SMART BUILDING REAL TIME PRICING FOR OFFERING LOAD-SIDE REGULATION SERVICE RESERVES

Enes Bilgin, Michael C. Caramanis, Ioannis Ch. Paschalidis

***Abstract*—Provision of Regulation Service (RS) reserves to Power Markets by smart building demand response has attracted attention in recent literature. This paper develops tractable dynamic optimal pricing algorithms for distributed RS reserve provision. It shows monotonicity and convexity properties of the optimal pricing policies and the associated differential cost function. Then, it uses them to propose and implement a modified Least Squares Temporal Differences (LSTD) Actor-Critic algorithm with a bounded and continuous action space. This algorithm solves for the best policy within a pre-specified broad family. In addition, the paper develops a novel Approximate Policy Iteration (API) algorithm and uses it successfully to optimize the parameters of an analytic policy function. Numerical results are obtained to demonstrate and compare the Actor-Critic and Approximate Policy Iteration algorithms, demonstrating that the novel API algorithm outperforms the Bounded LSTD Actor-Critic algorithm in both computational effort and policy minimum cost.**

***Index Terms*—Regulation Service Provision, Distributed Demand Management, Actor-Critic Algorithms, Approximate Policy Iteration**

I. INTRODUCTION

Regulation Service (RS) Reserves, a particular type of bi-directional Capacity Reserves, are secured routinely in the Hour Ahead Market by Independent System Operators (ISOs) to enable themselves to use them in real time for the purpose of maintaining the supply-demand energy balance. ISOs utilize RS reserves in real time by broadcasting a signal that is updated every few seconds and that specifies the utilization of the RS reserve that providers need to track by modulating their generation/demand ([1]). Conventional centralized generators have been so far the main providers of RS reserves. Nevertheless, participation of the demand side in the RS market is emerging and broad participation will most probably be essential with the increasing penetration of renewable energy resources into power systems whose volatility and intermittency have a commensurate impact on the required RS reserves ([2], [3]). Several models have been proposed for managing real time demand response in Smart Buildings for responding to RS signals in [4], [5], [6], including our past work in [7]. A main issue with these models is that they become intractable as the problem size increases. This prevents the implementation of tractable real size models, including for example multiple class appliances. The contribution of this paper is a more thorough investigation of the characteristics of our model proposed in [7], which enables the use of near-optimal approximate methods that render the realistic size models tractable.

In [7], we proposed a Dynamic Programming (DP) formulation where the Smart Building Operator (SBO) uses a pricing mechanism to modulate the rate of electricity consumption by building appliances. The resulting DP problem price control variable was discretized and the optimal policy solved by Linear Programming (LP) as described in [8]. We observed that the DP differential cost function appeared to be convex in the state variable that represents the RS signal value, and the optimal control policy exhibited monotonicity that appeared to fit an appropriately parameterized sigmoid function. This paper actually proves the aforementioned convexity and monotonicity characteristics for a simplified version of the problem. Based on the intuition provided by these properties, this paper proposes an "Actor-Critic" algorithm adapted to our problem, as well as a novel Approximate Policy Iteration (API) algorithm. The monotonicity and convexity properties are relied upon in selecting parameterized functional approximations that are used to solve the problem with the proposed algorithms.

In actor-critic algorithms, the parameters of a Randomized Stationary Policy (RSP) are optimized using policy gradient estimation. The critic uses a parameterized approximation of a value function and evaluates the policy based on the observations of the state and the cost along the simulation. The actor obtains a direction of improvement from the critic and updates the RSP parameters. Eventually, these parameters are optimized so that "good" actions are chosen with high probability. On the other hand, randomness is required for better exploration of the state and action spaces. In [9], Konda et al. use Temporal Differences (TD) in the critic to update the parameters of the value function approximation. In [10], Estanjini et al. adapt Least Squares TD (LSTD) in the critic and obtain a better rate of convergence. In this form of the actor-critic algorithm, RSPs need to be differentiable probability/distribution mass functions (p.m.f./p.d.f.). When the action space is discrete, Boltzmann-like distributions can be used to obtain a differentiable p.m.f. However, when the action space is bounded and continuous, which is the case in our problem, there is no p.d.f. that is differentiable over the whole action space. Discretization of the action space while using an actor-critic algorithm is a poor option since the state of the system does not tell very much about why to prefer one discrete action over another.

The authors are with the Center for Information and Systems Engineering, Boston University, College of Engineering. Research partially supported by the NSF under grants EFRI-1038230, CNS-1239021, and IIS-1237022, by the ARO under grants W911NF-11-1-0227 and W911NF-12-1-0390, and by the ONR under grant N00014-10-1-0952. Corresponding author email: mcaraman@bu.edu.

In literature, there are several approaches developed for the bounded and continuous action space case. One approach is to use a continuous distribution (such as Gaussian) as in [11] and [12], and, whenever the RSP generates an action outside of the action space, peg the action to the closest boundary. In [13], Kimura et al. point out that the algorithm easily fails with this approach when the action space is bounded, because randomness is lost when the mean value of the distribution moves out of the action space. Instead, they offer an alternative where the action generated by the RSP is rejected if it does not fall into the action space. This approach increases the computational burden in two ways: *(i)* the RSP is asked to continue to generate actions until an acceptable one is obtained, and *(ii)* one needs to calculate the probability that the RSP generates an action that falls into the action space, for which the authors provide an approximated numerical calculation. The approach proposed in this paper eliminates the aforementioned disadvantages. It employs a Gaussian distribution, with a parameterized mean, to generate an action related value that lies in $(-\infty, +\infty)$. This action related value is then transformed through a sigmoid function projection onto a feasible control, which is then used to simulate the next state. In order to keep the Gaussian mean within desired limits, we bound the values that the actor parameters can take. With a proper selection of the variance, randomness in the actions is preserved. We henceforth refer to this modified algorithm as the "Bounded LSTD Actor-Critic", and present results from its successful implementation.

Turning next to the alternative Approximate Policy Iteration (API) algorithm, we note that the challenge with solving the DP problem via the Linear Programming (LP) approach in [8] is that the number of the constraints in the LP is multiplied by the size of the discretized action space. To overcome the need to solve large LPs resulting from the discretization of the action space, we *(i)* approximate the policy function by a parameterized sigmoid analytic function, and *(ii)* adopt a policy iteration approach whose basic step, that evaluates the differential cost function associated with a tentative policy, requires the solution of an LP with constraints that are equal to the number of states, since now there is a single policy for a given tentative value of the sigmoid function parameters. The proposed API algorithm uses an efficient policy improvement step by overcoming a significant challenge. In the standard policy iteration algorithm, the policy improvement step requires a value iteration operation for each state of the system which guarantees an improvement in the average cost ([8]). However, when the action is generated by a parameterized policy function, updating these parameters may decrease the cost for some states while increasing it for others, and if the overall impact on the average cost is positive, the algorithm does not converge. The proposed API algorithm overcomes this problem by updating the parameters in a fashion that guarantees a decrease in the average cost. This allows it to converge, and usually it does so to a lower average cost than the average cost to which the actor-critic algorithm converges after a practical number of iterations.

The rest of the paper is organized as follows. In Section II, we revisit the Dynamic Programming (DP) problem formulation in [7] for completeness of this paper. In Section III, we prove convexity of the differential cost function and monotonicity of the optimal control policy. The sigmoid approximation of the optimal control policy proposed in [7] is also revisited in Section III. The structure implied by these properties motivates and inspires the Bounded LSTD Actor-Critic algorithm of Section IV and our novel Approximate Policy Iteration (API) algorithm of Section V. Section VI describes structural similarities and differences of the two algorithms, numerical results are provided and discussed in Section VII, and Section VIII concludes.

II. PROBLEM FORMULATION

The price based management of smart building flexible load consumption control problem proposed in [7] is summarized next, while the reader is referred to [7] for details. We consider a smart building operator (SBO) who has committed in the Hour Ahead Market to consume electricity at an average rate of A kW during a specific hour, and has promised to offer R kW of Regulation Service during the same hour. The SBO is thus obliged to track the RS signal $y \in [-1, 1]$, which is updated by the ISO every Δt seconds, by modulating the consumption in the building. The rate of total consumption in the building at time t is $n(t)r$, where $n(t)$ is the number of active appliances at time t, i.e., the ones that are consuming electricity, and r denotes the consumption rate in kW per appliance. We assume that the idle appliances "connect", i.e., start to consume electricity, according to a Poisson process with rate $\lambda \in [0, \lambda^M]$, where λ^M is the parameter that denotes the maximum connection rate. The SBO controls the connection rate λ by broadcasting a control signal $u \in [0, U^M]$ to the appliances, which we will call "price", where U^M is the maximum price that can be broadcasted. The price u is updated just after the RS signal is updated, and it stays constant in between. Suppose the price at time t is $u(t)$; then the connection rate is given by $\lambda(t) = \lambda^M \left(1 - u(t)/U^M\right)$. The apliance that has connected stays active for an exponentially distributed amount of time with rate μ. Moreover, we assume that the number of active appliances satisfies $n^m \leq n(t) \leq n^M$. As a result, $n(t)$ corresponds to the length of a truncated $M/M/\infty$ queue.

The dynamics of the RS signal y are independent of the actions of SBO and modeled as follows: We assume that y can only take discrete values in its range such that $y(t) \in \{-1, -1 + \Delta\bar{y}, \ldots, 0, \ldots, 1 - \Delta\bar{y}, 1\}$, where $\Delta\bar{y}$ is the discretization constant and $\Delta\bar{y} \geq 0$. The update of the RS signal is similar to a random walk, i.e., $y(t+\Delta t) = y(t)+\Delta y$, where Δy is a random variable and $\Delta y \in \{-\Delta\bar{y}, 0, \Delta\bar{y}\}$. We also say that y has a direction "up" or "down" at time t, which is denoted by $d(t) \in \{+1, -1\}$, respectively. The probability distribution of the random variable Δy is a function of $y(t)$ and $d(t)$.

When the SBO observes the RS signal update at time t, it needs to modulate the consumption in the building in Δt seconds, in such a way that the consumption rate becomes

$A + Ry(t)$ at time $t + \Delta t$. Otherwise, the following tracking cost is assessed

$$\left[(n(t+\Delta t))\, r - (A + Ry(t))\right]^2 \kappa \Delta t \tag{1}$$

where $\kappa \geq 0$ is the cost coefficient. We also say that an appliance that connects at time t realizes a utility $\phi(t)$ that is seen as a contribution to the social welfare. $\phi(t)$ is a random variable and assumed to be uniformly distributed in $[u(t), U^M]$. Then, the rate of utility at time t is given by

$$\left(1 - u(t)/U^M\right)\left(u(t) + U^M\right)/2. \tag{2}$$

The problem of which a concise summary is provided here is modeled as an Average Cost Infinite Horizon Dynamic Programming problem with the following Bellman Equation.

$$\begin{aligned} &h(n,y,d) + \bar{J} = \\ &\min_{u \in [0,U^M]} \left\{ \mathop{\mathbb{E}}_{\Delta n, \Delta y | u,n,y,d} \Big[\Delta t \kappa \left((n+\Delta n)r - (A+Ry)\right)^2 \right. \\ &- \Delta t \lambda^M \left(1 - u/U^M\right)\left(u + U^M\right)/2 \\ &\left. + h(n+\Delta n, y+\Delta y, d') \Big] \right\} \end{aligned} \tag{3}$$

where d' is the new direction, Δn is a random variable that is a function of the price $u(t)$, and represents the change $n(t+\Delta t) - n(t)$ that is the number of connections minus the number of disconnections in $[t, t+\Delta t]$. In the Bellman equation, $\bar{J}$ represents the cost per Δt time interval, and $h(n,y,d)$ is the differential cost function, which can be interpreted as the "disadvantage" of being in the state (n,y,d).

III. IMPORTANT PROPERTIES OF THE OPTIMAL POLICY

This section proves properties of both the Differential Cost Function and the associated optimal control policy for the purpose of understanding the problem's structure and receiving intuition and guidance in the formulation of tractable near-optimal solution approaches. In [7], numerical results suggested the existence of two important properties: *(i)* The differential cost function appeared to be convex in the number of active appliances n for fixed values of the RS signal variables y and d, and *(ii)* the optimal control policy appeared to be non-decreasing in n for fixed y and d. Before we attempt to implement the approximate DP algorithms presented in Section IV and V, we prove these properties formally. The proof is derived for a simplified version of the problem with more streamlined transition probabilities.

A. Convexity of Differential Cost Functions in a Simplified Problem

Consider the problem defined in Section II with the following modifications: The RS signal $y \in [-1, 1]$ is fixed, so the SBO is obliged to keep the consumption at $A + Ry$ kW all the time. Moreover, since the arrival rate is a linear function of the price, we change the control from price to the arrival rate $\lambda \in [0, \lambda^M]$. In addition, the control is updated every $\Delta\tau$ time unit, which is very short so that at most one event (connection/disconnection) can take place. Particularly, $\Delta\tau = 1/\nu$ and $\nu = n^M\mu + \lambda^M$. Without loss of generality, we assume that $\nu = 1$. The cost function is defined as a convex function $g(n)$. This simplified problem can be again formulated as an Average Cost Infinite Horizon DP problem with the following Bellman Equation:

$$\begin{aligned} h(n) + \bar{J} &= g(n) + n\mu h(n-1) + (n^M - n)\mu h(n) \\ &\quad + \min_{\lambda \in [0,\lambda^M]} \{\lambda h(n+1) + (\lambda^M - \lambda) h(n)\} \end{aligned} \tag{4}$$

Note that this problem, and therefore the following propositions, correspond to the case of a truncated $M/M/\infty$ queue where the goal is to keep the queue length constant at a desired level. Before providing our propositions, we define the difference operator D as follows:

$$Dv(n) \coloneqq v(n+1) - v(n) \tag{5}$$

for any function v. Then, the second difference operator is defined as $D^{(2)}v(n) \coloneqq D\left(Dv(n)\right) = v(n+2) + v(n) - 2v(n+1)$.

Lemma 1. At least one of the two extremes of the control range $[0, \lambda^M]$ is an optimal solution to the Bellman equation for the simple problem defined in Equation (4).

Proof: If we rewrite the expression where the optimization takes place in (4) as follows:

$$\begin{aligned} &\min_{\lambda \in [0,\lambda^M]} \{\lambda h(n+1) + (\lambda^M - \lambda) h(n)\} \\ &\qquad = \lambda^M h(n) + \min_{\lambda \in [0,\lambda^M]} \{\lambda D h(n)\} \end{aligned} \tag{6}$$

then it is clear that the optimal solution $\lambda^* = 0$ when $Dh(n) > 0$ and $\lambda^* = \lambda^M$ when $Dh(n) < 0$. For the cases where $Dh(n) = 0$, both λ^M and 0 are optimal solutions. ■

From now on, we replace the expression $\min_{\lambda \in [0,\lambda^M]} \{\lambda h(n+1) + (\lambda^M - \lambda) h(n)\}$ in Equation (4) by $\lambda^M min\{h(n+1), h(n)\}$.

Proposition 1. The differential cost function $h(n)$ that satisfies the Bellman equation of the simple problem defined in Equation (4) is convex if the cost function $g(n)$ is convex and bounded.

Proof: In [15], Porteus shows that if the structure that is identified for some value function v is preserved under the optimal operator T, then the same structure can be established for the value function h of the corresponding Bellman equation. This approach is also used in [16] to show certain properties of some other queuing applications. Now, let V be the set of functions such that if $v \in V$, then $D^{(2)}v \geq 0$, i.e., convex. Therefore, we need to show that $D^{(2)}Tv(n) > 0$ for $v \in V$, where the T operator is defined according to Equation (4) as

$$\begin{aligned} Tv(n) &= g(n) + n\mu v(n-1) + (n^M - n)\mu v(n) \\ &\quad + \lambda^M min\{v(n+1), v(n)\}. \end{aligned} \tag{7}$$

Now, define $\bar{n} = n^M - n$. Then,

$$\begin{aligned} D^{(2)}Tv(n) &= Tv(n+2) + Tv(n) - 2Tv(n+1) \\ &= [g(n+2) + g(n) - 2g(n+1)] \\ &\quad +n\mu\,[v(n+1) + v(n-1) - 2v(n)] \\ &\quad +\bar{n}\mu\,[v(n+2) + v(n) - 2v(n+1)] \\ &\quad +\lambda^M\big[min\{v(n+3), v(n+2)\} \\ &\qquad +min\{v(n+1), v(n)\} \\ &\qquad -2min\{v(n+2), v(n+1)\}\big] \\ &\geq 0 \end{aligned} \tag{8}$$

where the first equality is due to the definition of the second difference operator, and the second equality is obtained using Equation (7). Given that $g(n)$ and v are convex, the related parts become nonnegative. For the expression with the minimization operators, four of the eight possible outcomes (where $v(n+2) > v(n+1)$, $v(n+2) > v(n+3)$; or $v(n+1) > v(n)$, $v(n+1) > v(n+2)$) are inconsitent with the convexity of v, hence will not occur. In the other two possibilities, the expression in square brackets is reduced to $v(n+1) - v(n+2)$, when $v(n+1) \geq v(n+2)$ is assumed; and to $v(n+2) - v(n+1)$, when $v(n+1) \leq v(n+2)$ is assumed. For the remaining two possibilities, non-negativity is guaranteed through convexity of v. Moreover, $g(n)$ is bounded and the limit of every Cauchy sequence of functions in V will be again in V. Hence, "Assumption IH" of [15] is satisfied and the result follows. ■

B. Monotonicity of the Optimal Policies

Next, we generalize the simplified model used in Section III-A by making $\Delta\tau$ sufficiently large so that many arrivals and departures are possible between control updates. Again, the RS signal variables, y and d, are fixed. On the other hand, we use the price to represent the control. Moreover, we make additional assumptions and use an approximation to state the next proposition. Given these modifications, the problem can be described by the following Bellman Equation:

$$h(n) + \bar{J} = g(n) + \min_{u\in[0,u^M]} \left\{ \mathop{\mathbb{E}}_{\Delta n|u,n} [h(n+\Delta n)] \right\} \tag{9}$$

Proposition 2. Assume that $n(t+\Delta t)$ is normally distributed with mean

$$M_{n(t+\Delta\tau)} = \frac{\lambda}{\mu}\left(1 - e^{-\mu\Delta\tau}\right) + ne^{-\mu\Delta\tau} \tag{10}$$

and variance

$$\begin{aligned} \sigma^2_{n(t+\Delta\tau)} &= ne^{-\mu\Delta\tau} - (ne^{-\mu\Delta\tau} - \lambda/\mu(e^{-\mu\Delta\tau} - 1))^2 \\ &\quad +n(n-1)e^{-2\mu\Delta\tau} + \lambda^2(e^{-\mu\Delta\tau} - 1)^2/\mu^2 \\ &\quad -\lambda(e^{-\mu\Delta\tau} - 1)/\mu \\ &\quad -2\lambda n(e^{-\mu\Delta\tau} - 1)e^{-\mu\Delta\tau}/\mu \end{aligned} \tag{11}$$

given $n := n(t)$ and $\lambda = \lambda^M(1 - u/U^M)$. Also assume that the differential cost function of the problem defined in Equation (9), $h(n)$, is convex and symmetric around its minimum. Then the optimal price u is nondecreasing in n.

Proof: As the normal distribution is symmetric around its mean, the optimal price $u \in [0, U^M]$ would be the one that gives $M_{n(t+\Delta\tau)} = n^*$, where $n^* = \arg\min h(n)$, or the closest boundary value of the control space when that is not feasible. Clearly, for $n' > n$, the mean shifts to the right. The shift needs to be balanced by a $\lambda' \leq \lambda$, which is equivalent to have a new optimal price that satisfies $u' \geq u$. ■

In this proof, the convexity assumption is reasonable given that it provenly holds for the simpler problem considered in Proposition 1. The symmetry of $h(n)$ is only needed to prevent pathological cases that may be caused by non-monotonic behavior that the variance of the normal distribution exhibits. In practice, the nondecreasing behavior of the optimal price is very unlikely to be violated. Nevertheless, symmetry of $h(n)$ is still a reasonable assumption, especially when a quadratic cost function is used and $n \gg 0$ where the departure rates do not differ significantly across different values of n. Moreover, the normal distribution is a reasonable assumption for $n(t+\Delta t)$ as we have shown in [17].

C. Sigmoid Structure of the Optimal Control Policy

The numerical results that we obtained in [7] exhibit a sigmoid function relationship of the optimal price to the tracking error, $nr - (A + Ry)$, for constant values of the regulation signal y. We also observed that the sigmoid shifts along the axis of n for different values of y and d, which means that y and d can also enter linearly in the argument of the sigmoid function and provide a good approximation of the optimal policy. Specifically, in [7], $u \approx U^M / \left(1 + e^{\theta_1(nr-(A+Ry))+\theta_2 y+\theta_3 d}\right)$ was used to describe the optimal policy. In Equation (19) of Section VI, this approximation will be updated based on the discussions provided in Section III-A and III-B.

IV. BOUNDED LSTD ACTOR-CRITIC ALGORITHM

As already noted, solving the Dynamic Programming problem defined in Section II by discretizing the action space and applying the LP approach becomes intractable for large problems. Moreover, the reasonable functional approximation of the optimal policy described above enables solution approaches that do not require action space discretization. We proceed to formulate an actor-critic algorithm adapted to the case where the action space is continuous and bounded. Actor-critic algorithms optimize the parameters of a Randomized Stationary Policy (RSP), denoted by the vector $\theta \in \mathbb{R}^m$, along a sample path $\{\mathbf{x}_k, u_k\}$, where $\mathbf{x}_k \in \mathbb{X}$ represents the state and $u_k \in \mathbb{U}$ represents the control for the k^{th} step ([9], [10]). The RSP is denoted by $p_\theta(u|\mathbf{x})$, and is a mapping that assigns a probability distribution to each state x over the action space $\mathbb{U}$. When the action space is continuous, $p_\theta(u|\mathbf{x})$ corresponds to a probability density function (p.d.f.). References [9] and [10] contain all of the technical conditions. We note, however, that a key condition is that $p_\theta(u|\mathbf{x})$ must be twice differentiable with respect to θ.

The critic uses a linearly parameterized approximation of the Q function, defined as

$$Q_\theta(\mathbf{x}, u) = \mathbb{E}\left[g_\theta(\mathbf{x}, u) + Q_\theta(\mathbf{x}', u')\right] - J_\theta \qquad (12)$$

where g_θ is the cost function, J_θ is the average cost, $\mathbf{x}'$ is the new state given the current state $\mathbf{x}$ and the current control u, and u' is the new control that will be generated by the RSP $p_\theta(u|\mathbf{x}')$. In [9], Konda et al. use the following linear approximation structure for the Q-value function

$$Q^s_\theta(\mathbf{x}, u) = \psi'_\theta(\mathbf{x}, u)\mathbf{s} \qquad (13)$$

where $\psi_\theta(\mathbf{x}, u)$ is the m-dimensional feature vector given $(\mathbf{x}, u)$ and $\mathbf{s} \in \mathbb{R}^m$ is the vector of the approximation parameters. Moreover, the feature vector is selected such that

$$\psi^i_\theta(\mathbf{x}, u) = \frac{\partial}{\partial\theta_i} \ln p_\theta(u|\mathbf{x}), \qquad i = 1, \ldots, m \qquad (14)$$

and then the policy gradient at k^{th} step is given by $-\psi'_{\theta_k}(\mathbf{x}_{k+1}, u_{k+1})\mathbf{s}_k\psi_{\theta_k}(\mathbf{x}_{k+1}, u_{k+1})$.

In [10], Estanjini et al. show that the actor-critic algorithm's performance is superior when a Least Squares Temporal Differences (LSTD) approach, rather than a temporal differences (TD) algorithm, is used by the critic to learn $\mathbf{s}$. We adopt the same approach and note that the details of the LSTD algorithm can be found in [8] and [10].

As mentioned earlier, the main challenge with using an actor-critic algorithm in our context is that the action space is continuous and bounded, for which we do not have a twice differentiable p.d.f. to use as RSP. One of the approaches would be to use the Gaussian distribution with a parameterized mean for the actor's choice as in [11] and [12], and peg the action to the boundary when the RSP produces an action outside the action space. However, in our problem's context, the learning algorithm easily fails when the mean of the distribution moves out of the action space. Kimura et al. [13] propose the remedy of rejecting actions until the RSP produces one that is in the action space. The disadvantages of this remedy are significant, however, as mentioned already in Section I. This paper proposes an alternative remedy that eliminates these disadvantages.

In Section II, the action space is defined as $u \in \mathbb{U} = [U^m, U^M]$, where $U^m = 0$. Moreover, in Section III-C, we argued relying on previous numerical experience that the optimal control policy can be approximated well by a sigmoid function. Using the fact that a sigmoid function is a mapping such that $\mathbb{R} \mapsto (0, 1)$, we consider first a related control $u^r \in (-\infty, +\infty)$ which allows us to use a Gaussian distribution with a parameterized mean M_θ and a fixed variance σ^2 to generate u^r. We then simulate the next state as required by the actor-critic algorithm by converting $u^r \in (-\infty, +\infty)$ to $u \in (0, 1)$ through a sigmoid function. It is important to emphasize that the control u never appears in the algorithm explicitly, since the actor-critic observes the next state after a simulation step that is a black-box to it. The control is transformed in this simulation black-box whose input is the current state and the control u^r. We also note that although the sigmoid function turns out to be an excellent approximation of the optimal control policy, it has a good potential to work well in any problem with a bounded action space as long as the p.d.f selection and the corresponding mean parameterization are carried out appropriately.

Even with a sigmoid function transformation, the learning process may fail when the mean of the Gaussian p.d.f., M_θ, gets away from the neighborhood of 0. This is because the RSP will persistently produce a control u^r that will lead to a control u very close to U^m or U^M, when M_θ is very negative or positive, respectively. To prevent this from happening, we impose bounds on θ in the actor, i.e. $\theta^\ell_i \le \theta_i \le \theta^u_i$, $i = 1, \ldots, m$. This is equivalent to choosing the step-size in the actor step to keep the parameters in the desired interval. Unlike the case described in [13], this does not increase the algorithm's computational complexity. Following the overview and motivation of this section, the next section proceeds with the detailed description of the Bounded LSTD Actor-Critic Algorithm.

Bounded LSTD Actor-Critic Algorithm

Following [10], the parameters for the k^{th} step of the algorithm are defined next. $\hat{J}_k$ stands for the estimate of the average cost, $\mathbf{s}_k \in \mathbb{R}^m$ is the iterate for the parameter set introduced in (13), $\mathbf{z}_k \in \mathbb{R}^m$ is called Sutton's eligibility trace ([14]), $\mathbf{b}_k \in \mathbb{R}^m$ is the eligibility trace scaled by the difference between the observed cost and the average cost estimate, $\mathbf{B}_k \in \mathbb{R}^{m\times m}$ is the sample estimate of the matrix formed by $z_k\left(\psi_{\theta_k}(\mathbf{x}_{k+1}, u^r_{k+1}) - \psi_{\theta_k}(\mathbf{x}_k, u^r_k)\right)$.

Initialization: All of the entries in $\hat{J}_0$,$\mathbf{s}_0$, $\mathbf{z}_0$,$\mathbf{b}_0$ and $\mathbf{B}_0$ are set to zero, and θ_0 takes some initial value.

Critic:

$$\hat{J}_{k+1} = \hat{J}_k + \gamma_k\left[g(\mathbf{x}_k, u^r_k) - \hat{J}_k\right],$$
$$\mathbf{z}_{k+1} = \Lambda\mathbf{z}_k + \psi_{\theta_k}(\mathbf{x}_k, u^r_k),$$
$$\mathbf{b}_{k+1} = \mathbf{b}_k + \gamma_k\left[\left(g(\mathbf{x}_k, u^r_k) - \hat{J}_k\right) - \mathbf{b}_k\right],$$
$$\mathbf{B}_{k+1} = \mathbf{B}_k + \gamma_k\left[z_k\left(\psi_{\theta_k}(\mathbf{x}_{k+1}, u^r_{k+1}) - \psi_{\theta_k}(\mathbf{x}_k, u^r_k)\right) - \mathbf{B}_k\right],$$
$$\mathbf{s}_{k+1} = -\mathbf{B}^{-1}_k\mathbf{b}_k,$$

where $\Lambda \in [0, 1)$ and $\gamma_k := 1/k$. In addition, we use the pseudo inverse of $\mathbf{B}$ when it is singular.

Actor:

$$\theta^r_{k+1} = \theta_k - \beta_k\Gamma(\mathbf{s}_k)\psi_{\theta_k}(\mathbf{x}_{k+1}, u^r_{k+1})\mathbf{s}_k\psi_{\theta_k}(\mathbf{x}_{k+1}, u^r_{k+1}),$$
$$\theta_{k+1} = med\left\{\theta^\ell, \theta^r_{k+1}, \theta^u\right\},$$

where med is the element-wise median operator and $\left(\theta^\ell\right)_i < \left(\theta^u\right)_i$ for $i = 1, \ldots, m$. Moreover, $\{\beta_k\}$ is a deterministic and non-increasing sequence that satisfies $\sum_k \beta_k = \infty$, $\sum_k \beta^2_k < \infty$ and $\lim_{k\to\infty} \beta_k/\gamma_k = 0$. An example would be $\beta_k = c/(k \ln k)$, for $k > 1$, where $c > 0$ is a constant parameter. The parameter $\Gamma(\mathbf{s}_k)$ is used to keep the actor updates bounded and it has the following form

$$\Gamma(\mathbf{s}_k) = \begin{cases} \frac{D}{||\mathbf{s}||}, & \text{if } ||\mathbf{s}|| > D, \\ 1, & \text{otherwise .} \end{cases}$$

Simulation of action and the next state: Suppose that the action space is given as $u \in \mathbb{U} = [U^m, U^M]$. Given the state $\mathbf{x}_k$, the control u_k^r is generated by using the p.d.f. $p_{\theta_k}(u_k^r|\mathbf{x}_k)$, i.e., $u_k^r \sim N(M_\theta, \sigma^2)$. Then, u_k is obtained as

$$u_k = U^m + \frac{1}{1+e^{-u_k^r}}(U^M - U^m),$$

and then employed to simulate the next state $\mathbf{x}_{k+1}$ using the problem-specific state transition dynamics.

V. A NOVEL APPROXIMATE POLICY ITERATION ALGORITHM

Actor-critic algorithms are especially advantageous when the state space is very large and/or there is little knowledge about the dynamics of the system. However, in our case, the state transition dynamics are known and we would like to make the problem more scalable by using a parameterized function that approximates the optimal control policy given the state of the system, rather than discretizing the action space that makes large problems intractable. In order to optimize the policy function parameter vector, denoted by $\theta \in \mathbb{R}^m$, we develop an Approximate Policy Iteration (API) algorithm, which iteratively *(i)* estimates the average and differential costs of the Bellman Equation in (3) under policy θ, $\bar{J}_\theta$ and $h_\theta(x)$, $\forall x \in X$, *(ii)* improves the policy θ. Although analogous to actor-critic and standard policy iteration schemes, the API is designed to overcome the challenges specific to the optimization of parameterized policies when the system dynamics are known.

Step *(i)*: Policy Evaluation step. A tentative policy θ (similar to the role of the critic of B-LSTD AC) is associated to the corresponding average cost and differential cost vector by solving a linear system. Denoting the discrete state space by X and its size by $|X|$, this is equivalent to solving the following LP with $|X|+1$ decision variables and $|X|$ constraints:

$$\begin{aligned} \max_{\bar{J}_{\theta_k}, h_{\theta_k}(x)\ \forall x \in X} \quad & \bar{J}_{\theta_k} \\ \text{s.t. } \bar{J}_{\theta_k} + h_{\theta_k}(x) \leq & \mathop{\mathbb{E}}_{y \in X}[g(x,y,\theta_k)] \\ & + \sum_{y \in X} P_{xy}(\theta_k) h_{\theta_k}(y) \qquad \forall x \in X \end{aligned}$$

where k denotes the iteration, g is the one step cost function of the DP problem as given in (3) and P_{xy} is the probability of transitioning from state x to y. Note that the LP solution to a DP with discrete states and actions reduces here to a problem with a single action for each state. The advantage of translating the linear system to an LP is that the dual variables of the LP, $\pi_{\theta_k}(x)$, $\forall x \in X$, give the proportion of the time that the system spends in state x (or equivalently the steady state probability) when controlled by the policy θ_k ([8]). These probabilities provide the foundation of the algorithm's next step.

Step *(ii)*: The tentative policy used in the previous step is improved. This is equivalent to the actor of the B-LSTD AC algorithm. However, unlike the actor-critic algorithm, convergence of the API algorithm requires the average cost to decrease in each iteration, i.e., $\bar{J}_{\theta_{k+1}} \leq \bar{J}_{\theta_k}$. The improvement in the cost is expressed as follows ([8]):

$$\bar{J}_{\theta_k} - \bar{J}_{\theta_{k+1}} = \sum_{x \in X} \pi_{\theta_{k+1}}(x) \delta_{(\theta_k, \theta_{k+1})}(x) \tag{15}$$

where we define

$$\begin{aligned} &\delta_{(\theta_k,\theta_{k+1})}(x) = \\ &\left[\mathop{\mathbb{E}}_{y \in X}[g(x,y,\theta_k)] + \sum_{y \in X} P_{xy}(\theta_k) h_{\theta_k}(y) \right] \\ &- \left[\mathop{\mathbb{E}}_{y \in X}[g(x,y,\theta_{k+1})] + \sum_{y \in X} P_{xy}(\theta_{k+1}) h_{\theta_k}(y) \right]. \end{aligned} \tag{16}$$

In vector notation, it is required that $\pi'_{\theta_{k+1}} \delta_{(\theta_k,\theta_{k+1})} \geq 0$.

In the standard policy iteration algorithm, non-negativity of $\bar{J}_{\theta_k} - \bar{J}_{\theta_{k+1}}$ is trivially achieved as the policy is improved for each state x, so that $\delta_{(\theta_k,\theta_{k+1})}(x) \geq 0, \forall x \in X$. In API, however, changing θ of the parameterized policy may result in negative values of $\delta_{(\theta_k,\theta_{k+1})}(x)$ as well as positive ones for different x. Therefore, we seek θ values that will make the whole summation in (15) positive. Namely, we aim to solve

$$\theta_{k+1} = \arg\max \pi'_{\theta_{k+1}} \delta_{(\theta_k,\theta_{k+1})}. \tag{17}$$

The issue with the optimization problem in (17) is that there is generally no closed form expression of $\pi_{\theta_{k+1}}$ as a function of θ_{k+1}, or in fact, it is an intractable highly nonlinear complex relationship. Therefore, we propose to approximate $\pi_{\theta_{k+1}}$ by π_{θ_k} that we have readily available from Step *(i)*. However, we need θ_{k+1} to be close to θ_k for the approximation to work, which leads us to solve the following optimization to obtain the new policy:

$$\max_{\theta_{k+1} \in [\theta_k - \Delta\theta, \theta_k + \Delta\theta]} \pi'_{\theta_k} \delta_{(\theta_k,\theta_{k+1})} \tag{18}$$

where $\Delta\theta \in \mathbb{R}^m$ with all non-negative entries. Note that this optimization involves only $h_{\theta_k}(x)$ and not $h_{\theta_{k+1}}(x)$, which is also provided by the LP in Step *(i)*.

In the iterative solution of (18), the $\Delta\theta$ vector is updated adaptively as follows: When the condition $\bar{J}_{\theta_{k+1}} \leq \bar{J}_{\theta_{k+1}}$ is violated, $\Delta\theta$ is multiplied by ρ, $0 < \rho < 1$ and the policy improvement step repeated, otherwise $\Delta\theta$ is multiplied by $1/\rho$. Numerical experience shows that, most of the time, $\Delta\theta$ does not have to be very small for the non-negativity condition to be satisfied and the API algorithm to converge.

VI. USE OF ANALYTIC FUNCTIONAL REPRESENTATIONS IN B-LSTD ACTOR-CRITIC AND API ALGORITHMS

As described so far, the B-LSTD AC and API algorithms iteratively optimize the parameters of an analytic function, f_θ shown in Equation (19), which, however, is employed in similar but different ways by the two algorithms to determine the value of the price control for a given point in the state space: In the API algorithm, for a given set of parameter

values, θ, the price control is given as the output of a sigmoid transformation of f_θ. In the B-LSTD AC, f_θ represents the mean of the Gaussian distribution shown in Equation (21); a random sample is generated from this distribution and the same sigmoid transformation of this sample, shown in Equation (20), provides the randomized price control. Noting in addition that f_θ is also used to approximate the Q-value function in B-LSTD AC described in Equations (13) and (14), it follows that f_θ should be consistent with the structural properties of the Q-value function, which are analogous to those of the differential cost function. Therefore, f_θ should be consistent to the properties of the optimal policy, as described in *(i)* Section III-C, namely be a function of the tracking error, *(ii)* Section III-B, namely be monotonic in n, and *(iii)* Proposition 1, which indicates convexity of the differential cost function w.r.t. n for fixed y. Interpreting f_θ as a linear function of features of the system state, we are then justified in defining $nr-(A+Ry)$ as the first feature of f_θ. More importantly, we can restrict the corresponding parameter, θ_1, to take only positive values by using the monotonicity property, which turns out to be extremely helpful for the performance of the B-LSTD AC algorithm. Other reasonable feature selections are y and d, while convexity suggests the last feature as $(nr-(A+Ry))^2$.

Thus, we define f_θ as

$$f_\theta = \theta_1 (nr-(A+Ry)) + \theta_2 y + \theta_3 d + \theta_4 (nr-(A+Ry))^2. \quad (19)$$

The price is given in the B-LSTD AC algorithm by the sigmoid transformation of u^r

$$u = U^m + \frac{1}{1+e^{-u^r}}(U^M - U^m), \quad (20)$$

where u^r is a random sample drawn from the normal distribution

$$u^r \sim N(f_\theta, \sigma^2). \quad (21)$$

Finally, in the API algorithm, the price is obtained as the sigmoid transformation of f_θ:

$$u = U^m + \frac{1}{1+e^{-f_\theta}}(U^M - U^m). \quad (22)$$

VII. NUMERICAL EXPERIENCE

In this section, unless stated otherwise, we use the following problem input: The average consumption rate is $A = 100$ kW, the amount RS provision is $R = 30$ kW, the maximum appliance connection rate is $\lambda^M = 150$/min, the appliance departure rate is $\mu = 1$/min, RS signal discretization constant is $\Delta\bar{y} = 1/30$, RS signal update interval is $\Delta t = 4/60$ min, consumption rate per appliance is $r = 1$ kW, the allowed change in the number of active appliances in Δt min is $-10 \leq \Delta n \leq 10$ and the allowed range of the number of the active appliances is $50 \leq n \leq 150$. The cost function includes only the tracking error since we decided to neglect the utility component for simplicity, which is in fact more compatible with the propositions introduce in this paper, and more appropriate for comparing the performance across algorithms since the selection of the feature vectors is more streamlined. Therefore, $g = (n(t+\Delta t)r - (A+Ry(t)))^2$.

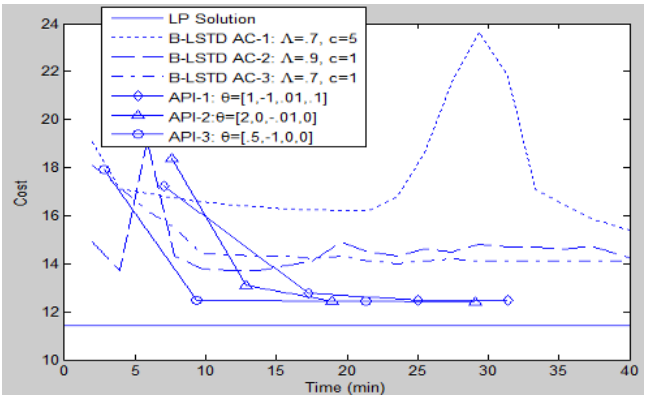

Fig. 1: Numerical Comparison of B-LSTD AC and API Algorithms

Figure 1 compares the performance of the two algorithms, where both are tested for different initial conditions and choice of algorithm step size. We note two important points regarding this comparison: *(i)* The B-LSTD AC algorithm is run several times with exactly the same parameter and step size levels but with different simulation seeds. Each trajectory of this algorithm in Figure 1 corresponds to the run that exhibited the best performances with that parameter/bound level. *(ii)* In order to make a fair comparison, we did not plot the average cost estimate of the B-LSTD AC algorithm, $\bar{J}_k$, which carries some portion of the high costs due to previous bad policies. Instead, with certain intervals, we used the tentative policy θ_k that the algorithm selects in the k^{th} iteration and calculated the associated average cost function by solving a linear system.

Under these circumstances, the numerical experience revealed two main results. *(i)* After many experiments, the adaptation that we have made on the existing actor-critic algorithms for the bounded and continuous action spaces, that is our proposed B-LSTD AC algorithm, has successfully optimized the policy parameters and returned good objective values in the vast majority of the cases. In very few cases, the mean of the RSP diverged to very positive/negative values. However, one may find better bounds on θ and make these failures less likely. In our experiments, we intentionally used looser bounds for some θ_i to test the algorithm. *(ii)* In all of the cases, the API algorithm has converged to a better objective value than the B-LSTD AC algorithm. Moreover, API has deviated by less than 9% from the objective value that we were able to obtain solving a computationally much more demanding discretized action space DP. Moreover, given that API optimizes the parameters of an analytic functional representation of the optimal policy, we can say with confidence supported also by current work that API's sub-optimality gap may decrease further with better choice of the functional form of the parameterized analytic policy representation. We close by comparing the two algorithms on

the basis of structural and information available to the SBO rather than numerical performance differences. First, we note that the two algorithms have different working principles. Actor-critic algorithms evaluate and optimize the policy parameters along a simulation sample path with the actor being unaware of the exact system dynamics, whereas API is more of an exact method in terms of evaluating and optimizing the control policy. Both approaches have their own advantages and drawbacks. B-LSTD AC algorithm is superior over API in the following areas: *(i)* B-LSTD AC algorithm is still able to return "good" policies even when the state space is huge and/or the θ parameter vector consists of many entries due to a large number of features. *(ii)* When the state transition dynamics are not exactly known or when the transition probabilities are hard to calculate, API algorithm cannot be used while B-LSTD AC is still applicable. On the other hand, API algorithm has the following important advantages over the actor-critic type of algorithms in general: *(i)* When the state space has a size that can be handled by an LP, and the number of features is compatible with the optimization problem in the policy improvement step, API will most likely return a better policy, which was always the case in our experiments. Moreover, the computation time is comparable or better than in the actor-critic algorithm. Given that API solves an LP that is equivalent to solving a linear system of equations with one degree of freedom, and the nonlinear optimization is applied on a limited set of parameters, problem size is generally not very restrictive. *(ii)* The B-LSTD AC algorithm suffers from poor convergence when the parameter range is not bounded well around the optimal, a condition that requires some prior intuition. For example, restricting θ_1 to the positive real numbers using the monotonicity property proven in this paper improved the performance dramatically. On the other hand, the parameter bounds are updated in the API algorithm with progressing iterations. *(iii)* The actor-critic algorithms are very sensitive to selection of initial parameters and one may need to go through many trials until a good choice is found. For example, the algorithm may easily converge to a local optimum that is far from the global optimum, if the step size is not chosen appropriately. In contrast, API updates the step size based on improvements in the cost function. Therefore, convergence is less sensitive to parameter initialization. In all of our experiments, API has converged to the same point regardless of the initial parameter choice. *(iv)* Even if one finds a parameter set that is known to have performed well in a previous run of the actor-critic algorithm, there is no guarantee that the same results will be obtained in a subsequent run, for the same problem and with the same parameters. This is because the algorithm is based on Monte Carlo simulation and may thus follow different sample paths in each run, whereas the API algorithm relies on a deterministic solution process.

VIII. CONCLUSION

This paper advanced the state of the art in decision support algorithms that allow Smart Building Operators to provide almost real time Regulation Service to the control center responsible for the operation of the Balancing Area where an advanced building is located. The associated stochastic DP problem was investigated in terms of the structural properties of its optimal solution and tractable algorithms were developed by exploiting these properties. The two algorithms represent improvements of existing Actor Critic and Policy Iteration algorithms and can address a wide range of situations ranging over perfect to imperfect knowledge of the system dynamics. Numerical experience with both algorithms is reported indicating that they are both tractable. Although the actor critic algorithm is slower when the system dynamics are known, it is still applicable when they are not. Our future research agenda includes to implement both algorithms in cases where there are multiple appliance types and the control is multi-dimensional.

REFERENCES

[1] *Manual 2: Ancillary Services Manual*, v. 3.26, NYISO, Rensselaer, NY, 2013.

[2] Y. V. Makarov, C. Loutan, J. Ma, and P. de Mello, "Operational impacts of wind generation on California power systems", *IEEE Trans. Power Syst*, vol. 24, no. 2, pp. 10391050, May 2009.

[3] G. Gowrisankaran, S. Reynolds, M. Samano, "Intermittency and the Value of Renewable Energy", *NBER Working Paper Series*, Cambrigde, MA, May 2011.

[4] I.C. Paschalidis, B. Li, M. C. Caramanis, "A Market-Based Mechanism for Providing Demand-Side Regulation Service Reserves", in *50th IEEE Conference on Decision and Control*, Orlando, FL, Dec. 2011, pp. 21-26.

[5] I.C. Paschalidis, B. Li, M. C. Caramanis, "Demand-Side Management for Regulation Service Provisioning through Internal Pricing", *IEEE Trans. Power Syst.*, vol. 27, no. 3, pp. 1531-1539, Aug. 2012.

[6] B. Zhang, J. Baillieul, "A Packetized Direct Load Control Mechanism for Demand Side Management", in *51st IEEE Conference on Decision and Control*, Maui, HI, Dec. 2012, pp. 3658-3665.

[7] M. C. Caramanis, I.C. Paschalidis, C.G. Cassandras, E. Bilgin, E. Ntakou, "Provision of Regulation Service Reserves by Flexible Distributed Loads", in *51st IEEE Conference on Decision and Control*, Maui, HI, Dec. 2012, pp. 3694-3700.

[8] D.P. Bertsekas, *Dynamic Programming and Optimal Control*. Belmont, MA: Athena Scientific, 1995, vol. II.

[9] V. Konda and J. Tsitsiklis, "On Actor-Critic Algorithms", *SIAM Journal on Control and Optimization*, vol. 42, no. 4, pp. 1143-1166, Aug. 2003.

[10] R. M. Estanjini, K. Li, I. Ch. Paschalidis, "A Least Squares Temporal Difference Actor-Critic Algorithm with Applications to Warehouse Management", *Naval Research Logistics*, vol. 59, pp. 197-211, Mar. 2012.

[11] R. J. Williams, "Simple Statistical Gradient Following Algorithms for Connectionist Reinforcement Learning", *Machine Learning*, vol. 8, pp. 229-256, May 1992.

[12] K. Doya, "Efficient Nonlinear Control with Actor-Tutor Architecture, *Advances in Neural Information Processing Systems*, vol. 9, pp. 1012-1018, 1997.

[13] H. Kimura, T. Yamashita, S. Kobayashi, "Reinforcement Learning of Walking Behavior for a Four-Legged Robot",in *40th IEEE Conference on Decision and Control*, Orlando, FL, Dec. 2001, pp. 411-416.

[14] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.

[15] E. Porteus, "Conditions for Characterizing the Structure of Optimal Strategies in Infinite-Horizon Dynamic Programs", *J. of Optimization Theory and Applications*, vol. 36, no. 3, pp. 419-432, Mar. 1982.

[16] A.Y. Ha, "Optimal Dynamic Scheduling Policy for a Make-To-Stock Production System", *Operations Research*, vol. 45, no. 1, pp. 42-53, Jan. 1997.

[17] E. Bilgin, M. C. Caramanis, "Decision Support for Offering Load-Side Regulation Service Reserves in Competitive Power Markets", in *52nd IEEE Conference on Decision and Control*, Florence, Italy, Dec. 2013.