2013 European Control Conference (ECC)
July 17-19, 2013, Zürich, Switzerland.

Iterated Approximate Value Functions

Brendan O'Donoghue Yang Wang Stephen Boyd

Abstract— **In this paper we introduce a control policy which we refer to as the iterated approximate value function policy. The generation of this policy requires two stages, the first one carried out off-line, and the second stage carried out on-line. In the first stage we simultaneously compute a trajectory of moments of the state and action and a sequence of approximate value functions optimized to that trajectory. The next stage is to perform control using the generated sequence of approximate value functions. This yields a time-varying policy, even in the case where the optimal policy is time-invariant.**

We restrict our attention to the case with linear dynamics and quadratically representable stage cost function. In this case the pre-computation stage requires the solution of a semidefinite program (SDP). Finding the control action at each time-period requires solving a small convex optimization problem which can be carried out quickly. We conclude with some examples.

I. Introduction

We consider an infinite horizon discounted stochastic control problem with full state information. In general this problem is difficult to solve exactly, although there are some special cases in which it is tractable. When the state and action space are finite and not too large, it is readily solved. Another example is the case of linear dynamics and convex quadratic stage cost, possibly with linear constraints. In this case the optimal action is affine in the state variable, and the coefficients are readily computed.

A general method for solving stochastic control problems is dynamic programming (DP). DP relies on characterizing the value function of the stochastic control problem, which is the expected cost incurred by an optimal policy starting from the given state. The optimal policy can then be written as an optimization problem involving the stage cost and the value function. However, in most cases the value function is hard to represent, let alone compute. Even in cases where this is not the case, it may be hard to solve the optimization problem that gives the policy.

In such cases a common alternative is to use approximate dynamic programming (ADP). In ADP the value function is replaced with a surrogate function, often referred to as an approximate value function (AVF). The approximate policy is found by solving the policy optimization problem, with the true value function replaced with the surrogate. The goal in ADP is to choose an approximate value function so that the problem of computing the approximate policy is tractable and the approximate policy performs well. In this paper, we introduce an approximate dynamic programming policy, in which the surrogate function we use to find the control action changes in each time step.

The policy we develop in this paper relies on parametrizing a family of lower bounds on the true value function. The condition we use to guarantee lower boundedness is referred to as the iterated Bellman inequality [1], and is related to the 'linear programming approach' to ADP [2], [3], [4]. In a series of recent papers Wang and Boyd demonstrated a technique for generating performance bounds for a class of stochastic control problems using the iterated Bellman inequality. Their method derives a numerical performance bound via the solution of an optimization problem. As a by-product of the optimization problem they generate a sequence of functions, each of which is a pointwise underestimator of the true value function. In this paper we justify the use of these functions as a sequence of approximate value functions. We do this by showing that the dual variables of the optimization problem correspond to a trajectory of first and second moments of the state and action of the system, under the policy obtained by the sequence of AVFs. We restrict our attention to the case with linear dynamics and quadratically representable stage cost, in which case the performance bound problem can be expressed as a semidefinite program (SDP). We conclude with some numerical examples to illustrate the technique.

II. Stochastic control

We begin by briefly reviewing the basics of stochastic control and the dynamic programming solution. For more detail the interested reader is referred to, *e.g.*, [5], [6]. We consider a discrete time dynamical system, with dynamics described by

$$x_{t+1} = f(x_t, u_t, w_t), \quad t = 0, 1, \dots, \tag{1}$$

where $x_t \in \mathcal{X}$ is the system state, $u_t \in \mathcal{U}$ is the control input or action, $w_t \in \mathcal{W}$ is an exogenous noise or disturbance, all at time t, and $f : \mathcal{X} \times \mathcal{U} \times \mathcal{W} \to \mathcal{X}$ is the state transition function. The noise terms w_t are independent identically distributed (IID), with known distribution. The initial state x_0 is also random with known distribution, and is independent of w_t.

The stage cost function is denoted $\ell : \mathcal{X} \times \mathcal{U} \to \mathbf{R} \cup \{+\infty\}$, where the infinite values of ℓ encode constraints on the states and inputs: The state-action constraint set is $\mathcal{C} = \{(z, v) \mid \ell(z, v) < \infty\} \subseteq \mathcal{X} \times \mathcal{U}$. (The problem is unconstrained if $\mathcal{C} = \mathcal{X} \times \mathcal{U}$.)

We consider causal state feedback control policies of the form

$$u_t = \phi_t(x_t), \quad t = 0, 1, \dots,$$

where $\phi_t : \mathcal{X} \to \mathcal{U}$ is the *control policy* or *state feedback function* at time t. The *stochastic control problem* is to choose ϕ_t in order to minimize the infinite horizon discounted cost

$$J_\phi = \mathbf{E} \sum_{t=0}^{\infty} \gamma^t \ell(x_t, \phi_t(x_t)), \tag{2}$$

where $\gamma \in (0, 1)$ is a discount factor. The expectations are over the noise terms w_t, $t = 0, 1, \ldots$, and the initial state x_0. We assume here that the expectation and limits exist, which is the case under various technical assumptions [5], [6]. We denote by $J^\star$ the optimal value of the stochastic control problem, *i.e.*, the infimum of J_ϕ over all policies $\phi : \mathcal{X} \to \mathcal{U}$. When the control policy functions ϕ_t do not depend on t, they are called time-invariant. For the stochastic control problem we consider it can be shown that there is always an optimal policy that is time-invariant.

A. Dynamic programming

In this section we briefly review the dynamic programming characterization of the solution to the stochastic control problem. For more details, see [5], [6].

The *value function* of the stochastic control problem, $V^\star : \mathcal{X} \to \mathbf{R} \cup \{\infty\}$, is given by

$$V^\star(z) = \inf_\phi \mathbf{E} \left(\sum_{t=0}^{\infty} \gamma^t \ell(x_t, \phi(x_t)) \right),$$

subject to the dynamics (1) and $x_0 = z$; the infimum is over all policies $\phi : \mathcal{X} \to \mathcal{U}$, and the expectation is over w_t for $t = 0, 1, \ldots$. The quantity $V^\star(z)$ is the expected cost incurred by an optimal policy, when the system is started from state z at time $t = 0$. The optimal total discounted cost is given by

$$J^\star = \mathop{\mathbf{E}}_{x_0} V^\star(x_0). \tag{3}$$

It can be shown that the value function is the unique fixed point of the Bellman equation [7]

$$V^\star(z) = \inf_v \left(\ell(z, v) + \gamma \mathop{\mathbf{E}}_w V^\star(f(z, v, w)) \right)$$

for all $z \in \mathcal{X}$. We can write the Bellman equation in the form

$$V^\star = \mathcal{T} V^\star, \tag{4}$$

where we define the Bellman operator $\mathcal{T}$ as

$$(\mathcal{T} g)(z) = \inf_v \left(\ell(z, v) + \gamma \mathop{\mathbf{E}}_w g(f(z, v, w)) \right)$$

for any $g : \mathcal{X} \to \mathbf{R} \cup \{+\infty\}$. Moreover, iteration of the Bellman operator, starting from any initial function, converges to the value function $V^\star$ (see [5], [6] for many more technical details and conditions). This procedure is referred to as value iteration.

A time-invariant optimal policy for the stochastic control problem is given by

$$\phi^\star(z) \in \mathop{\mathrm{argmin}}_v \left(\ell(z, v) + \gamma \mathop{\mathbf{E}}_w V^\star(f(z, v, w)) \right), \tag{5}$$

for all $z \in \mathcal{X}$. (We drop the subscript t since this policy is time-invariant.)

B. Approximate dynamic programming

In many cases of interest, it is intractable to compute (or even represent) the value function $V^\star$, let alone carry out the minimization required evaluate the optimal policy (5). A common alternative is to replace the value function with an *approximate value function* $\hat{V}$ [8], [9], [10]. The resulting policy, given by

$$\hat{\phi}(z) \in \mathop{\mathrm{argmin}}_v \left(\ell(z, v) + \gamma \mathop{\mathbf{E}}_w \hat{V}(f(z, v, w)) \right),$$

for all $z \in \mathcal{X}$, is called an *approximate dynamic programming* (ADP) policy. When $\hat{V} = V^\star$, the ADP policy is optimal. The goal of approximate dynamic programming is to find a $\hat{V}$ for which the ADP policy can be easily evaluated (for instance, by solving a convex optimization problem), and also attains near-optimal performance. We can also consider time-varying ADP policies, obtained from a sequence of approximate value functions $\hat{V}_t$, which results in a time-varying AVF policy. While an optimal policy can always be chosen to be time-invariant, a time-varying AVF policy may give better control performance than a time-invariant AVF policy.

III. Iterated approximate value function policy

In this section we introduce the iterated AVF policy. We begin by reviewing the iterated Bellman inequality and discuss how it can be used to generate performance bounds on stochastic control problems. Finally we introduce the iterated AVF policy.

A. Iterated Bellman inequality

Any function which satisfies the Bellman inequality,

$$V \leq \mathcal{T} V, \tag{6}$$

where the inequality is pointwise, is a guaranteed pointwise lower bound on the true value function (under some additional mild technical conditions) [2], [3], [4], [11], [5], [6]. The basic condition works as follows. Suppose $V : \mathcal{X} \to \mathbf{R}$ satisfies $V \leq \mathcal{T} V$. Then by the monotonicity of the Bellman operator and convergence of value iteration [5], [6], we have

$$V \leq \lim_{k \to \infty} \mathcal{T}^k V = V^\star,$$

so any function that satisfies the Bellman inequality must be a value function underestimator.

We can derive a weaker condition for being a lower bound on $V^\star$ by considering an iterated form of the Bellman inequality. Suppose we have a sequence of functions $V_t : \mathcal{X} \to \mathbf{R}$, $t = 0, \ldots, T+1$, that satisfy a chain of Bellman inequalities

$$V_0 \leq \mathcal{T} V_1, \quad V_1 \leq \mathcal{T} V_2, \quad \ldots \quad V_T \leq \mathcal{T} V_{T+1}, \tag{7}$$

with $V_T = V_{T+1}$. Then, using similar arguments as before, we can show that $V_t \leq V^\star$ for $t = 0, \dots, T+1$. The condition is weaker since any function feasible for the single Bellman inequality is also feasible for the iterated Bellman inequality.

If we parametrize the functions to be linear combinations of k fixed basis functions $V^{(i)} : \mathcal{X} \to \mathbf{R}$ with coefficient vectors $\alpha_t \in \mathbf{R}^k$, *i.e.*,

$$V_t = \sum_{i=1}^{k} \alpha_{ti} V^{(i)}, \tag{8}$$

for $t = 0, \dots, T+1$, then the Bellman inequalities lead to convex constraints on the coefficients α_t. To see this, we write the Bellman inequality relating V_t and V_{t+1} as

$$V_t(z) \leq \inf_v \left(\ell(z, v) + \gamma \mathop{\mathbf{E}}_w V_{t+1}(f(z, v, w)) \right),$$

for all $z \in \mathcal{X}$. For each z, the left hand side is linear in α_t, and the right hand side is a concave function of α_{t+1}, since it is the infimum over a family of affine functions. Hence, the set of α_t, $t = 0, \dots, T+1$ that satisfy the iterated Bellman inequalities (7) is convex [12].

B. Performance bound

Now that we have a tractable condition on value function lower-boundedness we can use it to generate a performance bound on the stochastic control problem, since if $V_0 : \mathcal{X} \to \mathbf{R}$ satisfies $V_0(x) \leq V^\star(x)$ for all $x \in \mathcal{X}$, then

$$J^{\text{lb}} = \mathop{\mathbf{E}}_{x_0} V_0(x_0) \leq \mathop{\mathbf{E}}_{x_0} V^\star(x_0) = J^\star.$$

To find the best (*i.e.*, largest) lower bound we solve the following problem:

$$\begin{array}{ll} \text{maximize} & \mathbf{E}_{x_0} V_0(x_0) \\ \text{subject to} & V_t \leq \mathcal{T} V_{t+1}, \quad t = 0, \dots, T \\ & V_T = V_{T+1} \end{array} \tag{9}$$

over variables $\alpha_0, \dots, \alpha_{T+1}$. Since the iterated Bellman condition is a convex constraint on the coefficients $\alpha_t \in \mathbf{R}^K$, $t = 0, \dots, T+1$, and the objective is linear in α_0 this is a convex optimization problem [12]. For (much) more detail on deriving bounds for stochastic control problems see [10], [1], [13] and the references therein. Note that in [1], where the iterated Bellman inequality was first introduced, the authors used $V_0 = V_{T+1}$ as the terminal constraint. We replace that with $V_T = V_{T+1}$ here, which generally gives better numerical bounds.

C. Policy

By solving the performance bound problem (9) we obtain a sequence of approximate value functions $V_0, \dots, V_{T+1}$, each of which is a lower bound on the true value function. The iterated AVF policy is given by

$$\phi_t(x) \in \mathop{\text{argmin}}_u \left(\ell(x, u) + \gamma \mathbf{E} V_{t+1}(f(x, u, w)) \right) \tag{10}$$

for $0 \leq t \leq T$, and

$$\phi_t(x) \in \mathop{\text{argmin}}_u \left(\ell(x, u) + \gamma \mathbf{E} V_{T+1}(f(x, u, w)) \right) \tag{11}$$

for $t > T$.

Note that for the problem we consider an optimal policy is time-invariant. However, the iterated AVF policy is *time-varying*. It may be advantageous to use a time-varying policy because, typically, an approximate value function cannot be a good approximation of the true value function everywhere, so knowledge about the initial state of the system, and subsequent states under the iterated AVF policy, can be exploited to our advantage. The rest of this paper is a justification for the use of this time-varying policy in the particular case of linear dynamics and quadratically representable stage cost.

We briefly mention another policy, the pointwise maximum policy:

$$\phi(x) \in \mathop{\text{argmin}}_u \left(\ell(x, u) + \gamma \max_{t=0,\dots,T} \mathbf{E} V_t(f(x, u)) \right). \tag{12}$$

Note that this policy is time-invariant. Since each V_t is an underestimator of the true value function, the pointwise maximum of these is also an underestimator and is at least as good an approximation of the true value function as any individual V_t. However, this policy is much more expensive to compute, and moreover its complexity grows with horizon T.

IV. Quadratically representable case

We restrict our attention to the case with linear dynamics and quadratically representable stage cost and constraint set. We consider this limited case for simplicity, but the results in this paper extend to other cases with some minor modifications, such as time-varying dynamics, random dynamics, and a finite horizon.

We consider the case where the state and action spaces are finite dimensional vector spaces, *i.e.*, $x_t \in \mathbf{R}^n$ and $u_t \in \mathbf{R}^m$, and the dynamics equation has the form

$$x_{t+1} = f(x_t, u_t, w_t) = Ax_t + Bu_t + w_t, \quad t = 0, 1, \dots$$

for some matrices $A \in \mathbf{R}^{n \times n}$ and $B \in \mathbf{R}^{n \times m}$. We will write the dynamics as

$$G \begin{bmatrix} x_{t+1} \\ u_{t+1} \\ 1 \end{bmatrix} = F \begin{bmatrix} x_t \\ u_t \\ 1 \end{bmatrix} + \begin{bmatrix} w_t \\ 1 \end{bmatrix}$$

where

$$F = \begin{bmatrix} A & B & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad G = \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

We pad the state-action vector with an additional 1 so that it has dimension $l = n + m + 1$, which will allow more compact notation in the sequel. We assume that the noise term has zero mean, *i.e.*, $\mathbf{E} w_t = 0$ for all t, and has second moment $\mathbf{E}\, w_t w_t^T = \hat{W}$ for all t. For compactness of notation we let

$$\mathbf{E}\left[\begin{array}{c} w_t \\ 0 \end{array}\right]\left[\begin{array}{c} w_t \\ 0 \end{array}\right]^T = \left[\begin{array}{cc} \hat{W} & 0 \\ 0 & 0 \end{array}\right] = W.$$

We consider stage cost functions of the form

$$\ell(x,u) = \begin{cases} \bar{\ell}(x,u) & (x,u) \in \mathcal{C} \\ \infty & \text{otherwise,} \end{cases}$$

where

$$\bar{\ell}(x,u) = \left[\begin{array}{c} x \\ u \\ 1 \end{array}\right]^T L \left[\begin{array}{c} x \\ u \\ 1 \end{array}\right],$$

$L \in \mathbf{S}^l$ (the set of $l \times l$ symmetric matrices), is a convex quadratic function and where $\mathcal{C}$ denotes the feasible set of state-action pairs. We assume that we can write the feasible set as the intersection of $k+1$ convex quadratic constraint sets, *i.e.*,

$$\mathcal{C} = \left\{ (x,u) \,\middle|\, \left[\begin{array}{c} x \\ u \\ 1 \end{array}\right]^T \Sigma_i \left[\begin{array}{c} x \\ u \\ 1 \end{array}\right] \leq 0, \; i = 0, \ldots, k \right\} \tag{13}$$

where $\Sigma_i \in \mathbf{S}^l$, $i = 0, \ldots, k$.

We shall parametrize the approximate value functions to be convex quadratic functions of the state, *i.e.*,

$$V_t(x) = \left[\begin{array}{c} x \\ 1 \end{array}\right]^T P_t \left[\begin{array}{c} x \\ 1 \end{array}\right]$$

for some $P_t \in \mathbf{S}^{n+1}$. With this choice of approximate value function the parameter P_t takes role of α_t in (8) and (9). Thus the iterated Bellman inequalities are a set of convex constraints on the parameters $P_0, \ldots, P_{T+1}$. This choice of approximate value function also ensures that the policy problems (10) and (11) are convex optimization problems and can be solved efficiently [14], [12].

A. Iterated Bellman inequalities

With the notation we have established, we can write

$$\mathbf{E}\, V(x_{t+1}) = \mathbf{E}\, V(Ax_t + Bu_t + w_t) = \left[\begin{array}{c} x_t \\ u_t \\ 1 \end{array}\right]^T F^T P F \left[\begin{array}{c} x_t \\ u_t \\ 1 \end{array}\right] + \mathbf{Tr}(PW).$$

We denote by $\mathcal{U}(z) = \{u \mid (z,u) \in \mathcal{C}\}$ the set of feasible actions at a given state z. The single Bellman inequality is written

$$V(x) \leq \min_{u \in \mathcal{U}(x)} \left(\ell(x,u) + \gamma\, \mathbf{E}\, V(Ax + Bu + w)\right),$$

for all $x \in \mathcal{X}$, which with our notation is

$$\left[\begin{array}{c} x \\ 1 \end{array}\right]^T P \left[\begin{array}{c} x \\ 1 \end{array}\right] \leq \gamma\, \mathbf{Tr}(PW) + \min\nolimits_{u \in \mathcal{U}(x)} \left[\begin{array}{c} x \\ u \\ 1 \end{array}\right]^T (L + \gamma F^T P F) \left[\begin{array}{c} x \\ u \\ 1 \end{array}\right]$$

for all x such that $\mathcal{U}(x)$ is non-empty, or equivalently

$$\left[\begin{array}{c} x \\ u \\ 1 \end{array}\right]^T (L + \gamma F^T P F - G^T P G) \left[\begin{array}{c} x \\ u \\ 1 \end{array}\right] + \gamma\, \mathbf{Tr}(PW) \geq 0 \tag{14}$$

for all $(x,u) \in \mathcal{C}$. This constraint is convex (indeed affine) in the variable P. However it is semi-infinite, since it is a family of constraints parametrized by the (infinite) set $(x,u) \in \mathcal{C}$. The $\mathcal{S}$-procedure [15] provides a sufficient condition that ensures (14) holds for all states. By using the $\mathcal{S}$-procedure, we can approximate the iterated Bellman inequalities as linear matrix inequalities (LMIs) and, in turn, (9) can be approximated as an SDP, which can be solved efficiently. Since the $\mathcal{S}$-procedure is sufficient, the resulting approximate value functions found by solving the SDP will still be pointwise underestimators of the true value function, and the numerical performance bound will still be valid.

The set $\mathcal{C}$ is defined by quadratic inequalities parametrized by Σ_i, $i = 0, \ldots, k$. From the $\mathcal{S}$-procedure we have that the Bellman inequality will hold if there exists $\lambda \in \mathbf{R}_+^{k+1}$ such that

$$L + \gamma F^T P F - G^T P G + \gamma\, \mathbf{Tr}(PW) e_l e_l^T \succeq -\textstyle\sum_{i=0}^k \lambda^i \Sigma_i$$

where e_l is the lth unit vector, λ^i denotes the ith component of λ, and $\succeq$ denotes matrix inequality.

The extension of the single Bellman inequality to the iterated case is written

$$L + \gamma F^T P_{t+1} F - G^T P_t G + \gamma\, \mathbf{Tr}(P_{t+1} W) e_l e_l^T \succeq -\textstyle\sum_{i=0}^k \lambda_t^i \Sigma_i$$

for $t = 0, \ldots, T$, along with the end condition $P_T = P_{T+1}$. This condition is convex in parameters $P_0, \ldots, P_{T+1} \in \mathbf{R}^{l \times l}$ and $\lambda_0, \ldots, \lambda_T \in \mathbf{R}_+^k$, in particular it is a linear matrix inequality (LMI) [15], [12].

B. Performance bound problem

In this section we will define the set $\mathcal{P}_t^T \subset \mathbf{S}_+^{n+1}$, which is the set of convex quadratics, parametrized by P_t, for which there exists $P_{t+1}, \ldots, P_{T+1} \in \mathbf{R}^{(n+1)\times(n+1)}$ that together satisfy $T+1-t$ iterated Bellman inequalities. Thus any $P_t \in \mathcal{P}_t^T$ defines a convex quadratic function that is a guaranteed lower bound on the true value function.

First, the iterated Bellman inequalities must hold, *i.e.*,

$$L + \gamma F^T P_{\tau+1} F - G^T P_\tau G + \gamma\, \mathbf{Tr}(P_{\tau+1} W) e_l e_l^T \succeq -\textstyle\sum_{i=0}^k \lambda_\tau^i \Sigma_i, \quad \tau = t, \ldots, T \tag{15}$$

where

$$\lambda_\tau \in \mathbf{R}_+^{k+1}, \quad \tau = t, \ldots, T. \tag{16}$$

For convexity of the approximate value functions we require

$$P_\tau = \left[\begin{array}{cc} \hat{P}_\tau & p_\tau \\ p_\tau^T & r_\tau \end{array}\right], \; \hat{P}_\tau \in \mathbf{S}_+^n, \quad \tau = t, \ldots, T+1. \tag{17}$$

Finally we require the terminal constraint

$$P_T = P_{T+1}. \tag{18}$$

We denote by $\mathcal{P}_t^T$ the set of parameters that satisfy these conditions, *i.e.*,

$$\begin{aligned}\mathcal{P}_t^T = \{P_t \mid \exists\ P_{t+1}, \dots, P_{T+1}, \lambda_t, \dots, \lambda_T, \\ \text{such that } (15), (16), (17), (18) \text{ are satisfied}\}.\end{aligned}$$

With this notation the problem of finding a lower bound on the performance of the optimal policy can be expressed as an SDP in the variables $P_0, \dots, P_{T+1}$ and $\lambda_0, \dots, \lambda_T$,

$$\begin{array}{ll} \text{maximize} & \mathbf{Tr}(P_0 X_0) \\ \text{subject to} & P_0 \in \mathcal{P}_0^T \end{array} \tag{19}$$

where

$$X_0 = \mathbf{E} \begin{bmatrix} x_0 \\ 1 \end{bmatrix} \begin{bmatrix} x_0 \\ 1 \end{bmatrix}^T$$

contains the first and second moments of the initial state.

V. Iterated Quadratic AVFs

The goal in this section is to justify the iterated AVF policy given in (10) and (11), *i.e.*, using the chain of value functions, arising from solving (19), as a sequence of approximate value functions. We assume throughout that (19) is feasible and that the optimal value is attained.

A. Relaxed policy problem

Here we introduce what we refer to as the relaxed policy problem. This is the problem of minimizing the *expected* cost in the policy problems (10) or (11), where instead of exact knowledge of the state we know only its first and second moments, and where we relax the constraints to hold *in expectation*. (The relaxed policy problem reduces to the standard policy problem when the state is known exactly). For an in-depth treatment on the use of moment relaxations in optimal control see [16]. This problem is written

$$\begin{array}{ll} \text{minimize} & \mathbf{E} \begin{bmatrix} x \\ u \\ 1 \end{bmatrix}^T (L + \gamma F^T P_{t+1} F) \begin{bmatrix} x \\ u \\ 1 \end{bmatrix} \\ \text{subject to} & \mathbf{E} \begin{bmatrix} x \\ 1 \end{bmatrix} \begin{bmatrix} x \\ 1 \end{bmatrix}^T = X_t \\ & \mathbf{E} \begin{bmatrix} x \\ u \\ 1 \end{bmatrix}^T \Sigma_i \begin{bmatrix} x \\ u \\ 1 \end{bmatrix} \leq 0, \quad i = 0, \dots, k \end{array}$$

where X_t contains the first and second moment information about the state at time t. If we let

$$Z_t = \mathbf{E} \left(\begin{bmatrix} x \\ u \\ 1 \end{bmatrix} \begin{bmatrix} x \\ u \\ 1 \end{bmatrix}^T \right) \in \mathbf{S}_+^l$$

we can write the problem as

$$\begin{array}{ll} \text{minimize} & \mathbf{Tr}(L + \gamma F^T P_{t+1} F) Z_t \\ \text{subject to} & G Z_t G^T = X_t \\ & Z_t \succeq 0 \\ & \mathbf{Tr}(\Sigma_i Z_t) \leq 0, \quad i = 0, \dots, k \end{array} \tag{20}$$

over variable $Z_t \in \mathbf{R}^{l \times l}$. The solution matrix $Z_t^\star$ contains the first and second moments of the state and action that minimize the expected cost-to-go, using V_{t+1} as the value function. Since we have relaxed the constraints to hold in expectation, we shall refer to Z_t as the *relaxed* second moment at time t.

B. Saddle point problems

In this subsection we shall show that the dual variables of the performance bound problem (19) correspond to the relaxed second moments of the state and action under the policy admitted by the sequence of approximate value functions. We will show that the sequence of approximate value functions is optimized to this trajectory, which concludes our justification.

We start by forming the partial Lagrangian of the problem, where we introduce dual variable $Z_0 \succeq 0$ for the Bellman inequality constraint relating P_0 and P_1, which is written

$$\begin{aligned} \mathcal{L}(P_0, P_1, Z_0) = \mathbf{Tr}(P_0 X_0) + \gamma\, \mathbf{Tr}(P_1 W) e_l^T Z_0 e_l \\ + \mathbf{Tr}\, Z_0 \left(L + \gamma F^T P_1 F - G^T P_0 G - \textstyle\sum_{i=0}^k \lambda_0^i \Sigma_i \right) \end{aligned}$$

where we have the additional constraint that $P_1 \in \mathcal{P}_1^T$. If we analytically maximize this over P_0 and λ_0 we obtain the following set of constraints on Z_0:

$$\mathcal{Z}_0 = \left\{ Z \mid G Z G^T = X_0,\ \mathbf{Tr}(\Sigma_i Z) \leq 0,\ Z \succeq 0 \right\}.$$

These constraints correspond exactly to the constraints in the relaxed second moment problem (20). The first constraint above requires Z_0 to be in consensus with the supplied second order information about the state, and the second constraint requires the state and action to satisfy the constraints (13) in expectation. From the first constraint we also have that $e_l^T Z_0 e_l = 1$. Making the appropriate substitutions we arrive at the saddle point problem

$$\begin{array}{ll} \text{minimize} & \mathbf{Tr}(L Z_0) + \gamma \max\limits_{P_1 \in \mathcal{P}_1^T} \mathbf{Tr}(F Z_0 F^T + W) P_1 \\ \text{subject to} & Z_0 \in \mathcal{Z}_0 \end{array} \tag{21}$$

over Z_0. If we swap the order of maximization and minimization in (21) we have

$$\begin{array}{ll} \text{maximize} & \gamma\, \mathbf{Tr}(P_1 W) + \min\limits_{Z_0 \in \mathcal{Z}_0} \mathbf{Tr}(L + \gamma F^T P_1 F) Z_0 \\ \text{subject to} & P_1 \in \mathcal{P}_1^T \end{array} \tag{22}$$

over variables $P_1, \dots, P_T$ and $\lambda_1, \dots, \lambda_{T-1}$. Assuming strong duality, problems (21) and (22) are equivalent and attain the same optimal value as (19). The second term in problem (22) is identical to problem (20) for $t = 0$. This implies that the optimal Z_0 is the optimal second moment of the *relaxed* problem (20) at $t = 0$ using V_1 as the value function, *i.e.*, we can interpret Z_0 as

$$Z_0 = \mathbf{E}\left(\begin{bmatrix} x_0 \\ u_0 \\ 1 \end{bmatrix}\begin{bmatrix} x_0 \\ u_0 \\ 1 \end{bmatrix}^T\right).$$

With this we can write

$$\begin{aligned} &FZ_0F^T + W \\ &\quad = \mathbf{E}\begin{bmatrix} Ax_0 + Bu_0 + w \\ 1 \end{bmatrix}\begin{bmatrix} Ax_0 + Bu_0 + w \\ 1 \end{bmatrix} \\ &\quad = \mathbf{E}\begin{bmatrix} x_1 \\ 1 \end{bmatrix}\begin{bmatrix} x_1 \\ 1 \end{bmatrix}^T, \end{aligned}$$

and thus we can rewrite the second term in problem (21) as

$$\max_{P_1 \in \mathcal{P}_1^T} \mathbf{Tr}(FZ_0F^T + W)P_1 = \max_{P_1 \in \mathcal{P}_1^T} \mathbf{E}\begin{bmatrix} x_1 \\ 1 \end{bmatrix}^T P_1 \begin{bmatrix} x_1 \\ 1 \end{bmatrix}.$$

This implies that the optimal P_1 is the quadratic lower bound that maximizes the expected cost over states at time $t = 1$ (and satisfies the iterated Bellman inequalities).

The above argument is repeated inductively: At iteration t we introduce a new dual variable Z_t for the Bellman inequality involving P_t and P_{t+1} and show that it corresponds to the relaxed second moment of the state and action at time t, provided the same holds for Z_{t-1}. Then, since the relaxed second moment of the state at time $t + 1$ is determined by applying the dynamics equations to Z_t, it follows that P_{t+1} maximizes the expected cost over states at time $t + 1$ (while satisfying the Bellman inequalities). This is repeated up to $t = T$. Finally, if the relaxed second moment of the state converges to a steady-state and the horizon T is large enough, then, by the argument above, P_T (and therefore P_{T+1}) is optimized to that steady-state distribution, which justifies the long-term policy (11).

VI. Examples

Here we introduce two examples to demonstrate the efficacy of the iterated AVF policy. For other practical examples of formulating the bound problem (9) as an SDP see [17], [18], [1], [10].

A. *One-dimensional example*

In this instance we take $n = m = 1$ and $\gamma = 0.99$, the dynamics were given by $x_{t+1} = x_t + u_t$ and the cost function was chosen to be

$$\ell(x, u) = \begin{cases} x^2 + (0.1)u^2 & |u| \le 1, \\ \infty & \text{otherwise.} \end{cases}$$

This allows us to visually inspect the approximate value functions and the true value function for this particular problem.

We solve the performance bound problem (19) with a horizon of $T = 25$ and with exact knowledge of the initial state, which we take to be $x_0 = 20$. Figure 1 shows the sequence of value functions generated by solving problem (19). The red circles are the trajectory of first moments of the state x_t, extracted from the dual variables Z_t, the blue curves are the corresponding quadratic value functions defined by P_t. Note that each quadratic is a good approximation in some regions, namely where the state is expected to be at that time, and a bad approximation in other regions. In this case we can find the true value function by discretizing the state and action spaces and performing value iteration [5], [6]. Figure 2 shows the approximate value function given by the pointwise maximum over all 25 quadratics in blue and the true value function in red. In this case the two are indistinguishable.

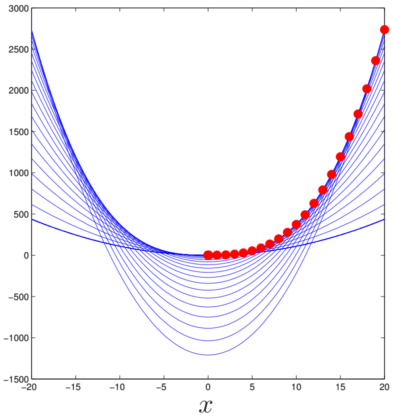

Fig. 1. AVF sequence and state trajectory.

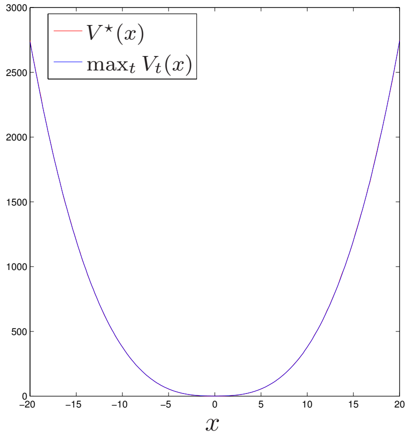

Fig. 2. True value function and pointwise max over AVFs.

We evaluated the performance of various policies using Monte Carlo simulation, starting from x_0. The iterated AVF policy obtained 2747.6, almost identical to the lower bound of 2745.0; a single approximate value function achieved 2750.1, a small but significant difference. The single AVF was generated by solving (9) with $T = 0$. The more computationally expensive model predictive control (MPC) policy [19], [20], [21], [22] with a lookahead of 50 time-steps, achieved a performance of 2745.1.

Policy	Performance
Lower bound	50737
MPC	50923
Iterated AVF	51286
Single AVF	52479

TABLE I
LOWER BOUND, AND POLICY PERFORMANCES.

B. Box constrained quadratic control

This example is similar to an example presented in [18]. We control a linear dynamical system with stage cost

$$\ell(x,u) = \begin{cases} x^T Qx + u^T Ru & \|u\|_\infty \leq 1 \\ \infty & \text{otherwise}, \end{cases}$$

where $Q \succeq 0$ and $R \succeq 0$, *i.e.*, our action is constrained to lie in a box at all time periods. The constraint that $\|u_t\|_\infty \leq 1$ can be rewritten as

$$(u_t)_i^2 \leq 1, \quad i = 1, \ldots, m.$$

We randomly generated a numerical instance with $n = 12$ and $m = 4$. The dynamics matrix A was randomly generated, and then scaled to have spectral radius 1. The horizon length for both MPC and the lower bound calculation was set to 50, the value of γ was chosen to be 0.99. We ran 1000 Monte Carlo simulations of 500 time periods to estimate performance of various policies. We compared the performance of the iterated AVF policy with a single AVF policy, where we solve (9) with $T = 0$. We also compared to MPC with a 50 step-lookahead. The results are summarized in table I. All computation was carried out on a 4-core Intel Xeon processor, with clock speed 3.4GHz and 16Gb of RAM, running Linux. The off-line pre-computation to generate the AVF policies was carried out using CVX [23] and required 8.1s for the $T = 50$ lookahead horizon. Custom interior point solvers were generated by CVXGEN [24] and used to solve the policy problems for both the AVF policies and the MPC policy. On average, it took 9ms to solve the MPC problem at each iteration, whereas it only took 83μs to solve the AVF policy problem, more than two orders of magnitude faster. From the lower bound we can certify that MPC is no more than 0.4% suboptimal, the iterated AVF policy is no more than 1.1% suboptimal, and the single AVF policy is no more than 3.4% suboptimal.

VII. SUMMARY

In this paper we introduced the iterated approximate value function policy, which is a time-varying policy even in the case where the optimal policy is time-invariant. The sequence of approximate value functions we use is derived from a performance bound problem, which, for the case we consider, is a convex optimization problem and thus tractable. We justified the use of this sequence of approximate value functions by considering the dual of the performance bound problem. We showed that the dual variables of this problem could be interpreted as a trajectory of moments of the state and action for the stochastic control problem. We concluded with some examples to show the performance of our policy.

REFERENCES

[1] Y. Wang and S. Boyd, "Approximate dynamic programming via iterated Bellman inequalities," http://www.stanford.edu/~boyd/papers/adp_iter_bellman.html, 2010, manuscript.

[2] A. Manne, "Linear programming and sequential decisions," *Management Science*, vol. 6, no. 3, pp. 259–267, Apr. 1960.

[3] P. Schweitzer and A. Seidmann, "Generalized polynomial approximations in Markovian decision process," *Journal of mathematical analysis and applications*, vol. 110, no. 2, pp. 568–582, Sep. 1985.

[4] D. De Farias and B. Van Roy, "The linear programming approach to approximate dynamic programming," *Operations Research*, vol. 51, no. 6, pp. 850–865, Nov. 2003.

[5] D. Bertsekas, *Dynamic Programming and Optimal Control: Volume 1*. Athena Scientific, 2005.

[6] ——, *Dynamic Programming and Optimal Control: Volume 2*. Athena Scientific, 2007.

[7] R. Bellman, *Dynamic Programming*. Dover Publications, 1957.

[8] D. Bertsekas and J. Tsitsiklis, *Neuro-Dynamic Programming*. Athena Scientific, 1996.

[9] W. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. J. Wiley & Sons, 2007.

[10] Y. Wang and S. Boyd, "Performance bounds for linear stochastic control," *Systems & Control Letters*, vol. 58, no. 3, pp. 178–182, Mar. 2009.

[11] D. de Farias and B. Van Roy, "A cost-shaping linear program for average-cost approximate dynamic programming with performance guarantees," *Mathematics of Operations Research*, vol. 31, no. 3, pp. 597–620, 2006.

[12] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, Sep. 2004.

[13] Y. Wang and S. Boyd, "Performance bounds and suboptimal policies for linear stochastic control via LMIs," *International Journal of Robust and Nonlinear Control*, vol. 21, no. 14, pp. 1710–1728, Sep. 2011.

[14] ——, "Fast evaluation of quadratic control-Lyapunov policy," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 4, pp. 939–946, Jul. 2011.

[15] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Society for Industrial and Applied Mathematics, 1994.

[16] C. Savorgnan, J. Lasserre, and M. Diehl, "Discrete-time stochastic optimal control via occupation measures and moment relaxations," in *Proc. 48th IEEE Conference on Decision and Control*, Dec. 2009, pp. 519–524.

[17] S. Boyd, M. Mueller, B. O'Donoghue, and Y. Wang, "Performance bounds and suboptimal policies for multi-period investment," http://www.stanford.edu/~boyd/papers/port_opt_bound.html, Jul. 2012, manuscript.

[18] B. O'Donoghue, Y. Wang, and S. Boyd, "Min-max approximate dynamic programming," in *Proceedings of the 2011 IEEE Multi-Conference on Systems and Control*, Sep. 2011, pp. 424–431.

[19] B. O'Donoghue, G. Stathopoulos, and S. Boyd, "A splitting method for optimal control," http://www.stanford.edu/~boyd/papers/oper_splt_ctrl.html, 2012, manuscript.

[20] C. Garcia, D. Prett, and M. Morari, "Model predictive control: Theory and practice," *Automatica*, vol. 25, no. 3, pp. 335–348, May 1989.

[21] J. Maciejowski, *Predictive Control with Constraints*. Prentice-Hall, 2002.

[22] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, Jun. 2000.

[23] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 1.21," http://cvxr.com/cvx, Apr. 2011.

[24] J. Mattingley and S. Boyd, "CVXGEN: A code generator for embedded convex optimization," *Optimization and Engineering*, vol. 13, no. 1, pp. 1–27, 2012.