

Reinforcement Learning base DR Method for ESS SoC Optimization and Users Satisfaction

Yong Hee Park, Seong Gon Choi

Information & communication Engineering, Chungbuk National University, Chungcheongbuk-do, Korea

yh0360@chungbuk.ac.kr, choisg@chungbuk.ac.kr

Abstract— We proposed a Demand Response (DR) method to optimize Energy Storage System (ESS) State of Charge (SoC) and prevent user satisfaction decrease using Reinforcement Learning (RL). ESS SoC should be managed as an optimal value for reasons such as peak load responsiveness and battery life. However, the existing method for ESS SoC optimization assumed a fixed environment. In the actual environment, each factor, such as the departure time of the vehicle, is variable. Therefore, there is a need for a DR plan that can adapt to the environment. In addition, since DR reduces supply power for user, there is a problem of decrease user satisfaction. We aim to learn ESS SoC optimization, shifted load minimization, and optimal management of EV SoC. To this end, we formulate each element for RL and design environment to simulate. In the simulation results, we found the optimal policy.

Keywords— Smart Grid, Demand Response, ESS SoC, Peak Shaving, Reinforcement Learning

I. INTRODUCTION

Smart grid (SG) is a next-generation technology that combines information communication technology with power grids. Traditional power grids are inefficient in energy processing and utilization and have low reliability. It can collect energy data using smart meters, etc., and it uses this data to provide functions such as power grid control and monitoring [1], [2]. If the power-supply balance is broken in the power grid, problems such as power outage and waste of power occur. One of SG goal is to solve this power-supply problem using power information [3].

Renewable Generator (RG) is an energy source containing Photovoltaic (PV) and wind turbine, which has a significant impact on power system operations. However, it is difficult to provide reliability because it does not make a complete prediction of RG's power generation source. Therefore, various methods have been proposed to increase the reliability of RG [4] [5]. Energy Storage System (ESS) is a method that can be used for peak shaving, load shifting, etc. [6]. Since this can be utilized like UPS, methods have been proposed to improve the quality of power. It can also be used to solve the instability of the power produced by RG [7].

Demand Response (DR) is one of SG's methods of efficiently consuming power by interoperating between power operators and consumers [8]. This can be utilized for power

operation, such as reducing peak demand or securing reserve power [9]. Methods of DR include Interruptible Load (IL) that reduces user demand power, Vehicle to Grid (V2G) that supplies vehicle energy to grid, etc. [11].

The considerations of ESS state of charge (SoC) are as follows.

1. It is necessary to have sufficient SoC to cope with peak load that may occur at any time.
2. Due to factors such as battery life and electric bill, it should maintain the optimal SoC. [12]
3. In order to maintain self-power, it must have a minimum of SoC. [13].

However, ESS SoC is passive because it is determined by the supply and demand of power. In addition, ESS's SoC has limited capacity, so exceeding SoC wastes the generated power (RG). Therefore, the SoC of the ESS needs to be optimized [14]. When DR is performed, there is a concern that user satisfaction may decrease. In the case of IL, the quality of the service is degraded because the power supplied to the user is reduced. In the case of V2G, SoC of Electric Vehicle (EV) is also consumed to cope with peak load. If the user does not satisfy the required SoC of the EV at the time of using the EV, the service quality is degraded.

Reinforcement Learning (RL) is a technique in which agent and environment interact and learn using results for action in a specific state. This aims to maximize compensation for the results [15], [16]. Learning time may take a long time if the state and action are expressed in a complex way in RL. To address this, Deep Reinforce Learning (DRL) using the advantages of Deep Neural Network (DNN) was proposed [17]. Previously, we conducted a study to optimize ESS SoC using V2G [18]. However, the proposed V2G method did not take into account changes in various environments. Several DR methods via RL have also been proposed, but ESS SoC optimization and user satisfaction have not been considered [19]-[21].

Therefore, we propose an ESS SoC optimization method using DRL. The goal of learning is to optimize ESS SoC, minimize IL, and prevent SoC degradation at the start of EV. To this end, we define state, action, and return function and design an environment. The simulation results show to found policy for optimizing ESS SoC and user satisfaction. Section II describes existing studies on the optimization of ESS SoC and DR using RL. Section III describes RL formulation and

system design. Section IV shows the experimental results. Section V describes conclusions and future studies.

II. RELATED WORKS

A. Optimal ESS SoC

Depending on the ESS SoC, the effects of battery life, charging efficiency, etc. appear differently [12]. The minimum power for self-power maintenance of the ESS is about 10% [13]. One study proposed a method of calculating the optimal value of SoC in ESS [15]. If ESS SoC is small in an environment with RG, the UPS function of ESS is reduces. On the contrary, if the ESS SoC is high, if the power generation of RG exceeds the capacity of ESS, the remaining power cannot be stored and thus is wasted. Since the appropriate SoC of ESS varies depending on the facility and installation purpose, in the above paper, a method of estimating the SoC limit of ESS according to the state is presented. According to the results of the above studies, the SoC of ESS should aim to maintain the optimal value.

We proposed a method of optimizing the SoC of ESS using DR [18]. In that paper, we proposed a system that maintains the optimal SoC by operating the DR system when the SoC of the ESS becomes lower than the value desired by the user. The proposed DR system calculates the EV to discharge and the amount of discharge. The discharge status and the discharge amount are determined by considering the time when the EV departs. However, the proposed system fixed the departure time and departure time of the EV. In real situations, all environmental factors occur variably, so the DR system must be operated to suit the changing situation. Therefore, an environmental adaptive DR system using AI is required.

B. DR Method using RL

Several studies have been conducted to perform DR using RL [19]. One study performed IL measures for DR using Dueling DQL. To this end, state, action, and return functions were formulated. The study showed the effect of reducing the power load, but the environment of V2G and ESS SoC were not considered.

One study utilized Multi-agent RL (MARL) to implement Real Time Pricing (RTP) DR in microgrid [20]. This demonstrated RTP DR in environments with PV and ESS, but still did not consider the environment of V2G and ESS SoC.

One study proposed a V2G conversion method using DDQN [21]. The paper showed learning the relationship between charging rates, sales rates, and charging time to learn high profits. It also does not consider ESS SoC.

We propose a method for optimizing ESS SoC using DRL. To this end, each element of reinforcement learning is modeled and the environment is designed. The goal of reinforcement learning is to find policies that optimize ESS SoC while minimizing user satisfaction (IL, dissatisfaction of EV SoC).

III. DEMAND RESPONSE USING REINFORCEMENT LEARNING

A. Reinforcement Learning Background

RL aims to reward the results for the action in a particular state by interacting with the agent and the environment, and to maximize this [15], [16]. RL is specified in 5-tuple $\langle S, A, P, R, \gamma \rangle$ form of Markov Decision Process (MDP) [22], [23]. Each element is a state in a specific state, an action meaning selection in a specific state, a transition probability, a return on the result, and a discount factor meaning the depreciation rate of the next state. MDP is characterized by Markov property. Q-function estimates Q-value as a value function made up of state-action pairs. Q-Learning is an algorithm that continuously updates Q-value. It is shown in Equation 1.

$$Q(s_t, a_t) \leftarrow (1 - \alpha)Q(s_t, a_t) + \alpha(r_t + \gamma \max_a (Q(s_{t+1}, a) - Q(s_t, a_t))) \quad (1)$$

Since the q-value of Q-learning consists of a state-action pair, the more complex the relationship between state and action becomes, the more complex the learning becomes and may be late. To reduce the complexity of these Q-learning, DRLs have been proposed [17]. DRL solves the complexity of the state-action pair expressed in the existing matrix using DNN.

B. Reinforcement Learning Formulation of Problem

The elements related to State can be seen in Table 1.

TABLE 1. OBSERVATION STATE

Notation	Description	Unit
t	Current time	Hour
ESS_t	ESS SoC for time t	kW
EV_t^n	EV SoC for time t of n	kW
IL_t	Interruptible Load for time t	kW
NIL_t	Non-Interruptible Load for time t	kW
$EV_t^{n, left}$	The time left until n's EV leaves.	Hour

n means the sequence of EVs, and m is the total number of EVs. The vector size of the state is $2m + 4$. Action is the amount of discharge of EV about n .

Action contains information on the number and amount of discharge of the EV to be discharged. The output of the action is expressed as Equation (2) below.

$$a_t = nm + D_t^n \quad (2)$$

a_t means action in time t . D_t^n is the amount of discharge of EVs for n in time t . In this system, action is expressed as a one-dimensional vector. According to the above equation, each EV has a specific value range and selects a discharge amount within the range of that value.

Reward is expressed as Equation 3.

$$r_t = \beta^{ESS} r_t^{ESS} - \beta^{IL} r_t^{IL} - \beta^{EV} r_t^{EV} - r^{Choice} \quad (3)$$

r_t^{ESS} is the ESS SoC Reward and is the amount of power discharged by the EV to maintain optimal power in the ESS. r_t^{IL} is IL Reward and the shifted amount of power. r_t^{EV} is EV Reward and is the amount that does not meet the required SoC at the time each EV leaves. r^{Choice} is a wrong selection Reward and returns a specific constant if the selection is wrong. β^{ESS} , β^{IL} , and β^{EV} are coefficients for setting weights for ESS SoC, IL, and EV, respectively. Our goal is to optimize ESS SoC while satisfying user satisfaction.

The user's satisfaction is the amount of shifted power and the amount of power that does not meet the SoC required by the user when the EV leaves.

C. System Architecture that RL

The overall system structure is shown in Figure 1 below.

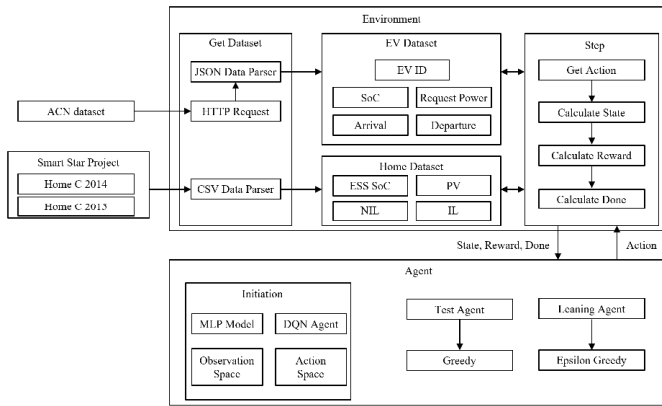


Figure 1. Overall System Architecture

The data to be collected are EV, power consumption, and PV. PV and power usage data are collected from Smart Start Projects [25]. It is collected in the form of CSV and parsed in the form of learning. EV data is collected from the ACN-Data API [26]. Data is requested in the form of HTTP Request and received in JSON format. Likewise, data is parsed in the form of data for learning. The collected data is managed by Environment. The step of Environment means a unit of time. In the step function, the Environment receives an action from the agent, and each time the step is performed, the state is updated, calculating and returning Reward and Done. We utilize the DQN Agent. The agent initializes Observation,

Action Space, MLP Model, and Agent, respectively. There are two functions of an agent: learning and test. Learning Agent uses the Epsilon Greedy method as an action selection method as an agent for learning. Test Agent is an agent that conducts experiments through learned results, and it uses the Greedy method to select the optimal action.

IV. SIMULATION

A. Simulation Setting

The specification of HW and SW for simulation can be seen in Table 2 below.

TABLE 2. SPECIFICATION

Type	Description
CPU	Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz
OS	Ubuntu 20.0.05 LTS
API	Pytorch 1.7.1
	OpenAI Gym
	MARL Framework [27]

We use the Pytorch API to use NN. We also use the box format of the OpenAI Gym API to represent observation and action space. The DRL is implemented using the DRL agent of the MARL framework.

Data collected by the Smart Star Project is classified as shown in Table 3 below.

TABLE 3. GENERATED / DEMAND POWER DATASET

List	Group
Photovoltaic	RG
Dishwasher	IL
Furnace 1	IL
Furnace 2	IL
Home office	NIL
Fridge	NIL
Wine cellar	NIL
Garage door	IL
Kitchen 12	NIL
Kitchen 14	NIL
Kitchen 38	NIL
Barn	NIL
Well	IL
Microwave	NIL
Living room	IL

Data collected by the Smart Star Project means PV and load data. The hourly unit is stored every 30 minutes.

Data collected from ACN-data is classified as shown in Table 4 below.

TABLE 4. EV DATASET

Field	Description
_id	Unique identifier of the session record
chargingCurrent	Time series of the measured current draw of the EVSE during the session
connectionTime	Time when the EV plugged in
disconnectionTime	Time when the EV unplugged
doneChargingTime	Time when of the last non-zero current draw recorded
kWhRequested	Energy requested by the user in kWh

The data settings of ESS and EV are shown in Table 5 below.

TABLE 5. ESS, EV SETTING

Field	Value
Maximum ESS SoC	20kW
Optimal ESS SoC	16kW (80%)
Minimum ESS SoC	2kW (10%)
EV ID	743, 562, 1083, 560, 569, 891

The environment parameter settings are shown in Table 6 below.

TABLE 6. ENVIRONMENT PARAMETERS SETTING

Field	Value
Observation space size	16-dimension, -inf to +inf (Continuous)
Action space size	30 (Discrete)
Start / end date of PV and load data	2014.10.01 / 2014.12.31
Start / end date of EV data	2018.10.01 / 2018.12.31
Maximum step count	4320
β^{ESS}	200
β^{IL}	0.1
β^{EV}	0.5
r^{Choice}	10 (incorrect choice) 0 (correct choice)

There are total of six vehicles, and the dimension size of the observation space is 16. The range of each dimension was set to -infinite to +infinite. The discharge amount range of the vehicle was set to 5 and the discharge amount unit was set to 1. Therefore, the size of the action space is 30. The year was set differently because the data ranges of the data of the Smart Start Project and ACN-data did not match exactly. It is assumed that the data of the EV is used as a pattern.

The parameter setting of the agent is shown in Table 7 below.

TABLE 7. AGENT PARAMETERS SETTING

Field	Value
Gamma	0.9
Learning rate	0.01
Epsilon	0.1

Figure 2 shows the powers of demand and PV.

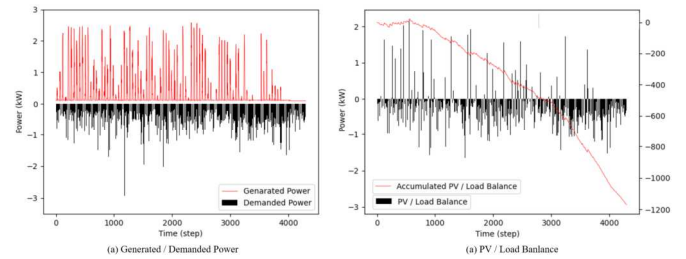


Figure 2. Power Change per Steps

Figure 2(a) shows the power produced by PV and the total power required by the user. Figure 2(b) shows the pv/load balance per step.

The system repeats the following process. The learning agent based on Epsilon greedy is performed in 1 episode. The test agent based on greedy is performed 1 episode. The update and storage of the policy are performed after the learning agent, and the result is stored after the test agent.

B. Simulation Result

The results for Reward per Episode can be seen in Figure 3.

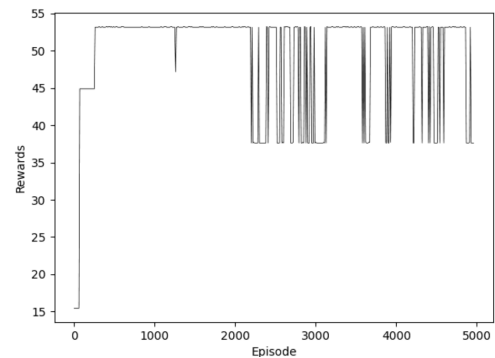


Figure 3. Rewards per Episodes

Reward increases sequentially from the initial 0 to about 15, 45, and 53. Reward value stays at 53 until the episode is about 2000. At the time when the episode is about 2000, the Reward value repeats between about 53 and 37. The reason for this is explained after checking the other results.

The results of ESS SoC per step for each Episode can be seen in Figure 4.

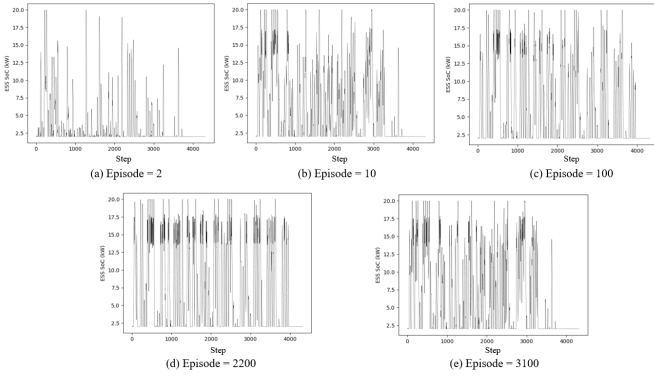


Figure 4. ESS SoC per Steps

In the figure above, it can be seen that the higher the Reward, the closer the ESS SoC becomes to the optimal ESS SoC. In the part where the reward decreases after Episode 2000, ESS SoC decreases.

The shifted load per step for each Episode can be seen in Figure 5.

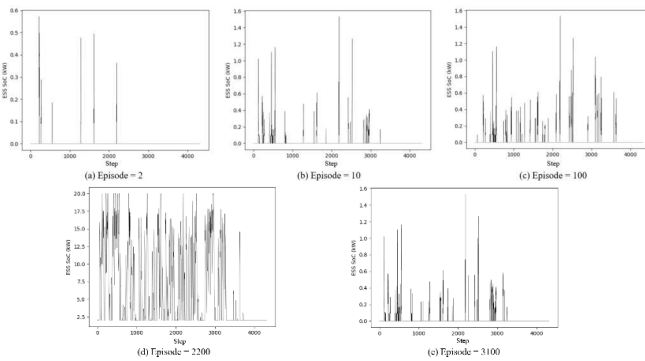


Figure 5. Shifted Load per Steps

Shifted load is the amount of power shifted by IL. It needs to be minimized because the user does not receive as much power as the shifted amount of power as the user needs to receive.

Detailed figures for Reward per Episode can be seen in Table 8.

TABLE 8. RESULTS PER REWARDS

Episode	Reward	Mean of ESS SoC	Total Shifted Load
2	14.3632	3.272018	12.737123

10	34.5507	5.726292	56.363631
100	44.8730	6.504482	94.413078
2200	53.1467	7.200487	139.036705
3100	37.6167	6.334712	75.574098

At 2200 Episode, the values of Reward and Mean of ESS SoC are maximized. Likewise, the value of Total Shift Load is also the maximum. However, the less the shifted load has, the higher the user's satisfaction increases. Therefore, in episode 2200, ESS SoC was optimized, but user satisfaction decreased. This is because the value of β^{ESS} was set larger than β^{IL} . So ESS SoC was given a greater score for optimization than IL when calculating the reward.

On the other hand, when Episode is 100, Mean of ESS SoC is 6.504 and Total Shift Load is 94.413. And when Episode is 3100, Mean of ESS SoC is 6.334, and Total Shift Load is 75.574. When comparing Episode 100 and Episode 3100, ESS SoC was 0.974 times, while Total Shift Load was 0.8 times. The greater Total Shifted Load reduction than the Mean of ESS SoC reduction means that we have found an optimal policy to optimize ESS SoC while minimizing IL.

Figure 6 shows the change for EV SoC in episodes 2200 and 3100.

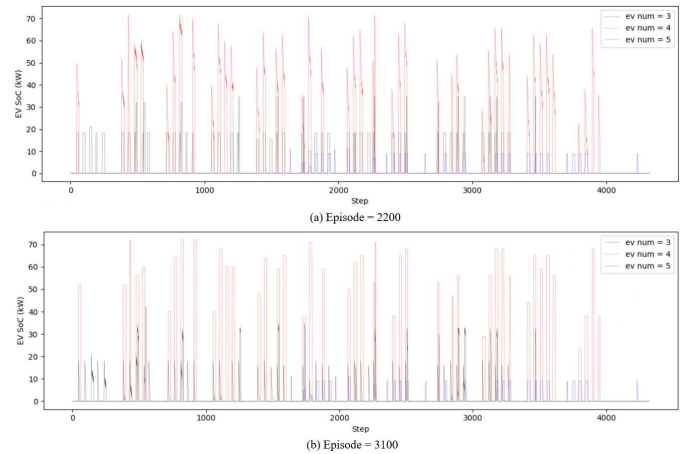


Figure 6. EV SoC per Steps

In the figure 6, the start of one bar is the time when the EV arrives, and the end is the time when the EV leaves. As can be seen from the figure, the SoC reduction when the EV departs is less in episode 2200 than in episode 3100.

Episode 2100 is an interval in which the previously described reward has a value between 53 and 37. To infer this reason, the agent prioritizes the action that optimizes ESS SoC because the value of β^{ESS} is the largest in the simulation setting. Afterward, with Reward persisted to a specific value, the agent updates the q-value while exploring other actions by the Epsilon Greedy. Then, when Episode reaches about 2100, it can be seen that Agent has found a way to optimize ESS SoC, shifted load, and EV SoC.

V. CONCLUSIONS AND FURTHER STUDY

We proposed a DR method for RL-based ESS SoC. To this end, state, action, and reward were formalized and the environment was designed. The goal of learning is to optimize ESS SoC, minimize shifted load, and minimize SoC reduction at the time EV leaves. In the simulation results, we found the point where ESS SoC is maximized. In addition, we found a policy in which ESS SoC, IL, and EV SoC are all optimized.

However, there are several things to supplement in the proposed method. First, because one agent handles multiple EVs, the state dimension and action space increase as the number of EVs increases. Since the Q-value consists of a pair of state and action, as the dimension of the state increases, the number of calculations required for learning increases exponentially. The increase in action space also increases the number of calculations. Therefore, there is a need for a plan to establish an environment in which multiple agents cooperate using Multi-Agent RL (MARL). In addition, we entered only information on ESS, power, and EV into State. If weather is added to the state, it is expected to be a higher quality learning. The reason for not adding this is that the state dimension in the current simulation was high enough.

ACKNOWLEDGMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(2020R1A6A1A12047945)

* Corresponding Author: S.G. Choi(choisg@cbnu.ac.kr)

REFERENCES

- [1] A. Ghosal; M. Conti, "Key Management Systems for Smart Grid Advanced Metering Infrastructure: A Survey", IEEE Communications Surveys & Tutorials, vol. 21, issue. 3, 28 March 2019
- [2] S. Ahmed; T. M. Gondal; M. Adil; S. Ahmad Malik; R. Qureshi, "A Survey on Communication Technologies in Smart Grid", 2019 IEEE PES GTD Grand International Conference and Exposition Asia (GTD Asia), 19-23 March 2019
- [3] K. Ogimoto, "Power system in low-carbon society", Journal of the Institute of Electrical Engineers of Japan, Vol.129, No.1, pp.16-19, 2009 (in Japanese)
- [4] A. Mehrtash; P. Wang; L. Goel, "Reliability Evaluation of Power Systems Considering Restructuring and Renewable Generators", IEEE Transactions on Power Systems, vol. 27, issue. 1, Feb. 2012
- [5] Y. Sun.; H. Yue; J. Zhang; C. Booth, "Minimization of Residential Energy Cost Considering Energy Storage System and EV With Driving Usage Probabilities", IEEE Transactions on Sustainable Energy, vol. 10, issue. 4, Oct. 2019
- [6] S. Kim; M. Kwon; S. Choi, "Operation and Control Strategy of a New Hybrid ESS-UPS System", IEEE Transactions on Power Electronics, vol. 33, issue. 6, June. 2018
- [7] X. Song; Y. Zhao; J. Zhou; Z. Weng, "Reliability Varying Characteristics of PV-ESS-Based Standalone Microgrid", IEEE Access, vol. 7, pp. 120872 - 120883, 26. August. 2019
- [8] K. Ma, G. Hu and C. J. Spanos, "A cooperative demand response scheme using punishment mechanism and application to industrial refrigerated warehouses", IEEE Trans. Ind. Informat., 2015.
- [9] T. J. Lui, W. Stirling and H. O. Marcy, "Get smart", IEEE Power Energy Mag., vol. 8, no. 3, pp. 66-78, May/Jun. 2010.
- [10] [S. Majumdar; D. Chattopadhyay; J. Parikh, Interruptible load management using optimal power flow analysis, IEEE Transactions on Power Systems, vol. 11, issue. 2, May. 1996
- [11] W. Su, R. H. Eich; W. Zeng; M. Y. Chow, "A Survey on the Electrification of Transportation in a Smart Grid Environment", IEEE Transactions on Industrial Informatics, vol. 8, issue. 1, Feb. 2012
- [12] D. Maly; K. Kwan, "Optimal Battery Energy Storage System (BESS) charge scheduling with dynamic programming", Proc. Inst. Elect. Eng. Sci. Meas. Technol., vol. 142, no. 6, pp. 453-458, Nov. 1995
- [13] D. Liu; Y. Xu; Q. Wei; X. Liu, "Residential energy scheduling for variable weather solar energy based on adaptive dynamic programming", IEEE/CAA Journal of Automatica Sinica vol. 5, no.1, pp.36-46, Jan. 2018
- [14] [N. E. Lee; W. W. Kim; J.-h. Shin; J. O. Kim, "Optimal SOC limit estimation of ESS considering renewable energy sources", 2014 International Conference on Information Science, Electronics and Electrical Engineering, 26-28 April 2014
- [15] S. D. Whitehead, "A complexity analysis of cooperative mechanisms in reinforcement learning," in AAAI, 1991, pp. 607-613.
- [16] J. Rafati, D. Noelle, "Learning Representations in Model-Free Hierarchical Reinforcement Learning", Learning Representations in Reinforcement Learning, October 2018
- [17] V. Mnih; K. Kavukcuoglu; D. Silver; A. Graves; I. Antonoglou; D. Wierstra; M. Riedmiller, "Playing Atari with Deep Reinforcement Learning", 2013 Neural Information Processing Systems
- [18] Y. H. Park; S. H. Lee; S. G. Choi, "ESS SoC Optimization System Using EV Control", 2020 22nd International Conference on Advanced Communication Technology (ICACTION), 16-19 Feb. 2020
- [19] A. Shojaeighadikolaei; A. Ghasemi; K. R. Jones; A. G. Bardas; M. Hashemi; R. Ahmadi, "Demand Responsive Dynamic Pricing Framework for Prosumer Dominated Microgrids using Multiagent Reinforcement Learning", 2020 52nd North American Power Symposium (NAPS), 11-13 April 2021
- [20] Z. Wen; D. O'Neill; H. Mazi, "Optimal Demand Response Using Device-Based Reinforcement Learning", IEEE Transactions on Smart Grid, vol. 6, issue. 5, Sept. 2015.
- [21] F. Kiaee, "Integration of Electric Vehicles in Smart Grid using Deep Reinforcement Learning", 2020 11th International Conference on Information and Knowledge Technology (IKT), 22-23 Dec. 2020
- [22] Otterlo, M.; Wiering, M. (2012). Reinforcement learning and markov decision processes. Reinforcement Learning. Adaptation, Learning, and Optimization. 12. pp. 3-42. doi:10.1007/978-3-642-27645-3_1. ISBN 978-3-642-27644-6.
- [23] N. Maudet, A. Beynier, P. Weng, "Markovian sequential decision-making in non-stationary environments: application to argumentative debates", HAL, November 2015
- [24] Thomas Simonini, "Diving deeper into Reinforcement Learning with Q-Learning", freecodecamp, Apr. 2018.
- [25] Smart Stat Project, (2019) The UMass website [Online]. Available: <http://traces.cs.umass.edu/index.php/Smart/SmartCAN>
- [26] CAN-Data, Adaptive Charging Network, Available: <https://ev.caltech.edu/dataset>
- [27] marl - Multi-agent reinforcement learning framework, Github, Available: <https://github.com/blavad/marl>



Yong Hee Park received B.S. degree in the College of Information & Communication Engineering, Chungbuk National University, Korea in 2019. He is currently a M.S. candidate in School of Information & Communication Engineering, Chungbuk National University. His research interests include Autonomous Vehicle, AI, Smart Grid.



Seong Gon Choi received B.S. degree in Electronics Engineering from Kyeongbuk National University in 1990, and M.S. and Ph.D. degree from Information Communications University, Korea in 1999 and 2004, respectively. He is currently an associate professor in College of Electrical & Computer Engineering, Chungbuk National University. His research interests include Autonomous Vehicle, smart grid, IoT, mobile communication, high-speed network architecture and protocol.