# Application of Deep Reinforcement Learning for Multi-objective and Continuous Control of Reactor Coolant

Junyong Bae and Seung Jun Lee
*Ulsan National Institute of Science and Technology, 50 UNIST-gil, Ulju-gun, Ulsan, 44919, Republic of Korea*
*junyong8090@unist.ac.kr, sjlee420@unist.ac.kr*

## 1. Introduction

Nuclear power plants (NPPs) widely employ automated systems to improve efficiency and safety. For instance, actuation signals of safety systems are automatically generated by a reactor protection system when the monitored parameters reach predefined thresholds. Proportional-integral-differential controllers (PID) controllers or controllers that combine two out of three types of controllers (e.g., proportional-integral controllers) output proper device status (e.g., valve position) to achieve the desired state of the system [1].

Although these systems cover many tasks, most tasks are still performed by operators in the main control room (MCR). Especially, the automation level is relatively low when the plant state is dynamically changing, such as during startup and shutdown operations. During these operations, operators are required to simultaneously adjust the pressure, level, and temperature of the reactor coolant. This task is a complex and mentally taxing activity since it is a multi-objective and continuous decision-making task. Automation through rule-based and PID controllers is only applied for adjusting each target parameter independently and can be disabled according to a plant state.

Recently, deep reinforcement learning (DRL) is automating complex, human-level tasks such as the game of Go and StarCraft unit control [2-4]. With the premise of these successes, research has been conducted to utilize DRL to automate tasks in the nuclear industry. Radaideh et. al utilized deep Q-learning, a DRL that trains a deep learning model to predict the expected discounted future reward (i.e., Q function) of each action for a given state, in order to optimize nuclear assembly design [5]. Kim et. al proposed an autonomous operating framework that can simultaneously combine DRL-based and rule-based automation for startup and shutdown operations [6]. Likewise, Lee et. al trained an agent that increases reactor power using an asynchronous advantage actor-critic (A3C) algorithm, which is a type of DRL specialized for multiple training environments [7]. They also trained reactor coolant pressure and level controllers for cold shutdown operation using a soft actor-critic (SAC) algorithm and prioritized experience replay (PER). SAC and PER can optimize the experience and the training data for controller training, respectively. They also compared the performances of DRL-based and PID-based controllers [1]. In an author's previous work, an autonomous pressure controller before bubble creation during start-up operation was trained by a deep Q-network [8].

Previous research has shown a possibility of a DRL, however, only applied a DRL for a single-objective task. This study applied a DRL for a multi-objective continuous task, that is control of pressure, volume, and temperature of reactor coolant. To this end, we implemented the SAC algorithm. The control agent was designed to output the appropriate status of five devices according to the target pressure, level, and temperature of the reactor coolant. Following this status, devices were continuously controlled by a proportional controller. As a result, a trained agent successfully controlled the status of the reactor coolant.

## 2. Soft Actor-Critic

Reinforcement learning is a process of optimizing an agent's action by using the experiences accumulated through the exploration of a given environment [9]. To explore the environment, the agent repeatably poses actions and receives the renewed state of the environment. Considering the state and the renewed one, the posed action is evaluated by a reward function and discounted summation of future rewards, as shown in Eq. (1), where $r$ is a discount factor, $R$ is a reward, and $G$ is the action value. If we have an approximate action value function $Q$, as shown in Eq. (2), Eq. (1) can be rewritten like Eq. (3). Since Eq. (3) gives a target $Q^*$, we can optimize the action value function $Q$. In DRL, a deep neural network is utilized as an approximating function of action value function $Q$.

$$G_t = R_{t+1} + \gamma R_{t+2} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \quad \text{Eq. (1)}$$

$$Q(s,a) = \mathbb{E}[G_t \mid S_t = s, A_t = a] \quad \text{Eq. (2)}$$

$$G_t = R_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k R_{t+k+2} \quad \text{Eq. (3)}$$
$$= R_{t+1} + \gamma G_{t+1} \approx R_{t+1} + \gamma \max_a Q(S_{t+1}, a) = Q^*(s,a)$$

DRL through optimizing the action value function has a limitation since the optimal action should be investigated by evaluating the value of every possible action. Actor-critic methods overcome this limitation by introducing two approximates. Actor-network approximates optimal action for a given state and the critic-network expects the value of action for a given state.

SAC is a DRL algorithm considering action entropy. Actor-network in SAC predicted optimal action

distribution and optimized its trainable parameter φ to minimize an objective function shown in Eq. (4). In this equation, $\alpha \log \pi_\varphi(a_t|s_t)$ represents an action entropy.

$$J_\pi(\varphi) = \mathbb{E}[[\alpha \log \pi_\varphi(a_t|s_t) - Q_\theta(s_t, a_t)]] \qquad \text{Eq. (4)}$$

If the action value (i.e., $Q_\theta(s_t, a_t)$) is expected to be a small value, the actor-network will be trained to maximize an action entropy (i.e., $-\alpha \log \pi_\varphi(a_t|s_t)$). It helps SAC agents to search an action space more meticulously when optimal action with a high action value is not founded.

## 3. Implementation

To verify the feasibility of DRL for multi-objective and continuous control of reactor coolant, we developed a training environment with an NPP simulator. A compact nuclear simulator (CNS), which is a simplified simulator that mimics the behavior of the Westinghouse 3-loop 1000MWe plant, was modified to build multiple training environments.

The SAC algorithm was implemented to train an agent that controls the pressure and temperature of reactor coolant and the level of pressurizer under cold-shutdown conditions. We allowed the agent to control five components: charging flow control valve (FV122), letdown flow control valve (HV142), residual heat removal system flow control valve (HV603), pressurized spray flow control valve, and proportional heater. In other words, the agent outputs the positions of four valves and heater power. Every 60 sec in simulator time, the agent generates the positions and power, and components are controlled by a proportional controller continuously.

The inputs to the agent were the current values, deviations from the target values, variations during 60 sec, and the target values of pressure, level, and temperature, respectively. These values were normalized between 0 to 1.

Each episode started from the initial conditions that are after bubble creation during cold-shutdown to hot-shutdown operation. The target pressure, level, and temperature are randomly selected between 20 ~ 35 *kg/cm*$^2$, 20 ~ 60%, and 110 ~ 170 ℃. The episode stopped when the pressurizer level reaches 99% or 17% or the pressure and temperature of reactor coolant infringe PT curve limitation. The maximum length of the episode is 21,600 sec (i.e., 6 hr).

We constructed a reward function with three success rewards. If the pressure of the reactor coolant remains within $[Pressure\ target \pm 1.0\ kg/cm^2]$, a pressure reward is +10. Likewise, the temperature reward and level reward are +10 when the temperature and level stay in $[Temperature\ target \pm 3\ K]$ and $[Level\ target \pm 2.0\ \%]$, respectively.

## 4. Results and Discussion

Figures 1 and 2 show a trend of rewards for each training episode. After experiencing more than 300 episodes, the rewards for pressure and level increased, as shown in Fig. 2. On the other hand, the temperature reward increased only after experiencing more than 4,000 episodes, as shown in Fig. 1. The total reward, that is a summation of these three rewards, scored a high value after 4,000 episodes and was maintained until the end of training.

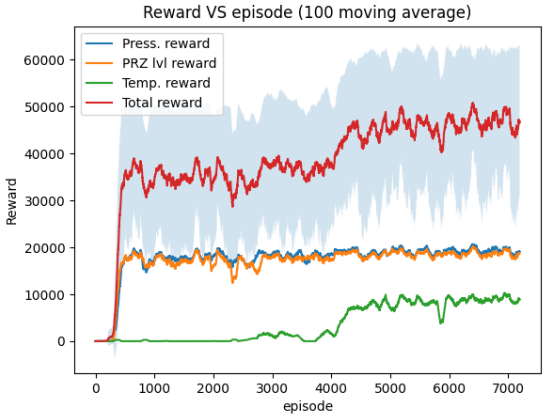


Fig. 1. Reward for each training episode.

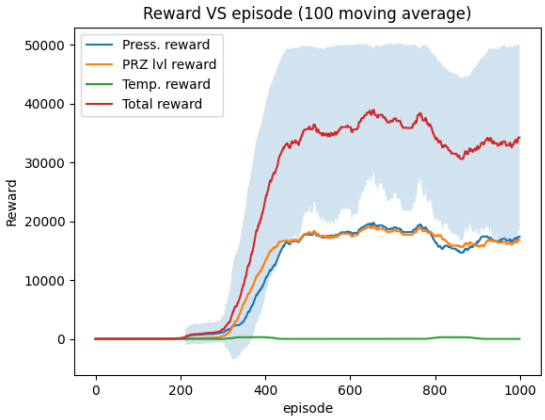


Fig. 2. Reward from 1st to 1000th episode.

Figures 3 and 4 illustrate the control results for the agent after experiencing more than 7,100 episodes. In the case of Fig. 3, the agent is required to increase the level and pressure while increasing the temperature. For this, the agent fully opened FV122 and fully closed the HV142. At the same time, the power of the proportional heater became almost 100 % and HV603 was fully closed to increase temperature. After the level and pressure reached the target range, the agent stabilized them by continuously controlling FV122, HV142, and spray flow. It took more than 11,000 sec to adjust the temperature in

the target range. After that, HV603 was slightly opened and spray flow was increased to stabilize the temperature.

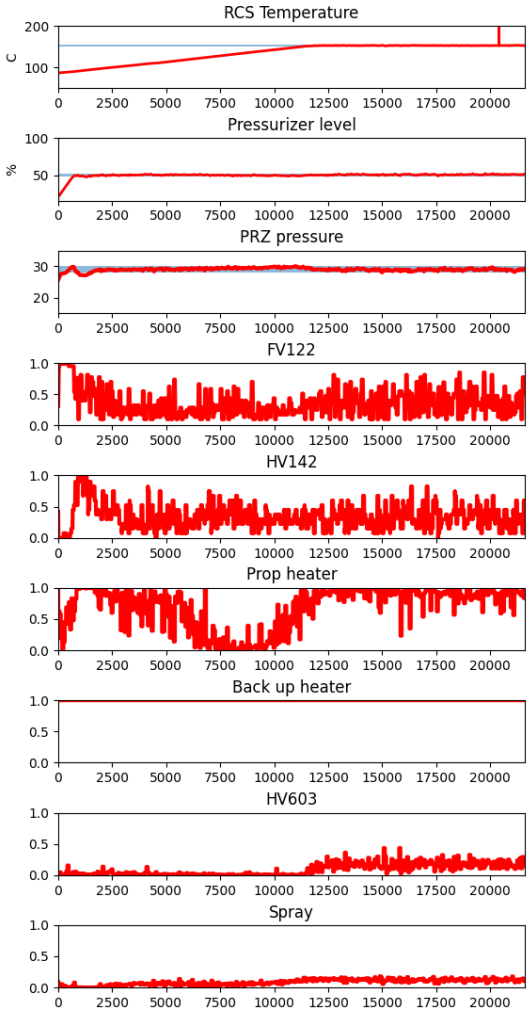


Fig. 3. Trend of target parameters and components states. Target pressure, level, and temperature are 29 kg/cm$^2$, 51%, and 153 ℃.

In the case of Fig. 4, the agent needed to decrease the pressure while increasing the level and temperature. To this end, spray flow became almost 50% and was closed when the pressure arrive in the target range. After 4,000 sec, the temperature reached the target range, and pressure and level were stabilized. Considering these control examples, the reason for the late increase in temperature reward shown in Fig. 1. Is that temperature control takes relatively longer compared to pressure and level.

As shown in Figures 3 and 4, the SAC algorithm successfully trained the agent that can control the pressure, level, and temperature of reactor coolant simultaneously. It means that DRL can address a multi-objective and continuous control problem.

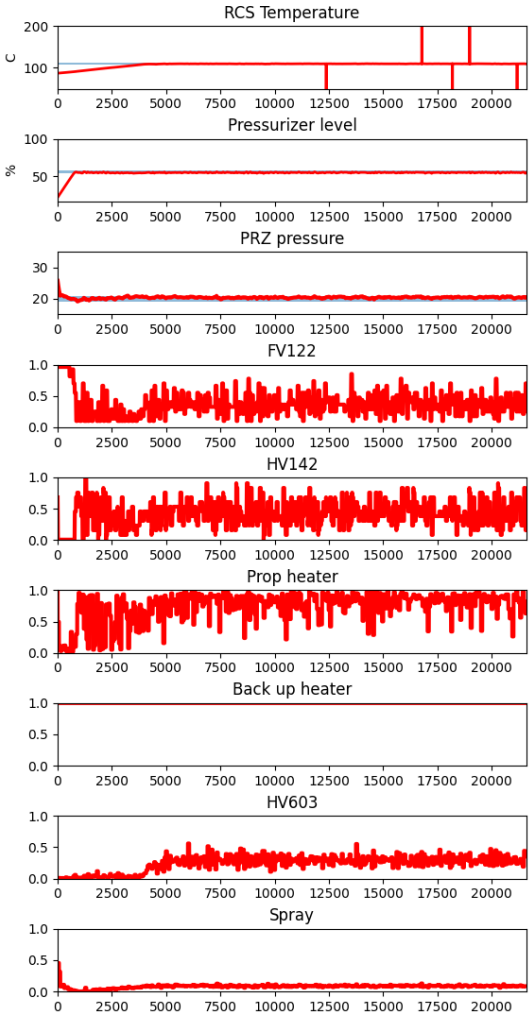


Fig. 4. Trend of target parameters and components states. Target pressure, level, and temperature are 20 kg/cm2, 56%, and 110 ℃

## 5. Conclusion

This research applied a SAC algorithm for multi-objective and continuous control task, that is control of the pressure, level, and temperature of reactor coolant. As a result, the agent trained by a SAC algorithm successfully adjusted and stabilized target parameters. Therefore, this research shows the possibility of DRL for automating the tasks that are more complex than the task automated by rule-based logic and PID controller.

## REFERENCES

[1] D. Lee, S. Koo, I. Jang, and J. Kim, Comparison of Deep Reinforcement Learning and PID Controllers for Automatic Cold Shutdown Operation, Energies, Vol. 15, No. 8, pp. 2834, 2022.

[2] Mnih, V., et al., Human-level control through deep reinforcement learning, Nature, Vol. 518, No. 7540, pp. 529-533, 2015.

[3] Silver, D., et al., Mastering the game of Go without human knowledge, Nature, Vol. 550, No. 7676, pp. 354-359, 2017.

[4] Vinyals, O., et al., Grandmaster level in StarCraft II using multi-agent reinforcement learning, Nature, Vol. 575, No. 7782, pp. 350-354, 2019.
[5] Radaideh, M.I., et al., Physics-informed reinforcement learning optimization of nuclear assembly design, Nuclear Engineering and Design, Vol. 372, pp. 110966, 2021.
[6] J. Kim, and S.J. Lee, Framework of two-level operation module for autonomous system of nuclear power plants during startup and shutdown operation, Transactions of the Korean Nuclear Society Autumn Meeting, 2019.
[7] D. Lee, A.M. Arigi, and J. Kim, Algorithm for autonomous power-increase operation using deep reinforcement learning and a rule-based system, IEEE Access, Vol. 8, pp. 196727-196746, 2020.
[8] J. Bae, J. Kim, and S.J. Lee, An Autonomous Pressure Controller based on Approximation of Action Value Function, Transactions of the Korean Nuclear Society Autumn Meeting, 2020.
[9] Sutton, R.S. and A.G. Barto, Reinforcement learning: An introduction, MIT Press, 2018.
pp.612-613, 1999.