Battery Energy Management System Using Reinforcement Learning

Sam Mottahedi, *Member, IEEE,*

***Abstract*—The stochastic nature of Renewable energy sources such as solar energy may lead to imbalance in supply and demand in micro-grid environment. Energy storage solutions and advances in battery control and operation can address this imbalance. In this study, a Dueling Deep Q-Network (DDQN) 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 current building simulation environment.**

In this paper a detailed Energy Plus (EP) building model interacts with Python using Functional Mock-up Interface (FMI) which enables us to apply reinforcement learning based strategies to sophisticated building model. The reinforcement learning agent learns the optimal energy management policy using past experiences.

***Index Terms*—Reinforcement Learning, Dueling Deep Q-Network, Renewable Energy, Battery, Photo-voltaic System, Functional Mock-up Interface, Python.**

I. Introduction

In recent years concerns about global temperature and negative environmental impact and rising prices o fossil fuels lead to increased share of renewable energies sources (RES) [Katiraei 2008]. The high penetration of renewable energies in microgrids gave rise new set of challenges for control and operation of microgrids that require different strategies from those adapted for traditional power systems. Although additional energy sources can be financially more economic, environmentally conscience and consequently increase the security of the microgrid, operation and control of multiple energy sources will be challenging due to different charge discharge, control mechanics and response time [Xin Qiu 2016]. In addition to distributed generation and variable RES, energy storage technologies are necessary to balance the electricity supply and demand and reduce the dependency to grid.

In recent years, many studies concerning the energy management of microgrids have been published in literature. In [7] a multi-agent system was proposed based on Q-learning in order to reduce the power consumption of a solar microgrid. In [7] a Q-learning based demand response algorithm was applied to minimize the electricity consumption of electric water heater. [] proposes a multiagent system for power generation and power demand scheduling. [13] applied Q-learning to provide short-term ancillary services to the power grid by using a cluster TCLs. In [Kofinas] proposed a reinforcement learning based energy management in an islanded solar microgrid.

II. Related Work

III. Problem Statement

Modern building energy simulation tools such as Energy-Plus 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 observable 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.

A. 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.

B. 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

$$\begin{aligned} P &: \text{Electrical power produced by photovoltaics [W]} \\ G_t &: \text{Total solar radiation incident on PV array [W/m2]} \\ f_{active} &: \text{\%Active area of pv} \\ \eta_{cell} &: \text{PV efficiency} \\ \eta_{inv} &: \text{Inverter efficiency} \end{aligned}$$

C. 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 $r \in [-1, 1]$.

IV. 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

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

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 online network except it is updated at every τ time step so that $\theta_t^- = \theta_t$. The target network output is defined as:

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

V. Result and Discussion

VI. Conclusion

The conclusion goes here.

Appendix A
Proof of the First Zonklar Equation

Appendix one text goes here.

Appendix B

Appendix two text goes here.

Acknowledgment

The authors would like to thank...

References

[1] H. Kopka and P. W. Daly, *A Guide to LATEX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.

PLACE PHOTO HERE

Michael Shell Biography text here.

John Doe Biography text here.

Jane Doe Biography text here.