atlanta, December 2010, pp. 935–941.
- [32] K. Krishnamoorthy, M. Pachter, S. Darbha, and P. Chandler, "Approximate dynamic programming with state aggregation applied to UAV perimeter patrol," *International Journal of Robust and Nonlinear Control*, vol. 21, no. 12, pp. 1396–1409, 2011.
- [33] A. Shapiro, D. Dentcheva, and A. Ruszczyński, *Lectures on stochastic programming: modeling and theory*. MOS-SIAM Series on Optimization, 2014, vol. 16.
- [34] J. B. Rawlings and D. Q. Mayne, *Model Predictive Control: Theory and Design*. Nob Hill Publishing, LLC, Madison, WI, 2009.
- [35] E. F. Camacho and C. B. Alba, *Model Predictive Control*, 2nd ed. Springer-Verlag, London, 2007.
- [36] A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski, "Adjustable robust solutions of uncertain linear programs," *Mathematical Programming*, vol. 99, no. 2, pp. 351–376, 2004.
- [37] A. Georgioui, W. Wiesemann, and D. Kuhn, "Generalized decision rule approximations for stochastic programming via liftings," *Mathematical Programming*, vol. 152, pp. 301–338, 2015.
- [38] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for linear and hybrid systems*. Cambridge University Press, July 2017.
- [39] D. P. Bertsekas, "Dynamic programming and suboptimal control: A survey from ADP to MPC," *European Journal of Control*, vol. 11, no. 4-5, pp. 310–334, 2005.
- [40] B. Luo, D. Liu, H.-N. Wu, D. Wang, and F. L. Lewis, "Policy gradient adaptive dynamic programming for data-based optimal control," *IEEE Transactions on Cybernetics*, vol. 47, no. 10, pp. 3341–3354, 2017.
- [41] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton *et al.*, "Mastering the game of Go without human knowledge," *Nature*, vol. 550, no. 7676, p. 354, 2017.
- [42] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, 2nd ed. The MIT Press, 2018.
- [43] S. J. Bradtke and A. G. Barto, "Linear least-squares algorithms for temporal difference learning," *Machine Learning*, vol. 22, no. 1, pp. 33–57, 1996.
- [44] J. A. Boyan, "Technical update: Least-squares temporal difference learning," *Machine Learning*, vol. 49, no. 2, pp. 233–246, 2002.
- [45] J. N. Tsitsiklis and B. Van Roy, "An analysis of temporal-difference learning with function approximation," *Automatic Control, IEEE Transactions on*, vol. 42, no. 5, pp. 674–690, 1997.
- [46] V. R. Konda and J. N. Tsitsiklis, "On actor-critic algorithms," *SIAM Journal on Control and Optimization*, vol. 42, no. 4, pp. 1143–1166, 2003.
- [47] B. Luo, H.-N. Wu, and T. Huang, "Optimal output regulation for model-free quadcopter helicopter with multistep Q-learning," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 6, pp. 4953–4961, 2018.
- [48] P. N. Beuchat, "ADP toolbox code repository," June 2017. [Online]. Available: <https://beuchatp@gitlab.ethz.ch/ADP/ADP-Toolbox.git>
- [49] P. N. Beuchat, A. Georgioui, and J. Lygeros, "Performance guarantees for model-based approximate dynamic programming in continuous spaces," *arXiv:1602.07273v3 [cs.LG]*, Aug 2018.

applications in the areas of building control, and coordinated flight.

on the development of tractable computational methods for the solution of stochastic and robust optimization problems, as well as applications in operations management, finance and energy.

U.C. Berkeley. Between 2000 and 2003 he was a University Lecturer at the Department of Engineering, University of Cambridge, U.K., and a Fellow of Churchill College. Between 2003 and 2006 he was an Assistant Professor at the Department of Electrical and Computer Engineering, University of Patras, Greece. In July 2006 he joined the Automatic Control Laboratory at ETH Zurich, where he is currently serving as the Head of the Automatic Control Laboratory and the Head of the Department of Information Technology and Electrical Engineering. His research interests include modelling, analysis, and control of hierarchical, hybrid, and stochastic systems, with applications to biochemical networks, automated highway systems, air traffic management, power grids and camera networks. John Lygeros is a Fellow of the IEEE, and a member of the IET and the Technical Chamber of Greece; since 2013 he serves as the Treasurer of the International Federation of Automatic Control.

Paul Beuchat received the B.Eng. degree in mechanical engineering and B.Sc. in physics from the University of Melbourne, Australia, in 2008, and the M.Sc. degree in robotics, systems and control from ETH Zürich, Switzerland, in 2014, where he is currently working towards the Ph.D degree at the Automatic Control Laboratory. From 2009–2012 he was as a subsurface engineer for ExxonMobil. His research interests are control and optimization of large scale systems, with a focus towards developing approximate dynamic programming techniques for

Angelos Georgioui received his M.Sci. degree in Mathematics in 2008 and his Ph.D. degree in Operations Research at the Department of Computing in 2012, both from Imperial College London. From 2012–2013, he was a postdoctoral researcher at the Process Systems Engineering Laboratory at MIT, and from 2013–2016 a post-doctoral researcher at the Automatic Control Laboratory at ETH Zurich. He is currently an Assistant Professor of Operations Management in the Desautels Faculty of Management at McGill University. His research focuses

John Lygeros completed a B.Eng. degree in electrical engineering in 1990 and an M.Sc. degree in Systems Control in 1991, both at Imperial College of Science Technology and Medicine, London, U.K.. In 1996 he obtained a Ph.D. degree from the Electrical Engineering and Computer Sciences Department, University of California, Berkeley. During the period 1996–2000 he held a series of post-doctoral researcher appointments at the Laboratory for Computer Science, M.I.T., and the Electrical Engineering and Computer Sciences Department at