An Autonomous Pressure Controller based on Approximation of Action Value Function

Junyong Bae, Jae Min Kim, 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, jaemink@unist.ac.kr, sjlee420@unist.ac.kr**

1. Introduction

Start-up operation is a key operational mode of nuclear power plants (NPPs). In general, operators in a main control room raise the reactor power to 100% following start-up operating procedures. The procedures instruct the operators to check important instrumentation values, establish a component control plan, manipulate devices, and perceive precautions. Since the operators should process the aforementioned tasks concurrently, start-up operation is a complex and mentally taxing activity. For instance, the operators need to adjust charging and letdown flows to maintain pressure and pressurizer water level in allowable range while actuating reactor coolant pump and heater for bubble creation. One way to reduce the burden of the operators is partial automation and especially, automation of continuously performed tasks.

Devices and systems have been partially automated in several operational modes of NPPs. However, dynamically changing plant status in start-up operation impedes the adaption of automation Since rule-based automation has limits when the empirical relationship between variables is unclear and too many possible conditions exist, the heuristic approach may be appropriate for partial automation of start-up operation. In recent years, reinforcement learning, which is heuristic data-based automation technology, has advanced and achieved human-level automation in various applications like Atari games, the game of Go, and Starcraft unit micro-control. [1][2][3].

This research employed deep Q-network (DQN), which is a reinforcement learning algorithm based on approximation of action-value function to develop an autonomous pressure controller in a start-up operation. We composed the controller of valve controllers and an organizer and applied to a compact nuclear simulator (CNS) that is a simplified NPP simulator that mimics the Westinghouse 3-loop plant. The developed controller with DQNs successfully figured out the optimal control strategy to maintain the pressure in the target range without any prior knowledge.

2. Action Value Function Approximation with Neural Network

2.1. Background of reinforcement learning

Reinforcement learning is an optimization process where an agent of reinforcement learning explores the given environment without any explanation and exploits their experiences to optimize actions [4]. In the exploration process, agent repeatably takes actions and receive a changed state of the environment and a reward of the state. The reward is a predefined value function of the state. Repeating the exploration, the agent accumulates the memory composed of the state before action, the chosen action, the state after the action, and the reward. An optimization process utilizes the memory to fine-tunes the agent to the direction of maximizing the future reward summation. In recent years, advanced reinforcement learning algorithms have been introduced with flexible agent designs, optimization direction-finding algorithms, experiences exploiting methods.

Approximation of action-value function is a reinforcement learning algorithm for the agent to estimate the value of possible action in a given state [5]. If the agent can evaluate the values of actions for the current state, actions can be optimized by selecting the worthiest action every time. In general, a approximation target of action-value function is the total discounted reward, G_t, in Eq. (1) where R_t is the reward at a time step, t, and γ is a discount factor. Therefore, the action-value function, $Q(s, a)$, is defined as shown in Eq. (2). Since G_t needs future reward up to the end and the actions up to the end maybe not optimal action, target $Q^*(s, a)$ in Eq. (3) based on the Bellman equation has been widely implemented instead of G_t. The simple approach is a tabular method where the value of combinations of actions and states are listed in the table. The values in the table are repetitively updated by the exploration and exploitation.

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

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

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

2.2. Deep Q-network

Since classical Q-learning including the tabular method can only address the environment with the discretized state space, it is limitedly applicable to real-world problems. Deep Q-network (DQN) method implements a deep neural network to approximate action-value function to address continuous state space [1]. For the given state, Q-network with the deep neural network estimates the action-value of possible action candidates, and the action with the highest action-value is chosen as output. This method inherits the strong points that deep neural network has. Additionally, DQN utilizes a replay buffer and a target network to exploit the experience efficiently and stably. Since the agents randomly selects the batch data from the replay buffer where experiences have been accumulated, a bias in training data can be removed. A loss function of Q-networks is Eq. (4) and adjustable parameters of the network are updated to minimized the loss. The update of parameters can be unstable because the target *Q*(s, a)* is affected by the Q-network itself as shown in Eq. (3). To address this problem, DQN introduced the target network that is used to calculate target *Q*(s, a)* and updated only at the end of batch training.

$$Loss = [Q^*(s,a) - Q(s,a)]^2 \qquad \text{Eq. (4)}$$

3. Development of Autonomous Pressure Controller

Two valves control the pressure of reactor coolant before bubble creation on start-up operation in CNS. FV122 is a charging flow control valve, and HV142 is a control valve for a flow from residual heat removal system to chemical and volume control system. We trained a valve controller with DQN for each valve independently. Because independently trained valve controllers do not cooperate, we added an organizer DQN that regulates a control signals from the valve DQNs. Fig. 1. shows the design of an autonomous pressure controller with three DQNs.

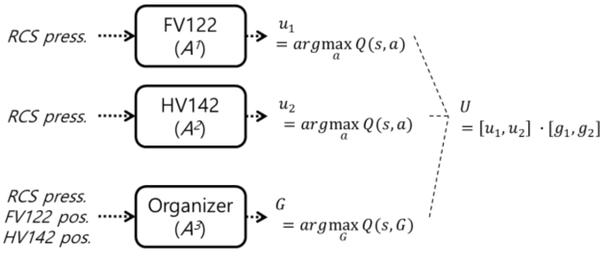

RCS press.
FV122
(A^1)
$u_1 = \underset{a}{argmax}\, Q(s,a)$
RCS press.
HV142
(A^2)
$u_2 = \underset{a}{argmax}\, Q(s,a)$
$U = [u_1, u_2] \cdot [g_1, g_2]$
RCS press.
FV122 pos.
HV142 pos.
Organizer
(A^3)
$G = \underset{G}{argmax}\, Q(s,G)$

Fig. 1. Overall structure of autonomous pressure controller. Three DQN were implemented and the organizer regulates the control signal from valve agents.

3.1. Valve controller design

Human-level controls of FV122 and HV142 are open and close of the valve. We set the valve open and close for 1 s respectively as a candidate action of the valve controller. The real-time input of the controller is a time-series record of reactor coolant system pressure. To give a trend information of pressure, we constructed the Q-network with long short-term memory, which has its strength in time-series data handling [6].

3.2. Organizer for cooperation of valve controllers

Because the action of one controller can distort the control of another controller, cooperation is essential in the multi-agent setting. We added the organizer to force the cooperation of two valve controllers. The organizer decides whether the control signal from each valve controller is implemented or not. Since the input of the organizer is time-series data of reactor coolant system pressure, FV122 valve position, and HV142 valve position, an organizer DQN also comprises long short-term memory cells. Output of the organizer DQN is the action-value of possible four permission signal cases (i.e. no permission, the permission of FV122 control only, permission of HV142 control only, and permission of FV122 and HV142 control). In the training stage, the organizer DQN employs a trained FV122 and HV142 controllers.

3.3. Reward design

The reward is the guidance of DQN training. In this application, we divided into three cases according to the pressure. If the pressure overcomes a boundary, the agent gets -1,000 points as a reward and an episode finish. To avoid -1,000 points, the agent should keep the pressure in the allowable boundary up to 1,800 s. When the pressure reaches a target range, 2 points are given at every time step. Fig. 2. details the reward design.

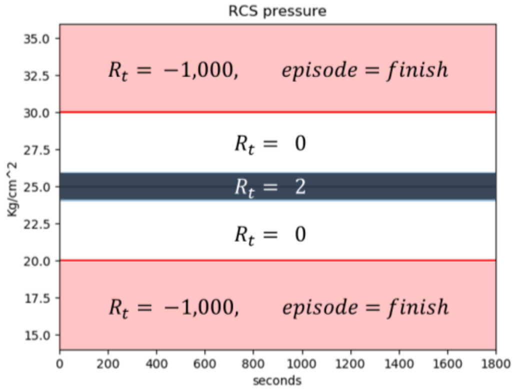
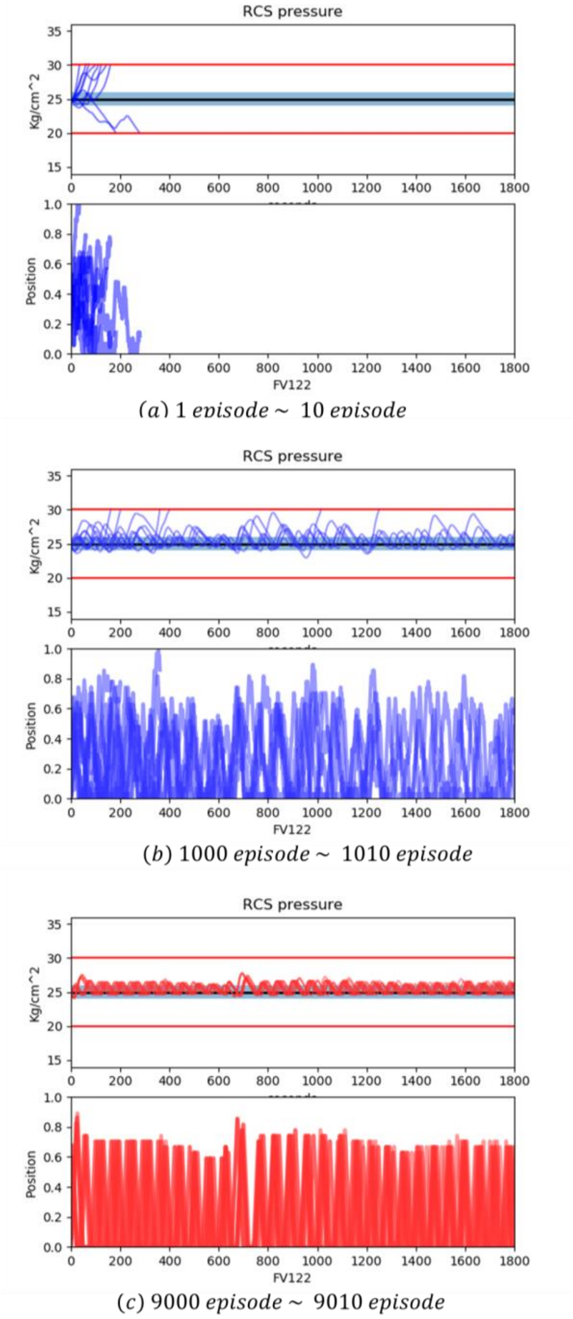


Fig. 2. Reward design

4. Result

4.1. Valve controllers

We made the FV122 and HV142 controllers to explore the 10,000 episodes with a decaying probability of random action selection. When we trained the FV122 controller, the HV142 valve was stuck and vice versa. Fig. 3. shows the progress of FV122 DQN. The FV122 controller initially selected random action as shown in Fig. 3. (a). When the DQN experienced 1,000 episodes, the pressure starts to visit the target range, however, some episodes failed (i.e. overcome the boundary) and a fluctuation of the pressure is significant as shown in Fig. 3. (b). From the 9,000th episode, the DQN converged and mostly located the pressure in the target range.



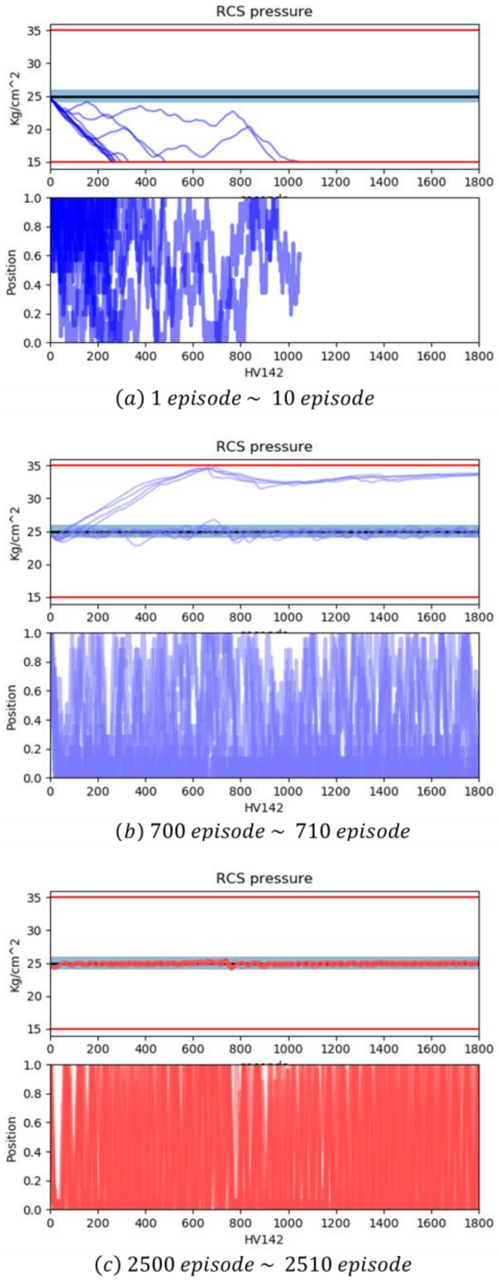
(a) 1 *episode* ~ 10 *episode*

(b) 1000 *episode* ~ 1010 *episode*

(c) 9000 *episode* ~ 9010 *episode*

Fig. 3. Progress of FV122 DQN.

For HV142 DQN training, we expanded the allowable pressure boundary comping with FV122 training. Fig. 4. illustrates the training result of the HV142 controller DQN. Like the FV122 DQN, the valve position was randomly adjusted as shown in Fig. 4. (a). Up to the 700th episode, HV145 agents continuously failed to reach the target boundary, however, it started to locate the pressure in target range near the 700th episode as illustrated in Fig. 4. (b). From the 710th episode, HV142 DQN rapidly converged. Fig. 4. (c) shows that HV142 DQN successfully figured out the optimal action and settled the pressure down in the target range.



(a) 1 *episode* ~ 10 *episode*

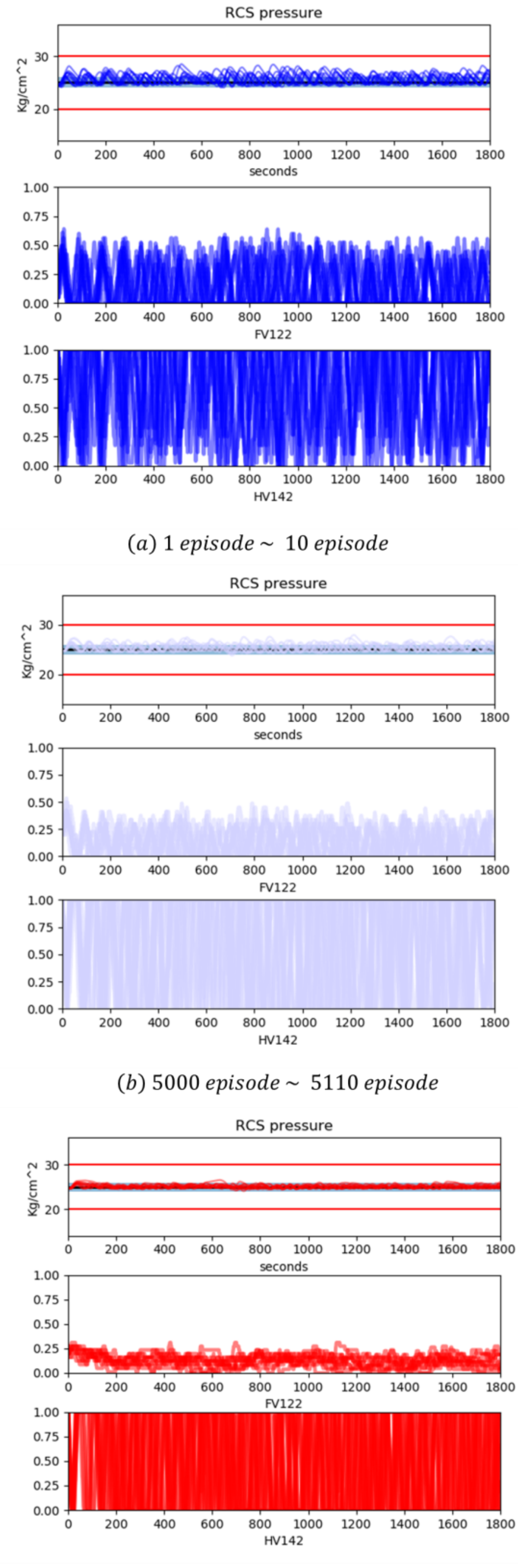
(b) 700 *episode* ~ 710 *episode*

(c) 2500 *episode* ~ 2510 *episode*

Fig. 4. Progress of HV142 DQN.

4.2. Organizer

With trained FV122 and HV142 agents shown in Fig. 3. and Fig. 4., organizer DQN practiced a pressure control 30,000 times. Fig. 5. shows the progress of control when the organizer exploited the 10 episodes, 5,110 episodes, and 20,010 episodes. Finally, the organizer understood the policy that mostly permits the control signal from HV142 controllers and rejects the signal from FV122 agents. The policy adjusts the position of FV122 to around 0.1 and mainly utilizes the HV142 controller to maintain the pressure in the target range. The policy coincides with the way of human operator.



(*a*) 1 *episode* ~ 10 *episode*

(*b*) 5000 *episode* ~ 5110 *episode*

(*c*) 20000 *episode* ~ 20010 *episode*

Fig. 5. Progress of organizer DQN.

Fig. 6. illustrates the evolution of episode reward. In an early part of the training, the organizer did not promote the cooperation of the valve controllers and scored relatively low episode reward. When the organizer experienced 12,000 episodes, the cooperation was achieved and started to score 3,600 points, which is the maximum score.

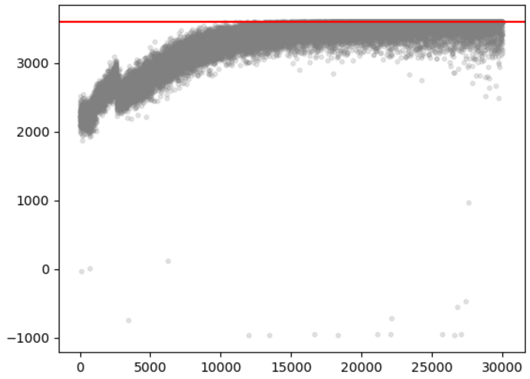


Fig. 6. Episode reward for every training trials of organizer DQN.

5. Conclusion

We applied the DQN, which is a reinforcement learning algorithm, to maintain the pressure at the targe range on start-up operation conditions. DQN for two valves successfully converged to optimal control actions. The organizer DQN also found the policy of cooperation and success to maintain the pressure in the target range. As an application of reinforcement learning on an NPP, this research shows the possibility that the DQN may be utilized to partial automation of NPP control without any prior knowledge about the control. However, since this research addressed a monotonous task and the simplified simulator, further investigation and feasibility tests for more complicated tasks and situations are necessary.

ACKNOWLEDGEMENT

This work was supported by the Korea Institute of Energy Technology Evaluation and Planning (KETEP) and the Ministry of Trade, Industry & Energy (MOTIE) of the Republic of Korea (No. 20171510102040).

REFERENCES

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

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

[3] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, Counterfactual multi-agent policy gradients, arXiv preprint arXiv:1705.08926, 2017.

[4] R. S. Sutton and A. G. Barto, Reinforcement learning: An introduction, MIT Press, 2018.

[5] C. J. Watkins and P. Dayan, Q-learning, Machine learning, Vol. 8, No. 3-4, pp. 279-292, 1992.

[6] J. Bae, J. Ahn, and S. J. Lee, Comparison of Multilayer Perceptron and Long Short-Term Memory for Plant Parameter Trend Prediction, Nuclear Technology, Vol. 206, No. 7, pp. 951-961, 2020.