An Actor-Critic approach for control of Residential Photovoltaic-Battery Systems

Amit Joshi [*] **Massimo Tipaldi** [*] **Luigi Glielmo** [*]

[*] *GRACE Lab, Department of Engineering, University of Sannio,
Piazza Roma 21, Benevento 82100, Italy (e-mail: amit.joshi, mtipaldi,
glielmo@unisannio.it)*

Abstract: The rationale of shifting towards green energy, along with the cost reduction and the increasing capacity of lithium-ion batteries, has motivated the end-users to go for energy storage systems integrated with solar technology solutions. Such systems provide the end-users with greater flexibility, thereby enhancing their role as prosumers in a range of grid-management programs. In this regard, we consider a residential household equipped with a battery and photovoltaic panels, collectively known as the photovoltaic-battery (PV-B) system. We further learn (off-line) a deterministic sub-optimal policy for charging/discharging of the residential battery using an actor-critic reinforcement learning based method. Such proposed approach, named polynomial deterministic policy gradient (PDPG), does not require any model of the system and uses polynomials as function approximator, as opposed to conventional neural networks. The use of polynomials makes the learning process faster and the method can also be applied for on-line learning. The usefulness of the proposed approach is tested on real power data (demand and PV generation) of a residential household in Australia. Numerical simulations indicate that the proposed PDPG algorithm outperforms the OFFON control approach in terms of electricity bill savings and the model-based receding horizon control in terms of computation time.

Keywords: Q-Learning, Actor-Critic, Photovoltaic-Battery Systems, Polynomial Deterministic Policy Gradient.

1. INTRODUCTION

Solar energy is witnessing a prominent role in the realization of green grids, thanks to policies supporting cheap installations, incentive tariff rates and low maintenance costs, see Zhang et al. (2017). According to the International Renewable Energy Agency (IRENA), 60% of the overall capacity growth in renewable technologies in 2019 was related to solar-powered systems, see Rabaia et al. (2020). However, the growing solar penetration poses technical challenges in terms of voltage and frequency regulation in the power grid, see Eftekharnejad et al. (2012). As an alternative, the integration of storage systems with solar energy resources is gaining popularity, as they can account for the intermittent nature of the solar energy resource, see Zahedi (2011). Some notable projects in this direction are: the Kauai Island Utility Cooperative [1], which unveiled the world's first utility-scale solar plus battery (13 megawatt Tesla solar field coupled with a 52 megawatt hour battery storage system) and the ambitious 4500 km Australia-Singapore Power Link [2] to be completed by late 2027 (solar farm producing 10 gigawatts of electricity, with a storage unit of 30 gigawatt hour).

Drawing parallels with the residential sector, there has been an increase in the installation of photovoltaic-battery (PV-B) systems, see Linssen et al. (2017). They assume the self-consumption of locally generated solar energy instead of the complete grid feed-in. Self-consumption of the PV energy has the following two major advantages 1) it resonates with the idea of minimizing the PV penetration into the grid, see Eftekharnejad et al. (2012) and 2) it can aid in minimizing the electricity bill of the end-user under time-varying price, see Weniger et al. (2014). Both the aforementioned objectives require control strategies for charging and discharging of the battery storage, also known as battery scheduling.

1.1 Related work

Control design strategies for battery scheduling can broadly be classified into model-based and model-free approaches. As for the former, we solve either a deterministic or a stochastic dynamic optimization problem subject to the dynamic model of the system at hand, along with the associated set of constraints, see Garcia et al. (1989). To this end, Elkazaz et al. (2020) proposed a model predictive control (MPC) based home energy management system (HEMS) for a home microgrid with PV generation. To include the disturbances on solar energy and load demand, Han et al. (2017) proposed a switched MPC approach. Further, Garifi et al. (2018) proposed chance-constrained MPC algorithm for demand response in a HEMS. An optimal energy management for a grid-connected PV-B hybrid system was proposed in Wu et al. (2015), where an optimal control is developed to schedule

[1] https://website.kiuc.coop/renewables (accessed on November 22, 2020)
[2] https://suncable.sg/ (accessed on November 22, 2020)

the power flow over 24 hours and a model predictive control is used as a closed-loop method to dispatch power in real-time.

In the model-free approach, we learn control policies (stochastic or deterministic) using the historical data of the system collected over time, see Sutton and Barto (2018). To this end, Kim and Lim (2018) proposed a tabular Q-learning for a discrete action space to minimize the operation cost of a grid-connected smart energy building. An optimization based home energy management strategy for a discrete action space based on deep Q-learning (DQN) and double deep Q-learning (DDQN) was proposed by Liu et al. (2020). In order to include a continuous action space, Yu et al. (2019) proposed a deep deterministic policy gradient (DDPG) based HEMS. Further, Li et al. (2020) proposed a policy search algorithm based on trust region policy optimization (TRPO) to train a neural network for real-time demand response. The proposed approach is capable of handling both discrete and continuous action spaces. A detailed review of DQN, DDQN, DDPG and TRPO in regards to smart building energy management can be found in Yu et al. (2020).

1.2 Objective and contribution of the paper

The objective of this work is the battery scheduling of a residential PV-B system, so as to minimize the electricity bill. To this end, we propose a model-free approach using Q-learning. The proposed approach is able to account for a continuous action space and learns a deterministic sub-optimal policy using a special class of actor-critic algorithms, based on deterministic policy gradient. The novelty of the work lies in using multivariate polynomials of degree 1 and 2 as opposed to state-of-art deep neural networks (DNN). The resulting algorithm, named polynomial deterministic policy gradient (PDPG), can be used to learn off-line a deterministic sub-optimal policy for charging/discharging of a residential battery system. In this respect, we present a comparative study between the proposed approach and the state-of-art methods for battery scheduling using the historical power and tariff data of a residential household in Australia and highlight the key points.

The rest of the paper is organized as follows. The PV-B system model and some state-of-art battery scheduling approaches are described in Section 2. After providing some background information on a typical reinforcement learning (RL) framework along with its related definitions, Section 3 presents the proposed PDPG algorithm as well as the specific RL system elements for the battery system scheduling problem. A comparative study highlighting the performance of the proposed and state-of-art scheduling algorithms is included in Section 4, followed by concluding remarks and future work in Section 5.

2. MODEL-BASED BATTERY SCHEDULING

This section describes the PV-B system model of the addressed residential household. Such model is used to solve model-based battery scheduling problems by means of specific state-of-art model-based control approaches, which are also outlined in this section. Such model-based control approaches will be used as benchmark for assessing the performance of the proposed PDPG based agent.

2.1 The PV-B system model

The considered residential building consists of a PV-B system and a smart inverter, see Fig. 1. The smart inverter decides the scheduling strategy of the battery so as to minimize the electricity bill, while taking into account the net load (difference between demand and PV power generation) and the tariff rates.

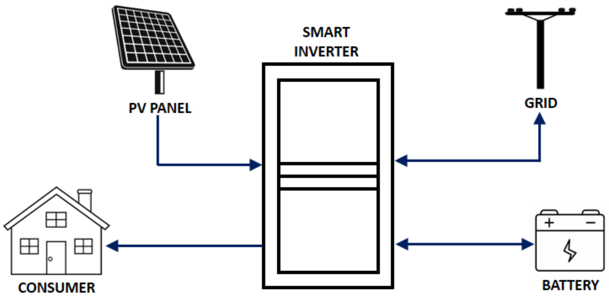

Fig. 1. Residential building under consideration.

Let $t \in \mathbb{Z}_+$ be the discrete time-step and p_t^{d}, p_t^{PV}, p_t^{B}, p_t^{R} denote the power demand, the PV power generation, the power to battery, the power from retailer, respectively; β indicates the energy capacity of the battery, with $\alpha^{\mathrm{B,ch}}$ and $\alpha^{\mathrm{B,dch}}$ as the charging and discharging efficiency, and $\overline{p}^{\mathrm{B,ch}}$ and $\overline{p}^{\mathrm{B,dch}}$ as the maximum charging and discharging power, respectively. The state-of-charge of the battery is denoted as SoC_t, with $\underline{\mathrm{SoC}}$ and $\overline{\mathrm{SoC}}$ as the minimum and maximum allowable state-of-charge; c_t^{R} and c_t^{F} denote the retailer tariff and the feed-in-tariff, respectively.

The load balance equation for the building at time t, can be written as

$$p_t^{\mathrm{R}} = p_t^{\mathrm{B}} + p_t^{\mathrm{d}} - p_t^{\mathrm{PV}}, \tag{1}$$

where $p_t^{\mathrm{B}} > 0$ indicates the charging of the battery storage (and vice versa); $p_t^{\mathrm{R}} > 0$ indicates the power purchased from the retailer (and vice versa).

The SoC is defined as the charge level of the battery in proportion to the energy capacity of the battery. The SoC dynamics of the battery can be expressed as

$$\mathrm{SoC}_{t+1} = \mathrm{SoC}_t + \Delta t(\tilde{\alpha}^{\mathrm{B,ch}} p_t^{\mathrm{B,ch}} - \tilde{\alpha}^{\mathrm{B,dch}} p_t^{\mathrm{B,dch}}), \tag{2a}$$

$$\underline{\mathrm{SoC}} \leq \mathrm{SoC}_t \leq \overline{\mathrm{SoC}}, \tag{2b}$$

where Δt is the time-interval between two consecutive time-steps; $\tilde{\alpha}^{\mathrm{B,ch}} = \alpha^{\mathrm{B,ch}}/\beta$ and $\tilde{\alpha}^{\mathrm{B,dch}} = \alpha^{\mathrm{B,dch}}/\beta$; SoC_0 is the initial SoC of the battery storage; $p_t^{\mathrm{B,ch}}$ and $p_t^{\mathrm{B,dch}}$ are the charging ($p_t^{\mathrm{B}} > 0$) and discharging ($p_t^{\mathrm{B}} < 0$) power of the battery storage, with $p_t^{\mathrm{B,ch}} - p_t^{\mathrm{B,dch}} = p_t^{\mathrm{B}}$. Further, $p_t^{\mathrm{B,ch}}$ and $p_t^{\mathrm{B,dch}}$ are constrained as

$$0 \leq p_t^{\mathrm{B,ch}} \leq \overline{p}^{\mathrm{B,ch}}, \tag{3a}$$

$$0 \leq p_t^{\mathrm{B,dch}} \leq \overline{p}^{\mathrm{B,dch}}, \tag{3b}$$

$$p_t^{\mathrm{B,ch}} p_t^{\mathrm{B,dch}} = 0, \tag{3c}$$

where (3a) and (3b) bound the battery charging and discharging power, respectively, while (3c) ensures non-simultaneous charging and discharging of the battery.

The cost of electricity at time t is given as

$$C(t) = \begin{cases} c_t^{\mathrm{R}} p_t^{\mathrm{R}}, & \text{if } p_t^{\mathrm{R}} > 0 \text{ (purchase from grid)}, \\ c_t^{\mathrm{F}} p_t^{\mathrm{R}}, & \text{if } p_t^{\mathrm{R}} < 0 \text{ (supply to grid)}. \end{cases} \quad (4)$$

In our study, the interval between two consecutive time-steps is 1 hour, i.e., $\Delta t = 1$ and we assume power as constant during such interval.

2.2 Receding Horizon Control (RHC)

It involves repeatedly solving a constrained optimization problem, via predictions of future costs, disturbances and constraints over a moving time horizon in order to choose the control action, see Mattingley et al. (2011). The RHC based battery scheduling is defined as

$$\begin{aligned} \min_{\boldsymbol{p}^{\mathbf{B}}} \quad & \sum_{k=t}^{t+T} C(k) \\ \text{subject to} \quad & (1), (2), (3), \end{aligned} \quad (5)$$

where $\boldsymbol{p}^{\mathbf{B}} = [p_t^{\mathrm{B}}, \ldots, p_{t+T}^{\mathrm{B}}]'$. Note that the control action is always p_t^{B} in all the model-based control approaches mentioned in this paper.

2.3 OFFON approach

In this approach, the battery daily undergoes one full cycle based on the specified depth of discharge. The charging and discharging occur at constant rate over several time-steps during off-peak and peak hours respectively, see Nottrott et al. (2013). Accordingly, the power to the battery (constant charging and discharging rate) is taken as

$$p_t^{\mathrm{B}} = \begin{cases} \beta(\overline{\mathrm{SoC}} - \underline{\mathrm{SoC}})/\alpha^{\mathrm{B,ch}}\, k_{\mathrm{off,peak}}, & \text{(off-peak hours)}, \\ \beta(\underline{\mathrm{SoC}} - \overline{\mathrm{SoC}})/\alpha^{\mathrm{B,dch}}\, k_{\mathrm{peak}}, & \text{(peak hours)}, \end{cases} \quad (6)$$

where $k_{\mathrm{off,peak}}$ and k_{peak} are the number of discrete time-steps (Δt) for the off-peak and peak hours, respectively.

3. MODEL-FREE BATTERY SCHEDULING

Reinforcement learning models the interaction between the agent and the environment as a Markov decision process (MDP), defined by the tuple $(\mathcal{X}, \mathcal{U}, \mathcal{P}, \mathcal{R}, \gamma)$. At each discrete time-step t, the agent takes an action $u_t \in \mathcal{U} \subset \mathbb{R}^m$. As a consequence of this action, its environment randomly performs a state transition from state $x_t \in \mathcal{X} \subset \mathbb{R}^n$ to $x_{t+1} \in \mathcal{X} \subset \mathbb{R}^n$ according to the time-homogeneous state transition probability distribution $\mathcal{P}(x_{t+1}|x_t, u_t)$, and returns a reward signal $r_t \in \mathcal{R} \subset \mathbb{R}$, which can also be associated to some reward probability distribution. Assuming the agent follows a stationary policy $\pi : \mathcal{X} \to \mathcal{U}$, the discounted sum of future rewards over the horizon (also known as return) is given by $G_t = \sum_{i=t}^{\infty} \gamma^{i-t} r_i$, where $0 \leq \gamma \leq 1$ is the discount factor. The objective therefore is to learn a policy π, which maximizes the expected return. To do so, given a specific stationary policy π, a value is associated with each state, called as the value function $V_\pi : \mathcal{X} \to \mathcal{R}$, given as

$$V_\pi(x_t) = \mathbb{E}_\pi[G_t|x_t] = \mathbb{E}_\pi\left[\sum_{i=t}^{\infty} \gamma^{i-t} r_i \middle| x_t\right], \forall x_t \in \mathcal{X}. \quad (7)$$

It can be understood as the expected return achieved, when in state x_t and following policy π. An optimal policy π^* is the one which maximizes the value function for all the initial states, i.e., $\pi^* = \arg\max_{\pi \in \Pi} V_\pi(x_t), \forall x_t \in \mathcal{X}$, where Π is the set of all feasible stationary policies. All the optimal policies share the same optimal value function, defined as $V^*(x_t) := V_{\pi^*}(x_t)$, see Sutton and Barto (2018).

Likewise, a value is associated with each state conditioned on a given initial action, called as the action-value function or the Q-function $Q_\pi(x_t, u_t) : \mathcal{X} \times \mathcal{U} \to \mathcal{R}$, given as

$$Q_\pi(x_t, a_t) = \mathbb{E}_\pi[r_t + \gamma V_\pi(x_{t+1})], \forall x_t \in \mathcal{X}, u_t \in \mathcal{U}. \quad (8)$$

It can be understood as the expected return achieved when in state x_t, taking u_t as the initial action and following policy π thereafter. All the optimal policies share the same optimal action-value function, defined as, $Q^*(x_t, u_t) := Q_{\pi^*}(x_t, u_t)$, see Sutton and Barto (2018).

Remark 2: For finite MDPs, there exists at least one optimal deterministic policy that is no worse than any other policy, see Sutton and Barto (2018) and can be obtained as

$$\mu^*(x_t) = \arg\max_{u_t \in \mathcal{U}} Q^*(x_t, u_t), \forall x_t \in \mathcal{X}, \quad (9)$$

where a deterministic policy is the one which admits only one control action for each state, i.e., $u_t = \mu(x_t)$. In line with the literature, π and μ are hereafter used interchangeably since this paper addresses only stationary policies, see Bertsekas (2012).

For the sake of simplicity, in the paper, we denote with the same symbol both the discrete time indexed random variable and the data sampled according to the related probability distribution during, e,g., a training episode.

3.1 PDPG: an Actor-Critic approach

Policy gradient (PG) algorithms belong to the actor-critic family, see Konda and Tsitsiklis (2000) and holds the merit of handling continuous action spaces to produce a smooth control reference signal. The basic idea is to parameterize policies $\pi \in \Pi$ by a parameter vector $\theta \in \mathbb{R}^d$, i.e., $\pi = \pi_\theta$ to avoid policy search over a complicated function class. Further, the policy parameters are updated in the direction of the gradient of the value function, see Sutton et al. (1999). As an alternative, Silver et al. (2014) proposed the update of policy parameters update in the direction of the gradient of the action-value function, known as deterministic policy gradient (DPG).

The proposed polynomial deterministic policy gradient (PDPG) leverages on the DPG, see Algorithm 1. As opposed to using deep-neural networks, see Lillicrap et al. (2015), PDPG uses first and second order multivariate polynomials as function approximators. In Algorithm 1, $\pi_\theta = \mu(x_t|\theta^\mu)$ is the parameterized actor function, which predicts the best action in a given state and $Q(x_t, u_t|\theta^Q)$ is the parameterized critic function, which estimates the optimal action-value function Q^* for each state-action pair. At each time-step t, the transition tuple (x_t, u_t, r_t, x_{t+1}) is stored in the replay buffer (RB), which is later used to update the weights of the actor and critic functions. In order to improve the stability while learning, copies of actor and critic functions, called as target functions are created. Such functions are parameterized by $\theta^{\mu'}$ and

Algorithm 1 Deterministic Policy Gradient

```
Input: $\theta^{\mu}, \theta^{Q}, \mathrm{RB}, N, \gamma, \tau$
Output: $\theta^{\mu'}$
Initialization: $\theta^{\mu'} \leftarrow \theta^{\mu},\ \theta^{Q'} \leftarrow \theta^{Q}$,
for $e \leftarrow 1$ to $E$ do
    Initialization: $\mathcal{N}$,$x_0$
    for $t \leftarrow 1$ to $S$ do
        Select action $u_t = \mu(x_t|\theta^{\mu}) + \mathcal{N}_t$
        Execute $u_t$, observe $r_t$ and $x_{t+1}$
        Store $(x_t, u_t, r_t, x_{t+1})$ in RB
        if $t$ mod $T = 0$ then
            Sample at random a minibatch of $N$ transitions $(x_i, u_i, r_i, x_{i+1})$ from RB
            Set $y_i = r_i + \gamma Q(x_{i+1}, \mu(x_{i+1}|\theta^{\mu'})|\theta^{Q'})$
            Update critic using (10)
            Update actor using (11)
            Update target actor and target critic using (12a) and (12b), respectively.
        end if
    end for
end for
```

$\theta^{Q'}$. Further, to speed up convergence, an exploratory behaviour of the agent is achieved by injecting correlated stochastic noise to the output of the actor function. The noise $\mathcal{N}$ is typically modelled as an Ornstein-Uhlenbeck process, see Uhlenbeck and Ornstein (1930), defined as $\mathcal{N}_t = \mathcal{N}_{t-1} + \kappa(\mu_{\mathcal{N}} - \mathcal{N}_{t-1})T_s + \sigma_{\mathcal{N}} n\sqrt{T_s}$, where κ is the "mean attraction constant", $\mu_{\mathcal{N}}$ and $\sigma_{\mathcal{N}}$ are the mean value and variance of the noise model, T_s is the agents sample time, and n is a random number uniformly selected between 0 and 1. Consequently, the action of the agent can be written as $a_t = \mu(x_t|\theta^{\mu}) + \mathcal{N}_t$. The weights of the actor and critic are updated after every T time-steps, where-in the critic is updated by minimizing the loss function (in terms of mean-squared error) between the target critic's estimate and the critic's estimate, given as

$$L = \frac{1}{N}\sum_{i}^{N}(y_i - Q(x_i, u_i|\theta^{Q}))^2. \tag{10}$$

The minimization of (10) is performed w.r.t. θ^{Q}, using a mini-batch of state transitions randomly sampled from the replay buffer, i.e., N. Sampling at random diminishes the correlation induced errors, see Silver et al. (2014). Then, the weights of the actor function are updated in the direction of the gradient of parameterized action-value function, given as

$$\nabla_{\theta^{\mu}} J \approx \frac{1}{N}\sum_{i}^{N} \nabla_{u} Q(x_i, u|\theta^{Q})|_{u=\mu(x_i|\theta^{\mu})} \times \nabla_{\theta_{\mu}}\mu(x_i|\theta^{\mu}), \tag{11}$$

where $J = Q(x, u|\theta^{Q})$. Finally, the weights of target actor and target critic are smoothly updated to prevent learning instabilities as

$$\theta^{\mu'} = \tau\theta^{\mu} + (1-\tau)\theta^{\mu'}, \tag{12a}$$

$$\theta^{Q'} = \tau\theta^{Q} + (1-\tau)\theta^{Q'}, \tag{12b}$$

where τ represents the smoothing factor. As the data available is limited, we iterate the entire process for E number of iterations, called as epoch.

Remark 3: The convergence of Algorithm 1 can be achieved for bounded rewards and differentiable policies, see Zhang et al. (2019).

3.2 Defining state, action and reward for the battery scheduling problem

The state of the agent at time t is composed by the SoC, the power demand, the PV power generation, the retailer tariff and the feed-in tariff, given by

$$x_t = (x^1, x^2, x^3, x^4, x^5) = (\mathrm{SoC}_t, p_t^{\mathrm{d}}, p_t^{\mathrm{PV}}, c_t^{\mathrm{R}}, c_t^{\mathrm{F}}). \tag{13}$$

At state x_t, the action taken by the agent is defined as

$$u_t = \begin{cases} \min\{\mu_t,\ \beta(\overline{\mathrm{SoC}} - x^1),\ \overline{p}^{\mathrm{B,ch}}\}, & \text{for } \mu_t \geq 0 \\ \max\{\mu_t,\ \beta(\underline{\mathrm{SoC}} - x^1),\ \overline{p}^{\mathrm{B,dch}}\}, & \text{for } \mu_t < 0 \end{cases}, \tag{14}$$

where $\mu_t = \mu(x_t|\theta^{\mu})$ is the output of the actor function, which coincides with either the charging or the discharging power of the battery storage.

The reward received by the agent from the environment on taking action u_t in state x_t is described in Algorithm 2. The overall reward (r_t) is composed of three terms, namely r_t^{R}, r_t^{SoC} and r_t^{B}. The first term indicates the cost to be paid by the prosumer for meeting the net load of the house. The second term is the penalty to be paid ($R^{\mathrm{SoC}} < 0$), for violating the SoC limits (2b). Likewise, the third term is the penalty ($R^{\mathrm{B}} < 0$) for violating the maximum charging and discharging power constraints (3a & 3b).

Algorithm 2 Reward by the Environment

```
Input: $c_t^{\mathrm{R}}, c_t^{\mathrm{F}}, p_t^{\mathrm{R}}, \mathrm{SoC}_t, \underline{\mathrm{SoC}}, \overline{\mathrm{SoC}}, p_t^{\mathrm{B}}, \overline{p}^{\mathrm{B,ch}}, \overline{p}^{\mathrm{B,dch}}$
Output: $r_t$
Initialization: $r_t^{\mathrm{E}} \leftarrow 0,\ r_t^{\mathrm{SoC}} \leftarrow 0,\ r_t^{\mathrm{B}} \leftarrow 0$
if $p_t^{\mathrm{R}} \geq 0$ then
    $r_t^{\mathrm{E}} = -c_t^{\mathrm{R}} p_t^{\mathrm{R}}$
else
    $r_t^{\mathrm{E}} = -c_t^{\mathrm{F}} p_t^{\mathrm{R}}$
end if
if $\mathrm{SoC}_t \leq \underline{\mathrm{SoC}}$ or $\mathrm{SoC}_t \geq \overline{\mathrm{SoC}}$ then
    $r_t^{\mathrm{SoC}} = R^{\mathrm{SoC}}$
end if
if $p_t^{\mathrm{B}} \leq -\overline{p}^{\mathrm{B,dch}}$ or $p_t^{\mathrm{B}} \geq \overline{p}^{\mathrm{B,ch}}$ then
    $r_t^{\mathrm{B}} = R^{\mathrm{B}}$
end if
$r_t = r_t^{\mathrm{E}} + r_t^{\mathrm{SoC}} + r_t^{\mathrm{B}}$
```

In the PDPG algorithm, we choose polynomials for function approximation. The actor functions are taken as

$$\mu_{\mathrm{Lin}}(x|\theta^{\mu}) = \theta_0^{\mu} + \sum_{i=1}^{n}\theta_i^{\mu} x^i, \tag{15a}$$

$$\mu_{\mathrm{Quad}}(x|\theta^{\mu}) = \theta_0^{\mu} + \sum_{i=1}^{n}\theta_i^{\mu} x^i + \sum_{i=1}^{n}\sum_{j=1}^{n}\theta_{ij}^{\mu} x^i x^j, \tag{15b}$$

where $\mu_{\mathrm{Lin}}(.)$ and $\mu_{\mathrm{Quad}}(.)$ are degree 1 and degree 2 multivariate polynomials, respectively. Likewise, the critic functions are taken as

$$Q_{\mathrm{Lin}}(x, u|\theta^{Q}) = \theta_0^{Q} + \sum_{i=1}^{n+m}\theta_i^{Q}[xu]^i, \tag{16a}$$

$$Q_{\mathrm{Quad}}(x, u|\theta^{Q}) = \theta_0^{Q} + \sum_{i=1}^{n+m}\theta_{ij}^{Q}[xu]^i + \sum_{i=1}^{n+m}\sum_{j=1}^{n+m}\theta_{ij}^{Q}[xu]^i[xu]^j, \tag{16b}$$

where $[xu] = (x^1, x^2, x^3, x^4, x^5, u)$ is the augmented state-action vector, $Q_{\text{Lin}}(.)$ and $Q_{\text{Quad}}(.)$ are degree 1 and degree 2 multivariate polynomials, respectively. Note that we have $n = 5$ and $m = 1$ in our battery scheduling problem.

4. NUMERICAL SIMULATIONS AND RESULTS

We use the power data of a single house (House 11) from the AUSGRID [3] dataset, which contains year long power demand and PV power generation for 300 residential houses at a resolution of 30 minutes. Further, we take the hourly retailer tariff scheme (Table 1) and flat feed-in tariff of 11 cents/kWh, as per the Australian Distribution Network Service Provider, described in Bean and Khan (2018). For OFFON control (6), shoulder tariff also constitutes the peak hours, i.e, $k_{\text{peak}} = 15$ and $k_{\text{off,peak}} = 9$.

Table 1. Retailer tariff.

Interval of the Day	Electricity Tariff (cents/kWh)		
	Off-Peak (20.3)	Shoulder (25.6)	Peak (36)
00:00 - 07:00	✓		
07:00 - 16:00		✓	
16:00 - 20:00			✓
20:00 - 22:00		✓	
22:00 - 00:00	✓		

To learn the sub-optimal policy using PDPG, we extract the data from Day 1 to Day 250, out of which the first 50 days are used to generate the replay buffer and the next 200 days are used to run the PDPG algorithm core. The components of the state such as p^{D}_t, p^{PV}_t, c^{R}_t and c^{F}_t can be accessed using the historical data, while SoC_t can be computed using (2a). Using the policy gradient algorithm, the actor policy is learnt, which defines the battery scheduling strategy. We test the actor policy on the test data, which is from Day 251 to Day 350.

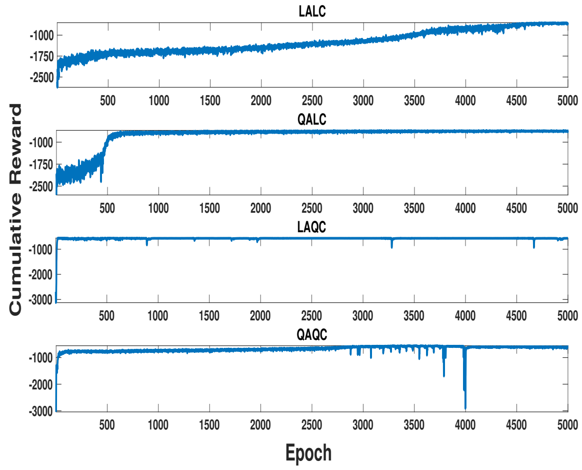

Fig. 2. Training with PDPG

The model-based parameters are set as $\beta = 2.04$, $\underline{\text{SoC}}_n = .1$, $\overline{\text{SoC}}_n = .9$, $\alpha^{\text{ch}}_{\text{B},n} = .9$, $\alpha^{\text{dch}}_{\text{B},n} = 1.1$, $\overline{p}^{\text{ch}}_n = \overline{p}^{\text{dch}}_n = .25\beta/\Delta t$. Further, the hyper-parameters in Algorithm 1 are set to $\gamma = 0.9$, $\tau = 0.003$, $N = 240$, $\text{T} = 24$, $E = 5000$, $\alpha = 0.001$, $N = 240$, $\kappa = 2$, $\mu_{\mathcal{N}} = 0$, $\sigma_{\mathcal{N}} = .1$.

Fig. 2 shows the cumulative reward received in each epoch, for different combinations of linear actor (LA), quadratic actor (QA), linear critic (LC) and quadratic critic (QC). It can be seen that the agent steadily explores when the critic function is linear, as opposed to quadratic critic function. Further, the cumulative reward settles around -600 for all four combinations. Table 2 shows the runtime of the PDPG algorithm over 5000 epochs. As expected, due to (10), the quadratic critic function influences the runtime significantly.

Table 2. Runtime of the PDPG algorithm.

Runtime (in seconds)	PDPG Algorithm			
	LALC	LAQC	QALC	QAQC
training	1193	2073	1819	2697

We now compare the proposed PDPG algorithm with the state-of-art model-based RHC and OFFON control. Table 3 shows the electricity bill to be paid using RHC, OFFON and PDPG for the training and the testing days. The following aspects can be noted: (i) the model-based RHC control strategy performs the overall best, both on the training and testing days; (ii) the proposed PDPG performs at par or better than the state-of-art OFFON control strategy; (iii) for different combinations of actor and critic functions, the one with linear actor and linear critic performs the best both on the training and the test data.

Table 3. Comparing costs for RCH, OFFON and PDPG.

Control Strategy	Cost in dollar ($)		Loss in (%) w.r.t. RHC	
	Training	Testing	Training	Testing
	(200 days)	(100 days)	(200 days)	(100 days)
RHC	**502.59**	**384.49**	-	-
OFFON	568.08	411.40	13.03	7.01
LALC	549.59	407.35	**9.35**	**5.94**
LAQC	561.65	411.32	11.75	6.98
QALC	572.40	416.68	13.89	8.37
QAQC	567.87	413.85	12.99	7.64

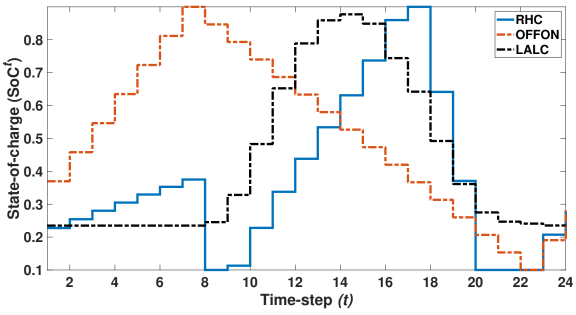

Fig. 3. SoC profile for a random day.

Fig. 3 highlights the charging and discharging profile of the residential battery for a random test day, obtained using RHC, OFFON and LALC based PDPG.

Finally, we compare the computation time required for taking the control action on both training and test data, given in Table 4

[3] https://www.ausgrid.com.au (accessed on November 22,2020)

Table 4. Computation time.

Data	Computation Time (in seconds)		
	RHC	OFFON	LALC
Training (200 days)	1448	**.0027**	.014
Test (100 days)	716	**.0012**	.0039

5. CONCLUSION AND FUTURE WORK

In this work, we proposed a polynomial based deterministic policy gradient (PDPG) approach for battery scheduling of a residential household. We further compared the proposed approach with state-of-art model based RHC and OFFON control, by using the real dataset of a residential house. The proposed PDPG outperforms the RHC and the OFFON control in terms of computation time and electricity bill savings, respectively. Interesting future research directions would be 1) to use decision trees as function approximator, so as to have an interpretable rule-based control strategy and 2) to model the problem as a multi-agent system, where the electricity tariff depends on the aggregate decision of the end-users.

REFERENCES

Bean, R. and Khan, H. (2018). Using solar and load predictions in battery scheduling at the residential level. *arXiv preprint arXiv:1810.11178*.

Bertsekas, D. (2012). *Dynamic programming and optimal control, Vol. II (4th ed.)*. Belmont, Massachusetts: Athena Scientific.

Eftekharnejad, S., Vittal, V., Heydt, G.T., Keel, B., and Loehr, J. (2012). Impact of increased penetration of photovoltaic generation on power systems. *IEEE transactions on power systems*, 28(2), 893–901.

Elkazaz, M., Sumner, M., Pholboon, S., Davies, R., and Thomas, D. (2020). Performance assessment of an energy management system for a home microgrid with pv generation. *Energies*, 13(13), 3436.

Garcia, C.E., Prett, D.M., and Morari, M. (1989). Model predictive control: theory and practice—a survey. *Automatica*, 25(3), 335–348.

Garifi, K., Baker, K., Touri, B., and Christensen, D. (2018). Stochastic model predictive control for demand response in a home energy management system. In *2018 IEEE Power & Energy Society General Meeting (PESGM)*, 1–5. IEEE.

Han, X., Ao, N., and Wu, Z. (2017). A switched mpc approach of hybrid system for demand side management. In *2017 36th Chinese Control Conference (CCC)*, 9197–9202. IEEE.

Kim, S. and Lim, H. (2018). Reinforcement learning based energy management algorithm for smart energy buildings. *Energies*, 11(8), 2010.

Konda, V.R. and Tsitsiklis, J.N. (2000). Actor-critic algorithms. In *Advances in neural information processing systems*, 1008–1014.

Li, H., Wan, Z., and He, H. (2020). Real-time residential demand response. *IEEE Transactions on Smart Grid*.

Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. (2015). Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Linssen, J., Stenzel, P., and Fleer, J. (2017). Techno-economic analysis of photovoltaic battery systems and the influence of different consumer load profiles. *Applied Energy*, 185, 2019–2025.

Liu, Y., Zhang, D., and Gooi, H.B. (2020). Optimization strategy based on deep reinforcement learning for home energy management. *CSEE Journal of Power and Energy Systems*.

Mattingley, J., Wang, Y., and Boyd, S. (2011). Receding horizon control. *IEEE Control Systems Magazine*, 31(3), 52–65.

Nottrott, A., Kleissl, J., and Washom, B. (2013). Energy dispatch schedule optimization and cost benefit analysis for grid-connected, photovoltaic-battery storage systems. *Renewable Energy*, 55, 230–240.

Rabaia, M.K.H., Abdelkareem, M.A., Sayed, E.T., Elsaid, K., Chae, K.J., Wilberforce, T., and Olabi, A. (2020). Environmental impacts of solar energy systems: A review. *Science of The Total Environment*, 754, 141989.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. (2014). Deterministic policy gradient algorithms.

Sutton, R.S. and Barto, A.G. (2018). *Reinforcement learning: An introduction*. MIT press.

Sutton, R.S., McAllester, D., Singh, S., and Mansour, Y. (1999). Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1057–1063.

Uhlenbeck, G.E. and Ornstein, L.S. (1930). On the theory of the brownian motion. *Physical review*, 36(5), 823.

Weniger, J., Bergner, J., Tjaden, T., and Quaschning, V. (2014). Economics of residential pv battery systems in the self-consumption age. In *29th European Photovoltaic Solar Energy Conference and Exhibition*, 2936–2938.

Wu, Z., Tazvinga, H., and Xia, X. (2015). Demand side management of photovoltaic-battery hybrid system. *Applied Energy*, 148, 294–304.

Yu, L., Qin, S., Zhang, M., Shen, C., Jiang, T., and Guan, X. (2020). Deep reinforcement learning for smart building energy management: A survey. *arXiv preprint arXiv:2008.05074*.

Yu, L., Xie, W., Xie, D., Zou, Y., Zhang, D., Sun, Z., Zhang, L., Zhang, Y., and Jiang, T. (2019). Deep reinforcement learning for smart home energy management. *IEEE Internet of Things Journal*, 7(4), 2751–2762.

Zahedi, A. (2011). Maximizing solar pv energy penetration using energy storage technology. *Renewable and Sustainable Energy Reviews*, 15(1), 866–870.

Zhang, K., Koppel, A., Zhu, H., and Başar, T. (2019). Convergence and iteration complexity of policy gradient method for infinite-horizon reinforcement learning. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, 7415–7422. IEEE.

Zhang, Y., Lundblad, A., Campana, P.E., Benavente, F., and Yan, J. (2017). Battery sizing and rule-based operation of grid-connected photovoltaic-battery system: A case study in sweden. *Energy conversion and management*, 133, 249–263.