Battery Energy Management System Using Reinforcement Learning

July 14, 2017

1 Abstract

stochastic nature of Renewable energy sources such solar energy may lead to imbalance supply and demand in micro-grid environment. Energy storage solutions and advances battery control and operation can address this imbalance. In this study, dual deep Q-Network Reinforcement Learning (RL) framework is proposed for control and operation of a commercial building equipped with battery storage and photo-voltaic (PV) system. In addition this study address the gap which currently exists, where sophisticated optimal control strategies often applied to less accurate building models due to difficulties in applying such strategies in accurate building simulation environment. In this paper a detailed Energy Plus (EP) building model inter-acts with Python using Functional Mock-up Interface (FMI) which enables us to apply reinforcement learning based strategies to sophisticated building model in real time. The reinforcement learning agent learns the optimal energy management policy using past experiences.

2 Problem formulation

Modern building energy simulation tools such as EnergyPlus are optimized for building energy performance simulation but they only offer low level of support by controlling set-points and components availability. Advanced control strategies require flexible I/O manipulation which is not currently available in Energy Plus. In this study co-simulation is used for coupling Energy Plus and external reinforcement learning control algorithms implemented in Python. In this setup, Python is used a the master and Energy Plus model is compiled using co-simulation to FMU which can be run in python.

This study considers a small office building, PV system, inverters and a battery storage facility which is connected to the main grid. The additional electricity can be bought from the grid if the PV production and the battery storage cannot meet the demand. The office building is 2 story building where each floor ($3300\, ft^2$) has 2 north and south facing thermal zone. Each zone is served by a packaged single zone system consisting of an outside air economizer, DX coil, gas heating coil, and draw through supply air fan. There is night set up and setback. The fans are scheduled off at night.

System dynamics can be formalized as a partially obsevable Markov decision process where the reinforcement learning agent interacts with the environment. The Markovian process can be described with state space $\mathcal{S}$, action space $\mathcal{A}$ and reward ∇ evaluated every 10 minutes over finite time horizon.

2.1 State space

The space space is characterized using a tuple of six components ($s \in \mathcal{S}$) at each time step:

- Time Components: s^d and s^m corresponding to day of the week and moth of the year.
- PV production: s^{PV} solar panel energy production.
- Battery state: s^b energy level of the battery.
- Demand load: s^l building load.

2.2 Action space

The action space a^{cd} is the charging and discharging energy of the battery discretized (kWh). (Future: use a modified actor critic algorithm to extend the reinforcement learning to continuous action space which is more realistic)

The battery dynamic at each time step is describe bellow:

$$E_{t+1} = E_t + \eta_c a^{cd}, \quad a^{cd} > 0 \tag{1}$$

$$E_{t+1} = E_t + \frac{a^{cd}}{\eta_c}, \quad a^{cd} < 0 \tag{2}$$

Energy produced by the PV panel is calculated as:

$$P = A_{panel}\, f_{active}\, \eta_{cell}\, \eta_{inv}\, G_t \tag{3}$$

where

P : Electrical power produced by photovoltaics [W]
G_t : Total solar radiation incident on PV array [W/m2]
f_{active} : %Active area of pv
η_{cell} : PV efficiency
η_{inv} : Inverter efficiency

2.3 Reward function

The reward function is each time step is the negative of electricity that should be bought or sold to the grid.

$$r = -(E_t^{demand} - E_t^{PV} - E_t^{battery}) \tag{4}$$

the objective of the reinforcement learning algorithm is to maximize the reward. In order to improve the stability of the agent, the reward function is clipped to $r \in [-1, 1]$.

3 Deep reinforcement learning with double Q-learning network (DDQN)

For an agent following policy π the value state-action pair (s, a) and the state s are defined as follows

$$Q^{\pi}(s,a) = \mathbf{E}[R_t|s_t = s, a_t = a, \pi] \\ V^{\pi}(s) = \mathbf{E}_{a\sim\pi(s)}[Q^{\pi}(s,a)] \tag{5}$$

To solve the sequential Markov decision problem we need to learn the optimal value of each state as expected sum of the future reward when taking an action and following the optimal policy π.

$$Q_{\pi}(s,a) = \mathbf{E}[R_1 + \sum_{i=2} \gamma R_i|S_0 = s, A_0 = a, \pi] \tag{6}$$

where $\gamma \in [0, 1]$ is the discounted factor of future rewards which increases the importance of immediate rewards. The optimal policy is derived by choosing the highest valued action at each state which is $Q_*(s,a) = \max_{\pi} Q_{\pi}(s,a)$. Another important quantity is the advantage function relating the value of each state V^{π} to the Q function:

$$A^{\pi}(s,a) = Q^{\pi}(s,a) - V^{\pi}(s) \tag{7}$$

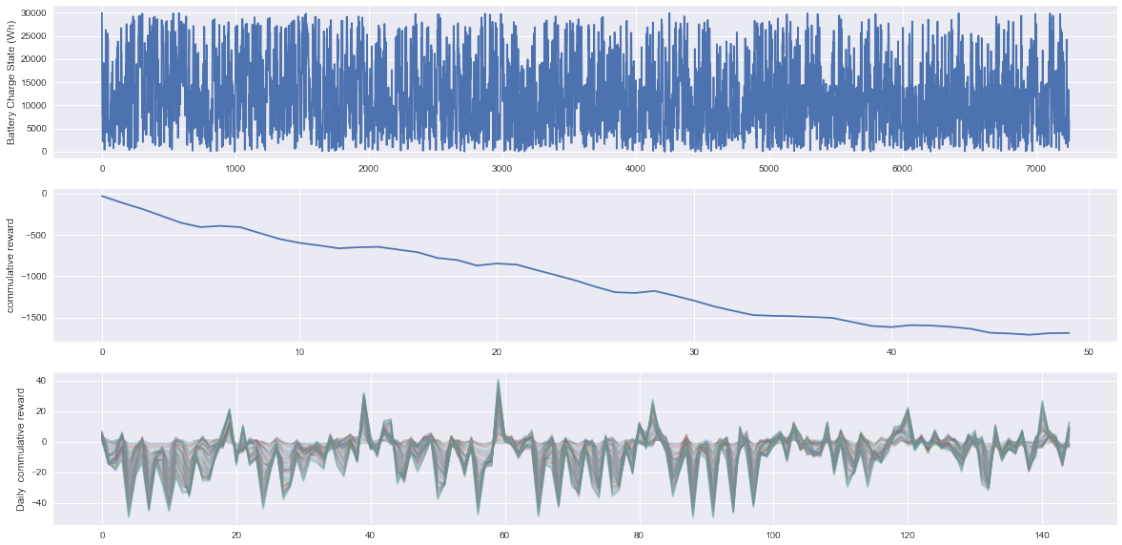
where the difference between value function V and the Q function is a relative measure of the importance of taking each action.

The double deep Q-network is a two multi layered neural network that for a given state s outputs the value of action values $Q(s,;\theta)$ where θ are the parameters of the network to be learned. Two properties of DDQN is the use of experience replay and separate target network, where both ingredients are proposed to improve the stability of predictions and reduce the over estimation of the certain actions which reduces the ability of the agent to learn new strategies. The experience replay is implemented by storing observed states and uniformly sample them for the memory to update the network. The target network with parameters θ^{-}, is the same as the on-line network except it is updated at every τ time step so that $\theta_t^{-} = \theta_t$. The target network output is defined as:

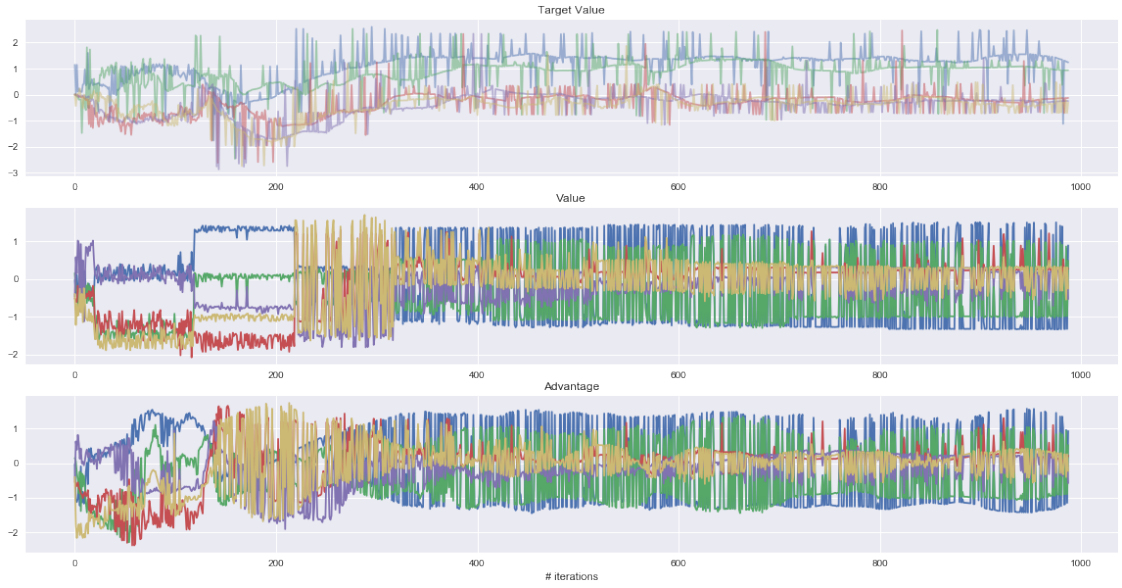
$$y_i^{DoubleDQN} = r + \lambda Q(s^{'}, argmax_{a^{'}} Q(s^{'}, a^{'}; \theta_i); \theta^{'}) \tag{8}$$

4 Experiment

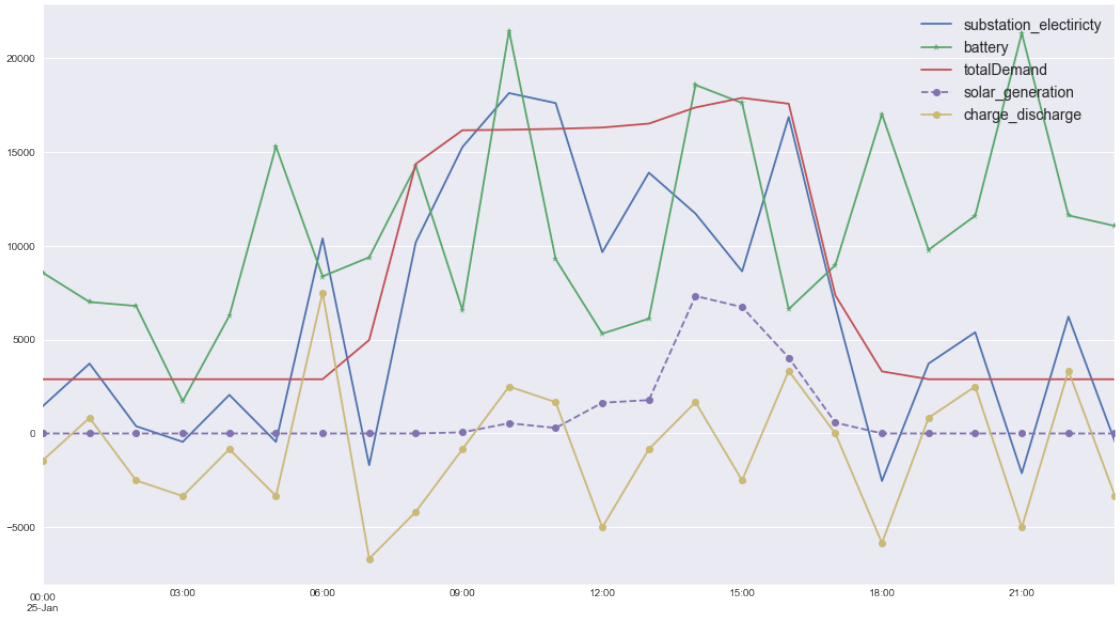
Training the agent for 50 days. Results shows that the agent behavior is almost random. The middle figure show the commutative reward during the simulation period. Since the agent is acting randomly, the reward received from the environment is negative and decreasing. After day 40, the magnitude of reward slope decreases which shows better performance.



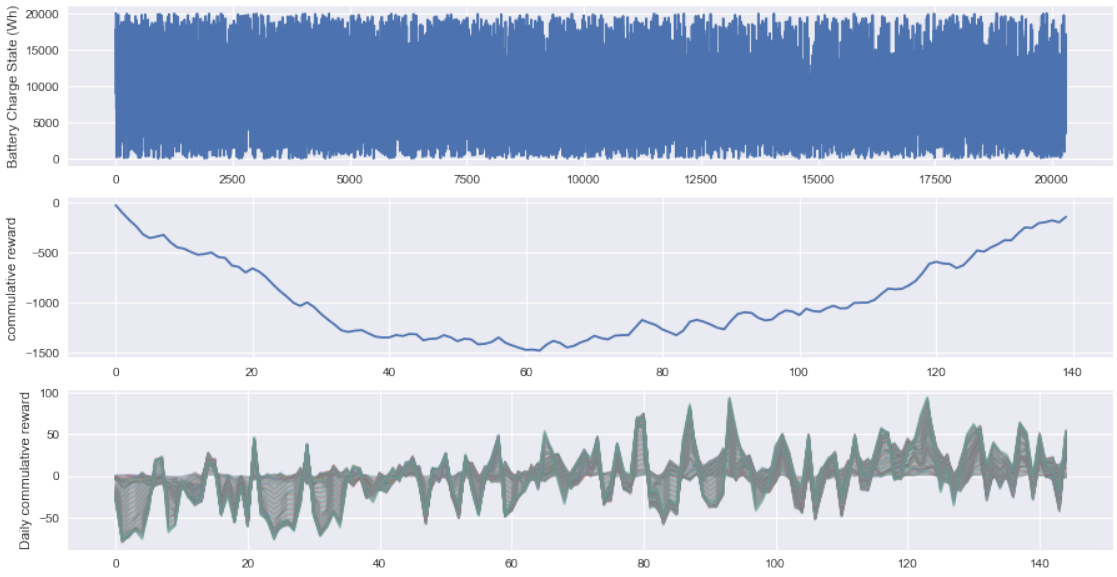
The Figures bellow show the target and Q-network action value function and value function which randomly oscillate about zero which again indicates random behavior.



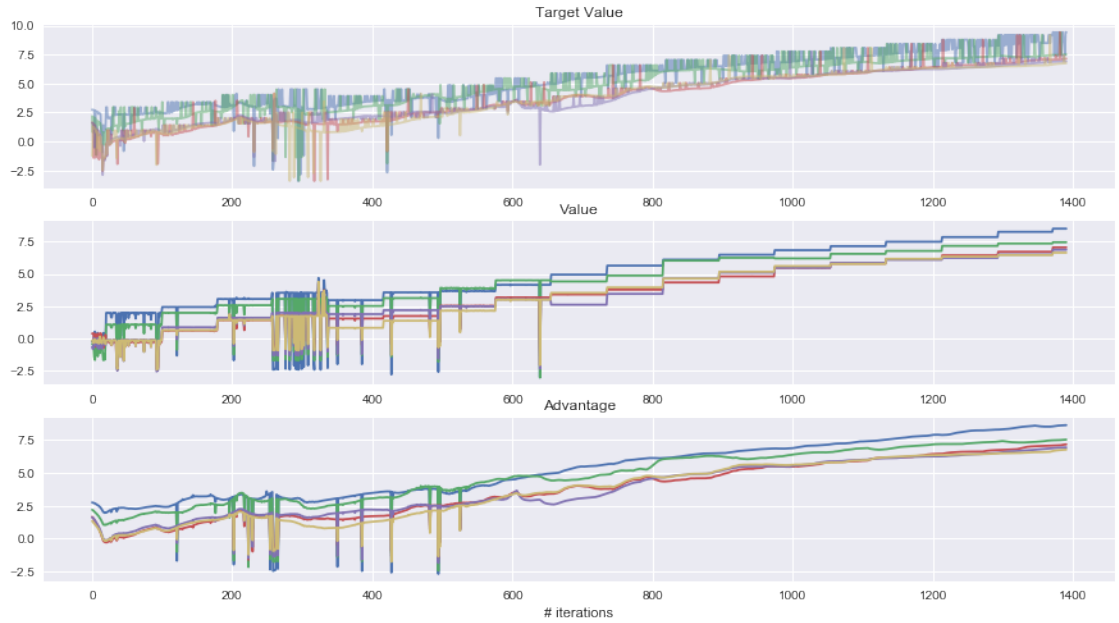
The building demand peaks around $4:30pm$ which is about $(17kW)$. There is some PV generation in the afternoon which slightly reduces the amount of electricity that is required to be bought. The random behavior of the battery is not helping. The agent decides to charge the battery around $10:00AM$ when there is no PV generation and building demand is near peak.



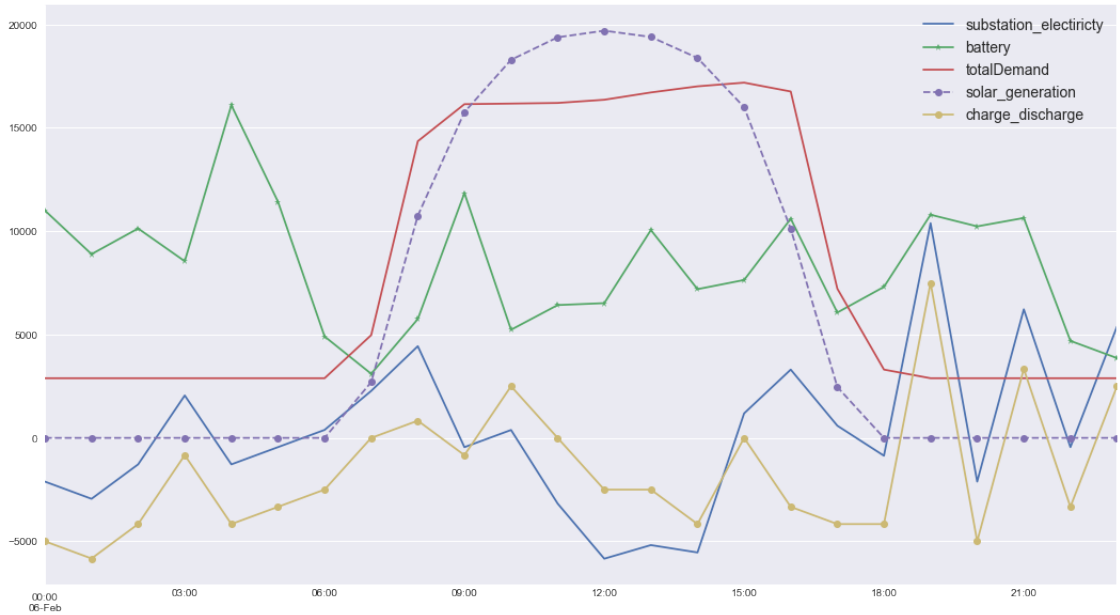
After additional 140 days training, the agent start to learn the mechanics of the environment. The middle figure bellow shows that the reward is flat between day 40 to 60 and after it starts to increase.



The Q and V values in the figure bellow shows that agent is learning the value of actions in each state. The positive slope shows that the Q-network output is increasing and further training can improve the performance.



The figure bellow presents the system dynamics during s single day (02/06/2017). This time the agent tries to increase the battery storage before peak hours and use it's stored capacity during the peak hours.



Future plans:

- The environment is designed to be modular and it can be extended to include any number of buildings, battery or PV panels.

- Modifying the reinforcement learning agent so that it can use a continuous action space which should improve the performance and prevent the unnecessary large jumps in the battery charge state.
- Modifying the reward function in order to consider the difference in buying and selling electricity price and battery charge/discharge cycle limitation.
- Control the HVAC system using reinforcement learning for passive thermal storage.