2023년 2월

박사학위 논문

Development of Autonomous Operation Algorithm using Deep Reinforcement Learning for Start-up/Emergency in NPPs

조선대학교 대학원

원자력공학과

이 대 일

Development of Autonomous Operation Algorithm using Deep Reinforcement Learning for Start-up/Emergency in NPPs

원자력발전소의 기동/비상 운전을 위한
심층강화학습 기반 자율운전 알고리즘 개발

2023년 2월 24일

조선대학교 대학원

원자력공학과

이대일

Development of Autonomous Operation Algorithm using Deep Reinforcement Learning for Start-up/Emergency in NPPs

지도교수 김 종 현

이 논문을 공학 박사학위신청 논문으로 제출함

2022년 10월

조선대학교 대학원

원자력공학과

이대일

이대일의 박사학위논문을 인준함

위원장	조선대학교	교수	김진원 (인)
위 원	조선대학교	교수	나만균 (인)
위 원	조선대학교	교수	김종현 (인)
위 원	한국원자력연구원	연구원	성승환 (인)
위 원	한국원자력연구원	연구원	구서룡 (인)

2023년 1월

조선대학교 대학원

CONTENTS

List of Tables

List of Figures

초 록

원자력발전소의 기동/비상 운전을 위한 심층강화학습 기반 자율운전 알고리즘 개발

이 대 일
지도 교수 : 김 종 현
원자력공학과
조선대학교 대학원

최근 컴퓨터 성능의 향상과 새로운 인공지능 알고리즘의 등장으로 인공지능 기술에 기반한 높은 자동화 수준을 가진 자율 운전 시스템이 많은 산업분야에 적용되었다. 자율 운전 알고리즘은 원자력 발전소 시스템의 기존에 전통적인 자동화 알고리즘보다 더 높은 수준의 개념을 가지고 있다. 자율 운전 시스템을 개발하기 위해서는 이미 기존에 자동된 하위 시스템들을 모니터링, 제어 및 진단할 수 있는 기능을 포함해야 한다. 본 연구에서는 원전의 시동 및 비상시 자율 운전을 위한 지능형 제어기를 개발한다. 제어기는 현재 운전원에 의한 원자력 발전소에 운영 전략과 유사한 높은 수준의 작업을 수행하는 데 중점을 둔다. 운영자와 유사하게 구성 요소를 조작하기 위해 컨트롤러는 현재 수동 제어를 자동화하는 것을 목표로 한다. 설계 목표를 달성하기 위해 지능형 컨트롤러는 심층 강화학습 방법이 적용된다. 심층 강화학습 기반에 컨트롤러의 설계는 현재 운영 전략, 즉 기존 시스템, 운영 절차 및 인력을 고려하여 설계된다. 컨트롤러는 Westinghouse 990MWe, 3 Loop 가압경수로에 적용된다. 시동 및 비상 운전에서 검증 결과는 자율 운전 알고리즘이 주어진 운영 목표에 따라 원자력 발전소의 시스템들을 관리할 수 있음을 보였다.

Abstract

Development of Autonomous Operation Algorithm using Deep Reinforcement Learning for Start-up/Emergency in NPPs

Daeil Lee
Advisor : Prof. Jonghyun Kim, Ph.D.
Department of Nuclear Engineering
Graduate School of Chosun University

With the improvement of computer performance and the emergence of cutting-edge artificial intelligence (AI) algorithms, an autonomous operation based on AI is being applied to many industries. An autonomous algorithm is a higher-level concept than conventional automatic operation in nuclear power plants (NPPs). In order to achieve autonomous operation, the autonomous algorithm needs to include superior functions to monitor, control and diagnose automated subsystems. This study develops an intelligent controller for an autonomous operation in NPPs during start-up and emergency. The controller is focused on conducting high-level operations that are similarly performed to the current operation strategy. To manipulate components similarly to operators, controllers currently aim to automate manual controls. To achieve the design goal, the intelligent controller applies a deep reinforcement learning method. The design of the Deep Reinforcement Learning (DRL)-based controllers considers the current operational strategy, i.e., existing systems, operating procedures, and staffing. The controllers are applied to a reference NPP, a Westinghouse 990 megawatts electric, three-loop pressurized water reactor.

In start-up and emergency operation, the validation results showed that the autonomous operation algorithm can mange the NPPs according to given operational goals.

I. Introduction

A. Background

Nuclear power plants (NPPs) use highly automated controllers to reduce probability of accident risk and increase availability [1, 2]. In addition, digitalized controllers help process large amounts of data, improve system reliability, automate periodic tests, perform diagnosis, and increase operation capability [3]. Regulatory bodies for NPPs require that safety systems must be designed to be consisted as a high level of automation to protect public safety. This is because, in the event of an abnormal situation, these safety systems operate stably and quickly to ensure the safety of the public. Even if these safety systems are well designed, the operator must intervene if the system does not work under unexpected conditions [4].

Typical operations (i.e., Start-up/shutdown operation or emergency operation) in NPPs largely rely on the operator's manual controls, whereas the full power operation is highly automated. Thus, these operations are known to be error-prone for the following reasons [5, 6]:

- There are many operator's tasks that are need decision-making, such as establishing a operational strategy and planning operational goals according to guidelines from the operating procedures.
- Operator's many manipulations due to a wide range of tests, maintenance, and monitoring parameters to prevent accident or abnormal situations

- Control of components that may be disable automatic systems and safety functions
- Incomplete or insufficient operational steps in which only operational goals are described detailed procedural information about the operator's actions

These scenarios may cause the operator's stress or bring about the possibility that the operators may task the wrong manipulations. In addition, these operations with a high proportion of manual actions may be highly prone to human errors due to increased operator's workload [5, 7-9]. Therefore, automation of operations collaborated with operator's manipulation and automatic control would be expected to be lowered this operator's burden.

Typical approaches to automatic controllers in current NPPs include proportional-integral-differential (PID) controller, field-programmable gate array (FPGA), as well as programmable logic controller (PLC) [10-13]. For safety systems, the PLC is generally used to automatically act as a fast and reliable response to prevent malfunctions from propagating into major accidents. For non-safety systems, PID controllers or controllers that combine two out of three types of controllers (e.g., proportional-integral controllers) are the most popular among the existing NPPs. These controllers generally aim to stabilize a system within a defined range.

To tune the PID controller, traditional tuning methods have been applied, such as Ziegler-Nichols (ZN) [14], Cohen-Coon [15], and Astrom and Hagglund [16]. However, traditional methods still need re-tuning before being applied to industrial processes because the methods may cause frequent oscillations, large overshoot, and delayed settling time for higher-order systems [17]. An intelligent tuning method has been proposed to improve the capabilities of the existing PID parameter tuning techniques. A Harris hawks optimization (HHO) algorithm, which is suggested by Davut Izci et al., can find the optimal parameters of a PID controller installed on an aircraft pitch control system [18]. Optimization algorithms are suggested for DC motor control. In [19], an atom search optimization algorithm was improved by using simulated annealing (SA). Mahmud Iwan Solihin et al. compared the performance of tuning algorithms between particle swarm optimization and ZN [20]. Ignacio Carlucho et al. developed a multiple PID controllers with Deep Reinforcement Learning (DRL) algorithm that can adapt to changes in a mobile robot [21].

Some studies suggest an automatic operation algorithm by applying knowledge-based method. Sekimizu et al. [22] suggested an automatic algorithm for start-up operation. This algorithm can execute sequential controls following operation procedures according to if-then rules. In [22], knowledge-based method is applied to develop an automatic start-up intelligent control system (ASICS). At a pressurized water reactor simulator, ASICS controlled the components to reach the 2% reactor power state from the cold shutdown condition.

These studies shown the knowledge-based system that have powerfully robustness when the if-then logics are clearly defined. However, there are still some limitations in automating the operation process of NPPs. First, many operational tasks are difficult to change into clear if-then rule. This means that some operating steps are not specific enough to be executed using if-then rules. For example, an operating step would instruct the operator to manage the control rods to increase the power to 20% without detailed explanation such as how many steps are moved. Second, since the knowledge-based system is composed of linear functions (if-then rules), it is hard to handle flexible operations and changes in operating objectives, which are provided as the non-linear function. Therefore, applying artificial intelligence (AI) method may be a one way for design of an algorithm for autonomous operations in NPPs.

Recently, controllers applying artificial intelligence (AI) techniques have been studied in several industrial fields [23]. Since the 2000s, deep-learning techniques have drawn attention for several reasons: increasing computing power, increasing data size, and advances in deep-learning research [24, 25]. Among them, DRL is a trending approach because it has a training process that is very similar human's training mechanism. A DRL-based controller learns using its own experiences collected via trial-and-error, similar to humans. In addition, this DRL-based controller can perform tasks that classical controllers cannot perform, such as determining an operation strategy, planning sequential controls, making decisions according to current plant conditions, and finding optimal paths. Consequently, several DRL-based controllers have been suggested in robotics [10, 26], smart building [23, 27], power management [28-31], autonomous vehicles [11-13, 32, 33], railway industry [34], wind turbine [35], traffic signal [36], and nuclear power plants [37, 38].

These advantages in AI technologies have led to increased interest in the development of intelligent controllers to extend the automation capabilities of NPPs. Various AI-based methods are suggested for tunning process of PID controllers. PID controllers are typically applied in NPPs [39-41]. Bowen et al. developed a two-level hierarchical controller combined with a neural-network-based PID controller and a fuzzy controller. This hierarchical controller applied a multiunit small modular reactor [42]. Upadhyaya et al. suggested an autonomous operation system for a space reactor by applying a PID controller. To get the gains in PID controller, this study used the genetic algorithm [43]. Several studies have proposed AI-based applications to operate NPPs. Na et al. proposed a neuro-fuzzy controller to control the power distribution without any residual flux oscillations between the upper and lower halves of the reactor core [44]. In [45], an adaptive fuzzy controller was suggested to track reactor power in a research nuclear reactor. The suggested controller demonstrated good performance that can reduce the rise time than the PID controller. Arab-Alibeik and Setayeshi developed a neural adaptive inverse controller that can control the reactor power of a PWR type. After simulating the inverse dynamics of the nuclear reactor by a the multilayer neural networks, it was utilized as a controller [46]. In [47] and [48], a fuzzy-PID composite controller that directly switches between a fuzzy controller and a PID controller is proposed and utilized for reactor power operation of a molten salt reactor.

B. Motivation

AI-based controllers have been developed in several studies, they are not applied to NPPs at a practical level. This is mainly because AI-based controllers do not sufficiently prove their performance to guarantee robustness and correctness and solve regulatory issues, such as the transparency of the algorithm. However, it is very likely that the AI-based controller implemented as part of autonomous reactor controls will be an important aspect of small modular reactors and microreactors that can be operated remotely by an offsite operations crew [49].

Therefore, the use of controllers based on more advanced AI may be an alternative to developing algorithms for autonomous operation of NPPs. Furthermore, the broader application of AI technology should be considered for autonomous control in NPPs [50].

C. Goal of Study

This study aims at developing a DRL-based controller for an autonomous operation in NPPs during start-up and emergency. The controller is focused on conducting high-level operations that are similarly performed to the current operation strategy. To manipulate components similarly to operators, controllers currently aim to automate manual controls. Developed DRL-based controller is designed to handle the procedure-based operation (as knowledge-based system) and the operator's experienced-based operation (as DRL-based controller). The scope of this study is the work of an operator using a manual controller rather than an existing automatic controller as illustrated in Fig. 1.

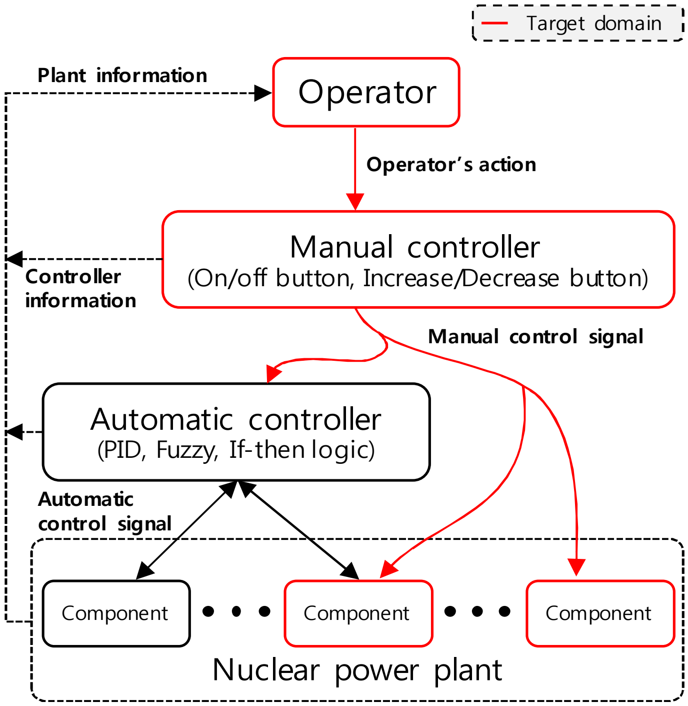


Fig. 1. Scope of the autonomous operation algorithm

D. Outline of Study

After the introduction, this paper describes the main concepts of reinforcement learning and then introduce representative DRL methods. In Chapter 2, among many DRL techniques, the techniques used in this study are mainly introduced. It will be helpful to understand the controller based on the DRL developed in Chapter 3 and 4. The DRL-controllers designed in Chapter 3 and 4 are considered the current operational strategy, i.e., existing systems, operating procedures, and staffing. Then, developed controllers are trained and demonstrated by using a compact nuclear simulator (CNS). In Chapter 3, DRL-controllers are developed for the normal operation. For normal operation, two DRL-controllers are proposed for the power-increase and bubble creation operation. In Chapter 4, this study designed a DRL-controller for the emergency operation. The controller shows the autonomous operation to reach the shutdown operation entry condition while keep the cooling rate (55 °C/hour), which is one of the required operational rules in the technical specification procedures. Then, this study discusses performance and limitations of developed DRL-controllers in Chapter 5. Last chapter is conclusion.

II. Methodology

A. Background of Reinforcement Learning

Reinforcement learning (RL) is a method for training an agent through its interaction with the environment [10], [49]-[51]. The agent interacts with the environment in a series of independent episodes, each of which comprises a sequence of turns. One episode consists of several discrete time steps, t=0,1,2,3⋯. At each time step (t), the agent receives a state (s_t) from the environment. Then, the agent selects an action (a_t) from a set of possible actions based on its policy (π). The policy is a mapping from states (s_t) to actions. The environment provides the next state (s_{t+1}) and a reward (r_t) for the action (a_t) of the agent. Through this interaction with the environment, the agent is trained to maximize the returned reward that is associated with the specified state (s_t) from the environment. Through this trial-and-error process, the agent determines the optimum policy for realizing the specified operational objective.

1. Background of Deep Reinforcement Learning

Using a controller with RL provides the possibility of finding an optimal policy, which includes solving the given problem or achieving operational goals in the sequential decision-making of the current state collected from the environment. One of the challenges in RL is finding an optimized policy function to obtain the maximum reward for all given states. Determining an optimized policy function may take a long time. To resolve this issue, recent studies have suggested using a neural network as an optimized policy function of the RL owing to the increased computing power and an improved method called the deep neural network.

Therefore, typical DRL algorithms combines RL and deep neural network models, to find the optimal policy.

First, this paper reviews of previous studies related to the use of DRL for the development and application to advanced control systems. Based on the summarized review, advantages of DRL-based controller are identified.

DRL, which is a method for training deep neural networks, provides a mechanism via AI agents that can optimize their control of an environment to realize a specified objective [10]-[13]. The interaction process between the AI agent and the environment can be represented by a closed-loop, which is very similar to the process of human learning [14], [15]. As a result, an AI agent can also develop its own experiences through trial-and-error, as humans do [16] and can perform tasks that a classic controller cannot do. Such actions may include selecting an operation strategy, operating nonlinear systems, making decisions based on current conditions, and optimizing operations [17]-[20].

Due to these characteristics of DRL, DRL is now an essential technology for the development of AI agents and is being used in many industries. Moreover, DRL is becoming a trend in advanced control systems due to increased safety and efficiency [21]. In the power system field, Suyang Zhou et al. [22] proposed an AI agent that was based on DRL for handling various operating scenarios for the economic dispatch of a combined heat and power system. In an application to wind turbines [23], DRL has been shown to overcome one of the most important disadvantages of the conventional control strategies, which is the tuning of control parameters and lowering fatigue. In energy management, Esmat

Samadi et al. proposed the use of decentralized multiagent systems (MASs) for integrated grid-connected microgrids. MASs with DRL have shown not only flexible management while considering customer consumption but also a reduced operating cost [24]. Hussain Kazmi et al. optimized the energy efficiency of hot water production by using a DRL controller, which could reduce the energy consumption by almost 20% for a set of 32 Dutch houses [25]. Tianshu Wei et al. also significantly reduced the energy cost of an HVAC (heating, ventilation, and air conditioning) system by using DRL instead of rule-based and model-based strategies [26]. In another study [27], DRL was adopted in urban rail transit to effectively improve energy management compared to the genetic algorithms and to provide dynamic programing.

The advantages of DRL for the development and application of advanced control systems through these research trends are briefly summarized as follows:

- Performance improvement compared to conventional control strategies (e.g., reducing operating costs, reducing failures, and increasing energy efficiency);
- Increased flexibility by adaptable control according to demand and change in practice;
- Optimal control to achieve the required goals.

B. Deep Reinforcement Learning Method

This section introduces the deep Q-learning network (DQN) that is well known as the basic methodology of DRL. Then, this section describes the training architecture and process for the DRL method used in this study.

1. Deep Q-learning Network (DQN)

DQN is an algorithm that combines deep learning methodology with Q-learning that is a kind of reinforcement learning. Q-learning aims to find optimal Q-values to achieve given goal. The Q-learning algorithm generates a Q-table in which states and Q-values are mapped. The Q-learning has the limitation that it is difficult to map the Q-values into a table for all states. To solve this problem, DQN computes Q-values by approximating states using deep neural networks. Google DeepMind has developed a DQN agent that can recognize information from the environment (game) and take action to get the highest score in the current state. The DQN agent trained through thousands of trial-and-errors and scored higher than human players in Atari games. In addition, the DQN agent showed that human-level manipulation is possible if the input values and rewards are properly designed, even if the domain is changed through training and verification in various Atari games.

2. Asynchronous Advantage Actor-Critic (A3C)

This study utilizes Asynchronous Advantage Actor-Critic (A3C), which is a type of DRL method, to reduce the agent training time for the continuous control module. Although DQN is a well-known basic model of DRL, slow training speed and biased actions are problematic. To address these issues, A3C utilizes parallel actor-learners that are based on the central processing unit's (CPU's) multiple threads and the asynchronous network update, while DQN utilizes one agent on one CPU.

Fig. 2 illustrates the A3C and DQN training algorithms. A3C replaces the experience memory with the local network memory to reduce the interactions between the collected training datasets. In addition, A3C utilizes multiple agents in the multiple simulations for training an agent that has a local neural network [51]. In A3C, each local network asynchronously updates the main network at regular intervals. In this asynchronous approach, after collecting a short memory (which is called a mini-batch) of data points, each of the local networks computes gradients and uses them to update the weights [52]. This update process increases the training speed by providing training datasets that consist of pairs of various actions that correspond to similar states. As illustrated in Fig. 3, the A3C agent updates the network's weights more frequently than the DQN agent.

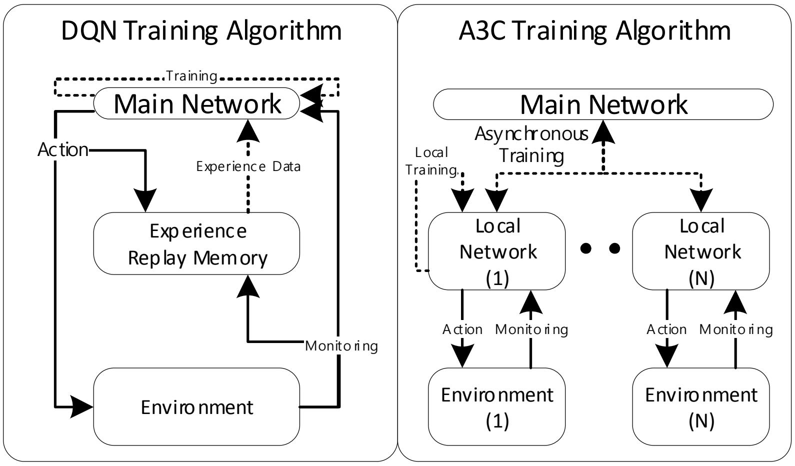


Fig. 2. DQN and A3C training algorithms

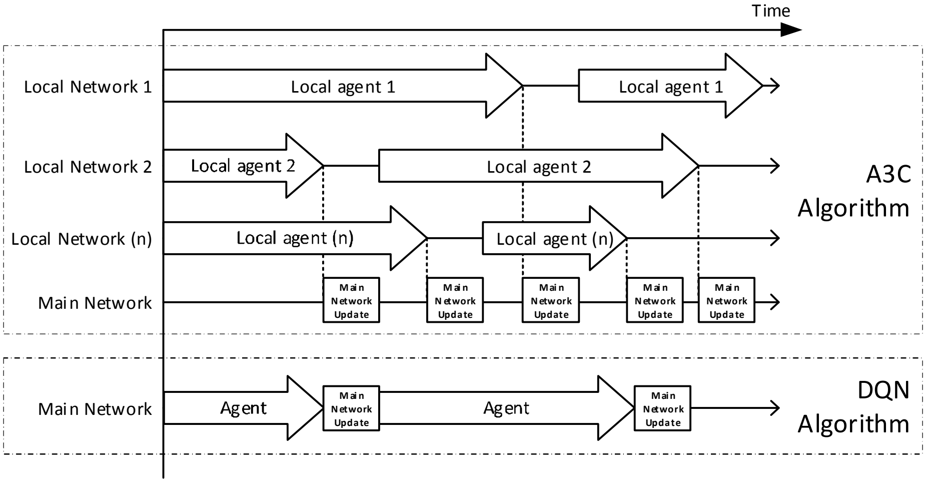


Fig. 3. Agent's weight update process

3. Soft actor critic (SAC) with distributed prioritized experience replay (DPER)

This study utilized SAC to improve the training stability of a DRL-based controller using an long short-term memory (LSTM) network model. The SAC was suggested to compensate for the deep Q-learning network (DQN), which is a basic model of the DRL. The drawback of the DQN is biased actions caused by predictions that rely on a single neural network. One network in the DQN predicts actions mixed with evaluations based on action probability and estimated reward.

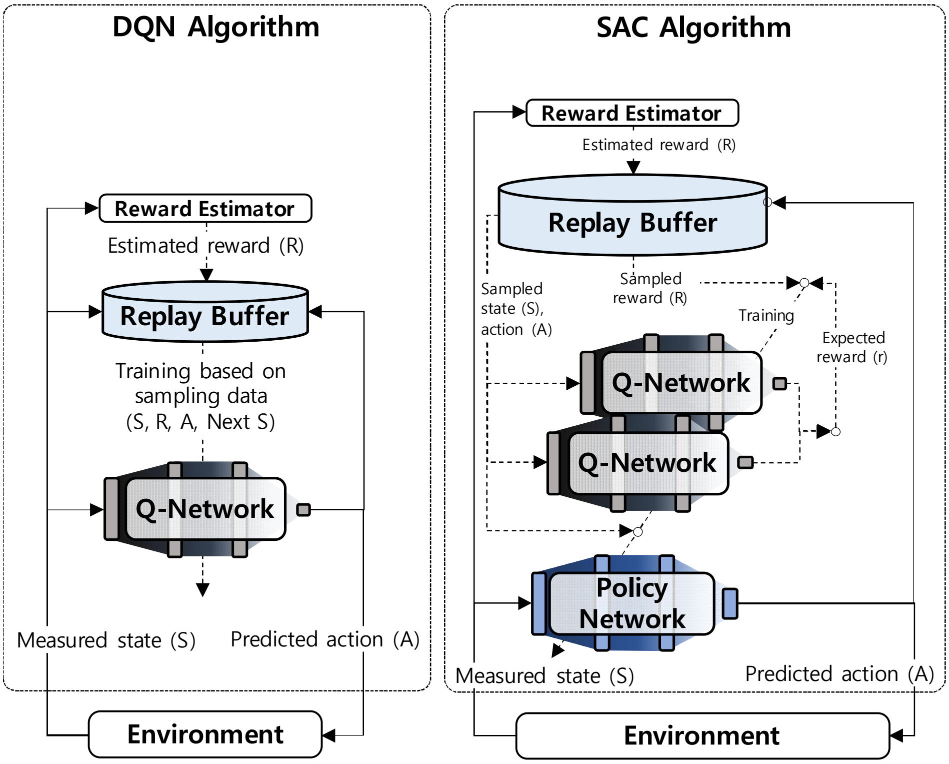


Fig. 4. Training algorithm of DQN (left) and SAC (right)

In contrast, SAC uses an actor-critic architecture with a separate value (Q-network) and policy network, as shown in Fig. 4. Q-networks calculate the expected rewards for an action taken in the current state. Then, the policy network predicts each action probability based on the expected reward and the current state. For training stability, SAC uses two Q-networks consisting of an online network and a target network. The target network update is delayed when updating the online network parameters over many iterations [10]. While updating online network parameters over many iterations, the target network parameters perform delayed updates from the online network at regular intervals, which helps reduce biased training.

Moreover, to reduce the training time, this study also adopted a distributed training architecture with distributed prioritized experience replay (DPER), a type of experience replay buffer, as shown in Fig. 5. In this architecture, the main network is trained with data collected from multiple simulations using a local neural network. Each local network contains only a policy network that regularly distributes training from the main policy network.

DPER was utilized to collect data simulated from local networks and improve the efficiency of the sampled data when the main network was trained [53]. DPER enables the DRL controller to remember and reuse experiences from the past, where observed transitions are stored for some time, usually in a queue, and sampled uniformly from this memory to update the network. For example, DQN training relies on randomly selected samples from the replay buffer. In contrast to the basic experience replay buffer, the DPER can sample data that are more frequently replayed transitions with high expected learning progress, as measured by the magnitude of their temporal difference (TD) error. TD error is the difference between the expected and actual rewards. Consequently, the main network learns by sampling data with higher stochastic priorities.

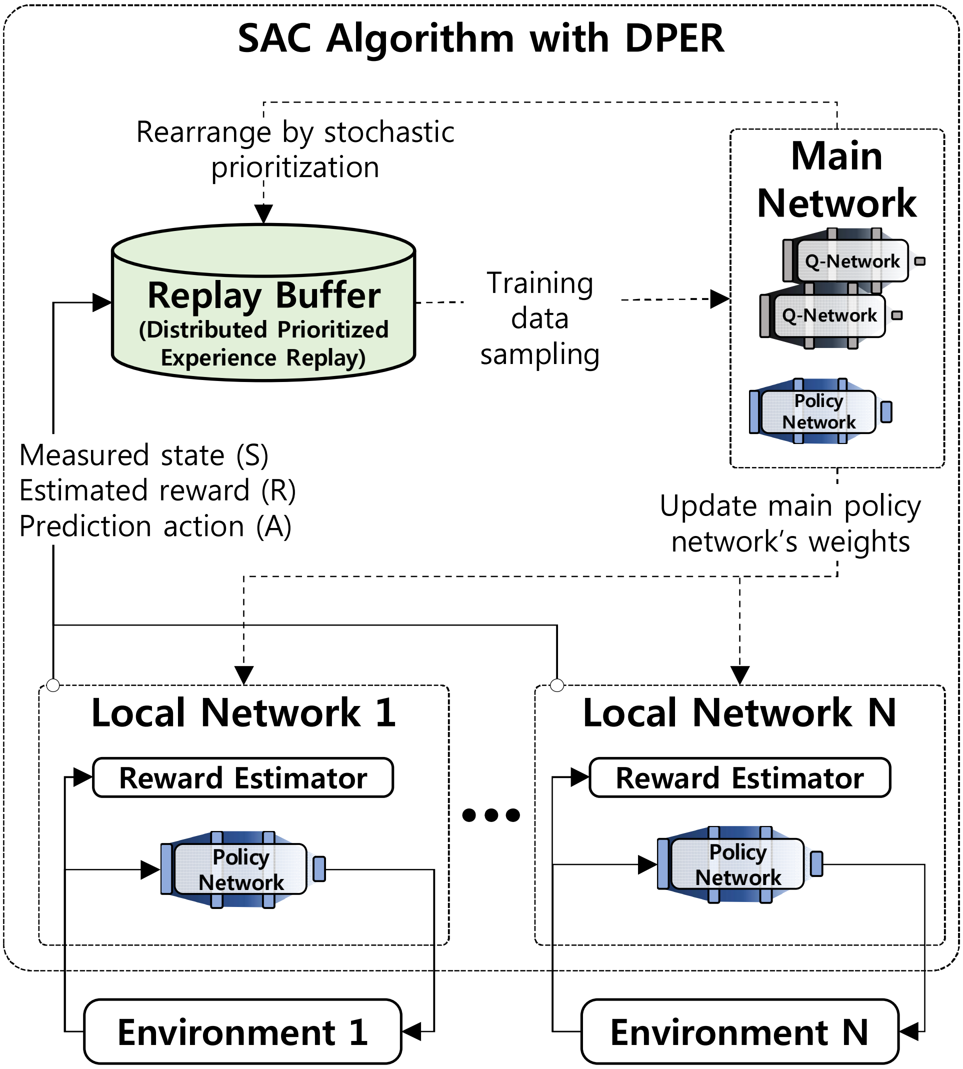


Fig. 5. Training algorithm of SAC with DPER

III. Normal Operation

Current NPP operating strategies were considered in the development of DRL-controllers for cold shutdown operations and increasing the reactor power from 2% to 100% autonomously. This study analyzed the operating procedures and the operator's tasks during the start-up operation of a reference plant, namely, a Westinghouse 900 MWe PWR. The analysis identified the operator's major tasks, and the tasks were categorized into automatic and manual actions. The manual actions were further divided into discrete and continuous actions.

A. Overview

The operation for increasing power from 2% to 100% is the part of the start-up operation that increases the temperature and power to the normal conditions for generating electricity after reactor refueling or shutdown. During the start-up operation, the operators follow general operating procedures (GOPs) for controlling systems and components. The cold shutdown operation is included in the GOPs, which provide instructions to start up the reactor and increase its power after refueling. A Westinghouse-900 MW PWR was used as the reference plant in the task analysis. The reference plant had six GOPs, as listed below [22, 54]:

- Reactor coolant system filling and venting
- Cold shutdown to hot shutdown
- Hot shutdown to hot standby
- Hot standby to 2% reactor power
- Power operation at than 2% power
- Secondary systems heat-up and startup

Fig. 6 shows the trend of the important parameters in the startup operation along with the relevant procedures. These parameters provide milestones for operators to achieve successful start-up operations. The target operations, i.e., the focus of this study, is in the gray area in Fig. 6.

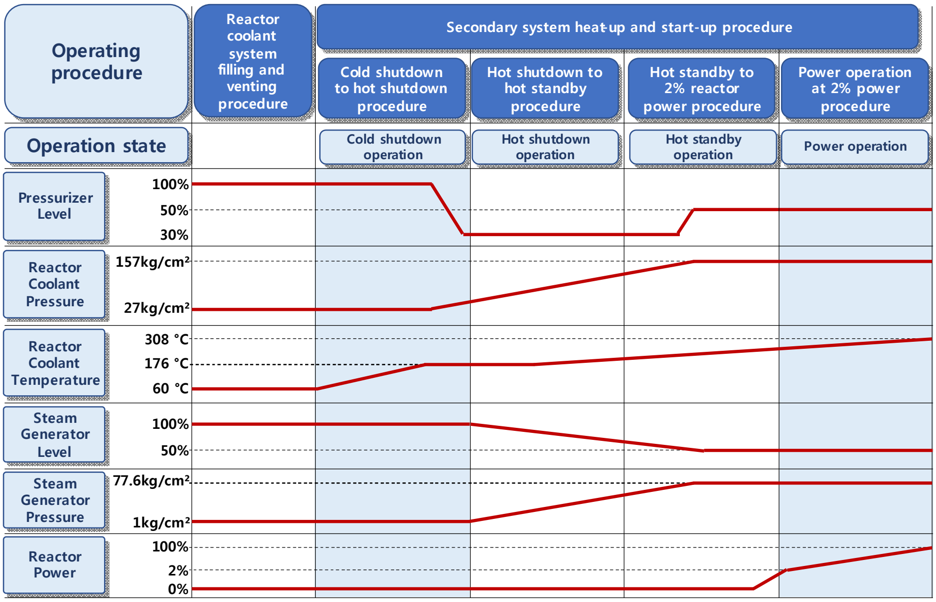


Fig. 6. Significant parameters of the startup operation

B. Bubble Creation Operation

This study compares the performances of DRL and PID controllers in the cold shutdown operation of NPPs. First, this study analyzes the GOPs of the bubble creation operation, which is part of the cold shutdown operation. It identifies the operational goals and manual controls by operators, and defines the inputs and outputs for the automatic controllers. Subsequently, a DRL-based controller is developed by combining a rule-based system, LSTM, and soft actor critic (SAC). Then, a PID controller was developed using the Ziegler-Nichols and DRL-based tuning methods. Finally, the performances of both controllers were compared and discussed.

1. Overview of Cold Shutdown Operation

The cold shutdown procedure provides instructions for heating the plant from the cold shutdown condition to the hot shutdown condition (Tavg < 176.7 °C, Keff < 0.99). This operation allows the components to increase the temperature of the primary system by maintaining pressure in the pressurizer. The goal of the cold shutdown operation is to create bubbles in the pressurizer, that is, the bubble creation operation, and then to control the pressure and level of the pressurizer.

Fig. 7 shows a simplified schematic of the components related to cold shutdown operation. The initial and final conditions of the operation of the plant variables are shown in Table 1.

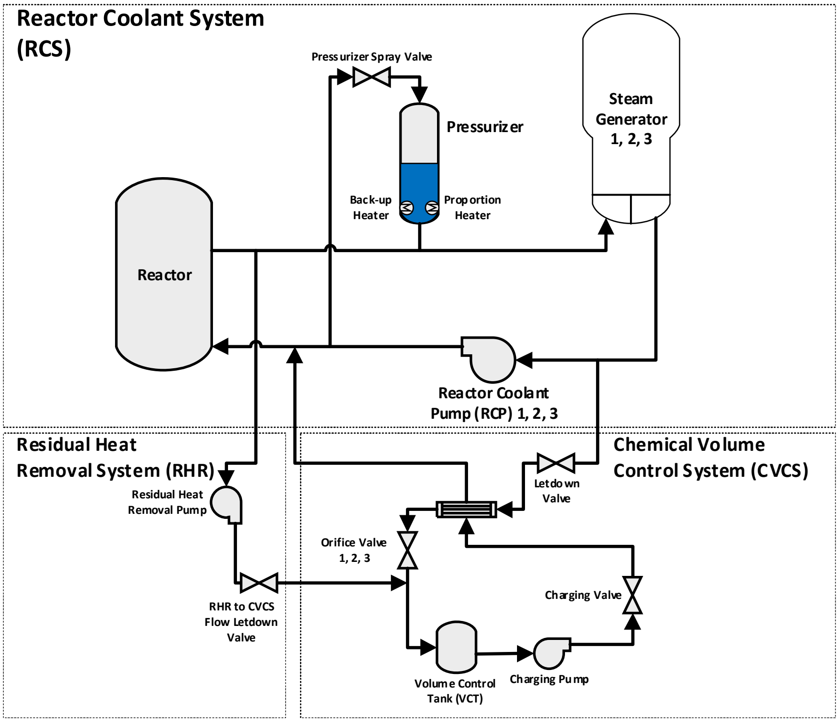


Fig. 7. Simplified schematic of related components

Table 1. Initial and final conditions of the cold shutdown operation

Major parameter	Initial condition	Final condition
Pressurizer pressure	27 kg/cm^2	27 kg/cm^2
Pressurizer temperature	84 °C	210 °C
Average temperature	81 °C	176 °C
Pressurizer level	100%	50%
Back-up heater	Off	On
Proportional heater	0%	100%
Letdown valve	0%	»40%
Pressurizer spray valve	0%	»30%
Charging valve	0%	»60%

The first step of the cold shutdown operation is to heat the coolant in the primary system by turning on all the pressurizer heaters (e.g., back-up and proportional heaters) and starting the reactor coolant pump. This leads to an increase in the temperature of the primary system and pressure inside the pressurizer. The pressure of the primary system, that is, the reactor coolant system (RCS), should be maintained between 25 kg/cm^2 and 29 kg/cm^2 despite the increase in the pressurizer temperature. Thus, the increase in pressure can be prevented by opening a letdown valve that handles the letdown flow rate from the RCS to a residual heat removal system (RHR). When the pressurizer temperature reached the saturation point of approximately 200 °C, its level decreased. A space filled with saturated steam was created on top of the pressurizer, allowing pressure to be controlled through the pressurizer spray. Subsequently, the level inside the pressurizer was maintained at 50% by adjusting the charging flow rate. It is known that this operation normally requires 8 h for actual NPPs.

2. Task Analysis of Cold Shutdown Operation

Task analysis identifies the objective of each operator action and defines the inputs and outputs of the actions to design the controllers. Table 2 presents the results of the task analysis of the operating procedure. The step numbers and tasks are the step numbers and instructions described in the procedure, respectively. Subsequently, task types are classified into control or check tasks. If the task type was "control," the task included action(s) on a component. The task type "check" is to check or monitor plant states without performing any action on components. The inputs and outputs of each task are then defined. The information necessary to design the controllers is then extracted by focusing on the task type of the control. Finally, four tasks were selected for implementation using the control algorithm, as listed in Table 3.

Control tasks were also classified as discrete or continuous controls. Discrete control has two separate states: "on or off" or "fully open or closed." For example, the task "controlling the proportional heater" in Table 3 is an example of discrete control because the heater only has two states: on or off. In contrast, continuous control adjusts the state of the component to satisfy the specific value of the parameters. In the cold shutdown operation, controlling the charging valve, letdown valve (RHR to CVCS flow), and spray valve belong to the category of continuous control because the positions of those valves are adjusted between 0% and 100% to maintain the specified RCS pressure.

Table 2. Task analysis result for cold shutdown operation

Step Number	Task	Task Type
1	Cold shutdown operation should be completed within 8 hours.	Check
2	The pressurizer pressure should be maintained between 25 kg/cm^2 and 29 kg/cm^2 during cold shutdown operation.	Check
3	The RHR system should be isolated from RCS before the pressurizer temperature reaches 200 °C or its pressure reaches 30 kg/cm^2.	Check
4	The reactor coolant loops and the pressurizer are filled and vented.	Check
5	The reactor coolant boron concentration is greater than or equal to that of the cold-shutdown condition.	Check
6	The residual heat removal (RHR) system is served with all loop isolation valves open and one operational RHR.	Check
7	Close main steam isolation valves.	Check
8	Close steam generator (S/G) power operated relief valves.	Check
9	The makeup control system is in Auto mode.	Check

Table 3. Simplified operational task for cold shutdown operation

Task	Input	Output	Control Type	Constraints
Put back-up heater from Off to ON.	Back-up heater state	Back-up heater control	Discrete control	1) Maintain RCS pressure between 26 kg/cm^2 and 28 kg/cm^2. 2) Maintain pressurizer level at 50%.
Increase the power of proportional heater from 0% to 100%.	Proportional heater state	Proportional heater control		
Adjust letdown valve (RHR to CVCS flow) within the RCS pressure boundary.	RCS pressure,	Letdown valve control	Continuous control	
Adjusting spray valve within the RCS pressure boundary.	Letdown valve	Spray valve control		
Adjust charging valve to maintain pressurizer level.	position	Charging valve control		

3. Development of a DRL-based Controller

This section introduces the development of a DRL-based controller to create bubbles for the pressurizer and then controls the pressurizer pressure and level during the cold shutdown operation. The DRL-based controller comprises two DRL controllers and a rule-based controller, as shown in Fig. 8. The rule-based controller performed the discrete controls listed in Table 4. As a result of the task analysis, if specific rules, that is, if-then logic, can be defined, the rule-based controller is applied, as shown in Table 4. Therefore, the back-up heaters and proportional heaters are controlled using a rule-based controller.

DRL controllers perform continuous control, for which it is difficult to define specific rules, for example, how much a valve should be open to maintain the pressure. DRL controllers are divided into pressure and level controllers, as shown in Table 3. As shown in Table 4, the pressure DRL controller aims to maintain the pressure by adjusting the letdown and spray valve, whereas the level DRL controller adjusts the charging valve to maintain the pressure level at 50%. DRL controllers use an LSTM network trained using the SAC learning algorithm. The LSTM is known to show a good performance in handling time-related, dynamic data. In addition, the authors' previous studies also showed that the LSTM could support well the operation of nuclear systems and the diagnosis of events [54-57].

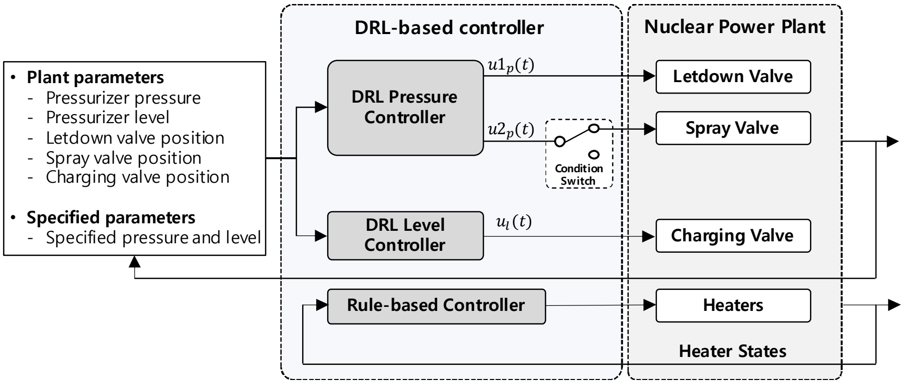


Fig. 8. DRL-based controller block diagram

Table 4. Controller tasks for cold shutdown operation

Task type	Controller	Action
Discrete control	Rule-based controller	If the back-up heater state is "Off," push "On" button.
		If the proportional heater power is 0% or below 100%, increase the power to 100%.
Continuous control	Pressure DRL controller	Maintain the pressurizer pressure between 26 kg/cm2and 28 kg/cm2by adjusting letdown valve.
		If the pressurizer temperature reaches the saturation point of about 200 °C, maintain the pressurizer pressure between 26 kg/cm2and 28 kg/cm2by adjusting the spray valve.
	Level DRL controller	Adjust the charging valve to maintain the pressurizer level at 50%

a. Design of reward algorithm

This section presents the reward algorithm for the DRL-based controllers. In DRLs, the reward is an essential element used to update the weights of the neural networks. A reward algorithm was used to evaluate the actions predicted by a network and provide guidance for updating the weights of the neural network [58, 59]. DRL-based controllers obtain rewards by evaluating the actions performed within given states. Thus, the reward algorithm evaluates the performed action under the given state and creates training datasets that consist of pairs of states, actions, and rewards.

Two reward algorithms for the level and pressure controllers were suggested to reflect the operational constraints identified in Table 3. Reward algorithms aim to minimize the distance from the current state to the desired state, for example, the midpoint of the pressure boundary or the specified pressurizer level.

As the level controller does not operate until the saturation point is reached, the level-reward algorithm provides a reward when the level controller starts control. As shown in Fig. 9, the reward value is defined as the difference between the current pressurizer level and desired level (50%), as shown in Equation (1). The pressure level in the pressurizer was varied between 0% and 100%. To provide a reward range between 0 and 1, the scaling value is defined as 50, which is the maximum distance from the desired pressurizer level to the limits of range (0% to 100%). For instance, the level reward is zero for the lowest reward when the current level is within the limits of the level range (Points A and B in Fig. 9). As the current level increases from Point C to D, approaching the desired value, the level reward increases from 0.8 to 1. The level reward algorithm provides a maximum reward of one when the level controller is running successfully to maintain the current level at the desired pressure level.

$$\text{Level reward} = \left(1 - \frac{|\text{Current level} - \text{Specified pressurizer level}|}{\text{Scaling value}}\right) \qquad (1)$$

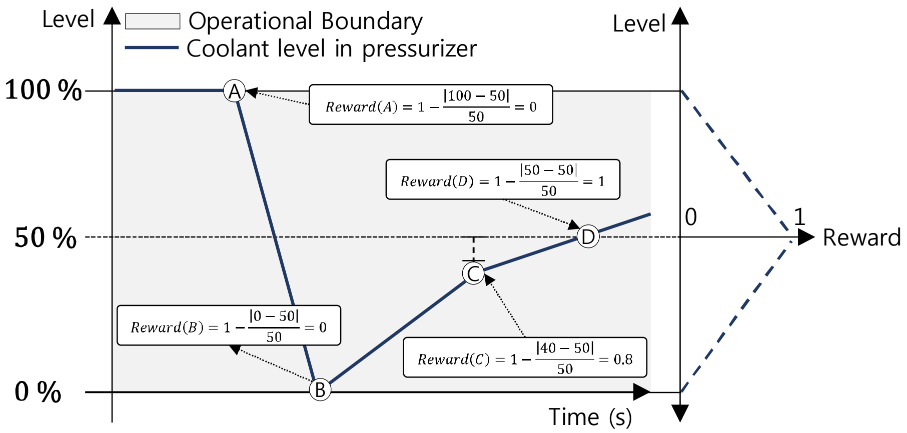
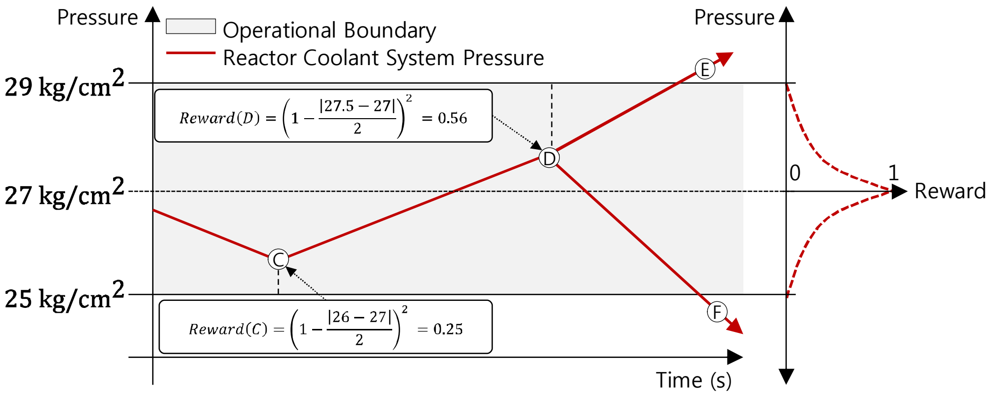

Level
Operational Boundary
Coolant level in pressurizer
Level
100 %
50 %
0 %
A
B
C
D
$Reward(A) = 1 - \frac{|100-50|}{50} = 0$
$Reward(D) = 1 - \frac{|50-50|}{50} = 1$
$Reward(B) = 1 - \frac{|0-50|}{50} = 0$
$Reward(C) = 1 - \frac{|40-50|}{50} = 0.8$
0
1
Reward
Time (s)

Fig. 9. Level reward algorithm for achieving operational goals

The pressure-reward algorithm provides a reward for the pressure controller to maintain a pressure between 25 kg/cm^2 and 29 kg/cm^2. The reward value was calculated as the difference between the current RCS pressure and the desired condition, as shown in Equation (2). The scale value is defined as 2, which is half the desired range of pressure. Because the pressure changes slightly during the cold shutdown operation, the pressure reward uses the squared reward to consider pressure changes sensitively. For instance, in Fig. 10, the reward value at Point C is the difference between the current RCS pressure and the lower desired pressure (25 kg/cm^2), which is 0.25 at the current pressure 26 kg/cm^2. In the case of point D, that is, at a current pressure of 27 kg/cm^2. The reward value is the difference between the current pressure and the upper desired pressure (29 kg/cm^2), which is 0.56. Therefore, the pressure reward increased as the pressure approached the middle of the pressure boundary, with a maximum of 1. When the current pressure exits the desired pressure boundary, e.g., Point E and F in Fig. 10, the training is terminated.

$$\text{Pressure reward} = \left(1 - \frac{|\text{Current pressure} - \text{Middle of pressure boundary}|}{\text{Scaling value}}\right)^2 \qquad (2)$$



Pressure
Operational Boundary
Reactor Coolant System Pressure
Pressure
29 kg/cm^2
27 kg/cm^2
25 kg/cm^2
$Reward(D) = \left(1 - \frac{|27.5 - 27|}{2}\right)^2 = 0.56$
$Reward(C) = \left(1 - \frac{|26 - 27|}{2}\right)^2 = 0.25$
C
D
E
F
0
1
Reward
Time (s)

Fig. 10. Pressure reward algorithm for achieving operational goals

In addition to this termination condition of pressure, another termination condition was defined during the training. If the operation time in the training reaches eight hours, the episode is terminated because the GOP instructs that the operation should be completed within 8 h.

b. Design of Long Short-term Memory Network (LSTM)

A neural-network-based architecture, that is, a part of the DRL-based controller, was developed to perform continuous controls. To generate control actions from the DRL-based controller, this study used LSTM cells that can calculate time-series data [60]. LSTM cells were developed from recurrent neural networks (RNNs). An RNN can naturally represent dynamic systems and capture their dynamic behavior. This is a powerful network for extracting information features related to a dynamic system in its hidden layer [61]. However, an RNN may exhibit a gradient vanishing problem when the network has five or more layers [62]. The drawback is that the gradient value becomes too large or vanishes exponentially to zero, whereas the weight in many layers is updated. Therefore, there is a restriction on the dataset for long-term memory within the RNN. Thus, LSTM cells have been proposed to solve this problem.

Fig. 11 shows the structure of a typical LSTM cell. Each LSTM cell is composed of units that retain the state across time steps, called "constant error carousels" (CECs), as well as three types of specialized gate units (input, output, and forget gates) [63]. The following equations describe the output from each gate unit in the LSTM cell, where x^t is the input of the LSTM cell. The input gate, forget gate, output gate, cell state, and output of the LSTM cell at the current time step t are i^t, f^t, o^t, c^t, and h^t, respectively. The weights between the input layer and input gate, the input layer and forget gate, and the input and output gates are W_{xi} , W_{xf} and W_{xo}, respectively. The weights between the hidden recurrent layer and forget gate, hidden recurrent layer and input gate, and hidden recurrent layer and output gate of the memory block are W_{hi} , W_{hf} and W_{ho}, respectively. Finally, b_i , b_f, and b_o are the additive biases of the input, forget, and output gates, respectively. This set of activation functions consists of the sigmoid function, elementwise multiplication, e.g., the inner product of a vector, $\circ$, and hyperbolic activation function. At time step 0, o_0 and h_0 were initialized as zero matrices.

$$\mathrm{i_t} = \sigma(\mathrm{x^t W_{xi}} + \mathrm{h_{t-1} W_{hi}} + \mathrm{b_i}) \quad (3)$$

$$\mathrm{f_t} = \sigma(\mathrm{x^t W_{xf}} + \mathrm{h_{t-1} W_{hf}} + \mathrm{b_f}) \quad (4)$$

$$\mathrm{o_t} = \sigma(\mathrm{x^t W_{xo}} + \mathrm{h_{t-1} W_{ho}} + \mathrm{b_o}) \quad (5)$$

$$c_t = f_t \circ c_{(t-1)} + i_t \circ tanh(x^t W_x c + h_{(t-1)} W_h c + b_c) \quad (6)$$

$$h_t = o_t \circ tanh(c_t) \quad (7)$$

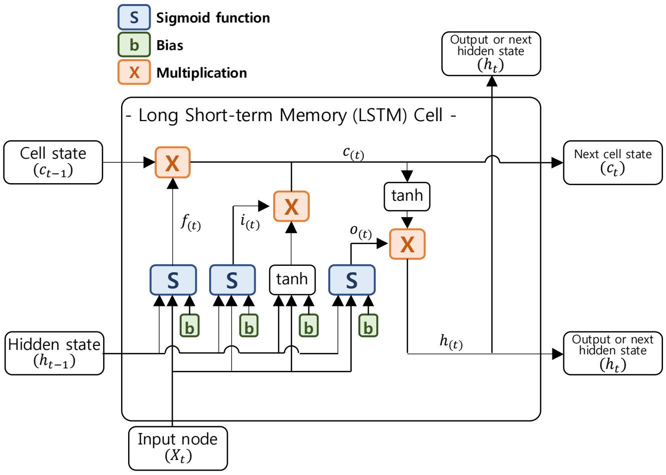


Fig. 11. Structure of an LSTM cell

LSTM cells allow the DRL controller to handle the NPP parameters and control the components with high performance because the NPP data exhibit the characteristics of non-linearity and time-series data. Fig. 12 illustrates the structure of the LSTM network applied to the DRL controller policy. Value networks have the same structure as the policy network, except for the output layer that generates the expected reward. Generally, an LSTM network model consists of an input layer, an LSTM layer, and an output layer. The sizes of the input and output layers are defined according to the number of plant parameters. The number of LSTM cells is equal to the size of the time window.

The input layer of the LSTM network has a time window of 10 s, which considers the trend of the plant parameters by exploiting the collected historical data. Therefore, the DRL controller uses states that include the current and previous states as a two-dimensional array. Thus, the number of LSTM cells is equal to the size of the time window.

As shown in Fig. 12, the proposed DRL controller includes two policy networks composed of LSTM networks to manage the pressurizer pressure and level. The DRL controller used five plant parameters and two specified pressures and levels. The plant parameters consisted of three component states (letdown/charging/spray valve positions) and two pressurizer states (pressure and level). The pressure policy network uses four plant parameters (pressurizer pressure, pressurizer level, letdown valve position, and spray valve position) and two modified variables that include the distance from the pressure boundary. The level-policy network uses two plant parameters (pressurizer level and charging valve position) and two distance values from the specified level.

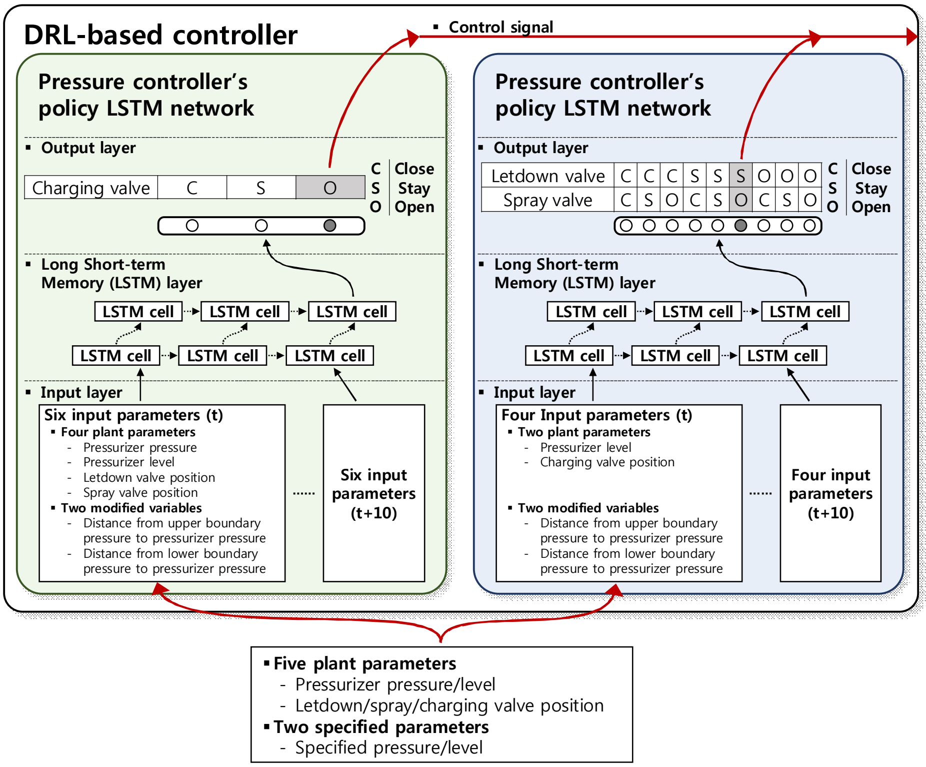

DRL-based controller
Control signal
Pressure controller's policy LSTM network
Output layer
Charging valve
C
S
O
C Close
S Stay
O Open
Long Short-term Memory (LSTM) layer
LSTM cell
Input layer
Six input parameters (t)
Four plant parameters
- Pressurizer pressure
- Pressurizer level
- Letdown valve position
- Spray valve position
Two modified variables
- Distance from upper boundary pressure to pressurizer pressure
- Distance from lower boundary pressure to pressurizer pressure
Six input parameters (t+10)
Pressure controller's policy LSTM network
Output layer
Letdown valve
C C C S S S O O O
Spray valve
C S O C S O C S O
C Close
S Stay
O Open
Long Short-term Memory (LSTM) layer
LSTM cell
Input layer
Four Input parameters (t)
Two plant parameters
- Pressurizer level
- Charging valve position
Two modified variables
- Distance from upper boundary pressure to pressurizer pressure
- Distance from lower boundary pressure to pressurizer pressure
Four input parameters (t+10)
Five plant parameters
- Pressurizer pressure/level
- Letdown/spray/charging valve position
Two specified parameters
- Specified pressure/level

Fig. 12. Structure of LSTM network for DRL the controller policy network

The output layer consists of a set of actions for controlling the target components, such as the letdown, spray, and charging valves. The control strategies of one valve are threefold, that is, open, closed, or no control. If a control strategy selects "open valve,' the valve position will increase. In the case of "no control," the valve maintains the current position. Therefore, the level policy network output is one of the three control strategies shown in Fig. 12. However, because the policy network for pressurizer pressure aims to control the letdown and spray valves, the set of actions includes nine cases that combine the three control strategies of the two valves. To select a control strategy, the output size of the output layer should be equal to the number of control strategies. Therefore, in the case of the DRL pressure controller, nine control strategies for controlling the letdown and spray valves were mapped to the output valves for the output layer of the LSTM network.

To select one control strategy among the nine cases, the DRL-based controller for pressurizer pressure also calculates the expected reward acquisition probability for each action in the LSTM network. The softmax function was used to calculate the probability of each control strategy in the output layer. The softmax function can map the network output to a probability distribution between zero and one. Therefore, the sum of the values of the generated output is one. Therefore, the LSTM network can calculate the probability value for each control strategy.

c. Training of DRL-based controller

CNS was used as a real-time testbed to train and validate the developed DRL-based controller. The CNS was originally developed by the Korean Atomic Energy Research Institute (KAERI) with reference to a Westinghouse 930 MWe three-loop PWR [64]. Fig. 13 shows the chemical and volume control system in the CNS.

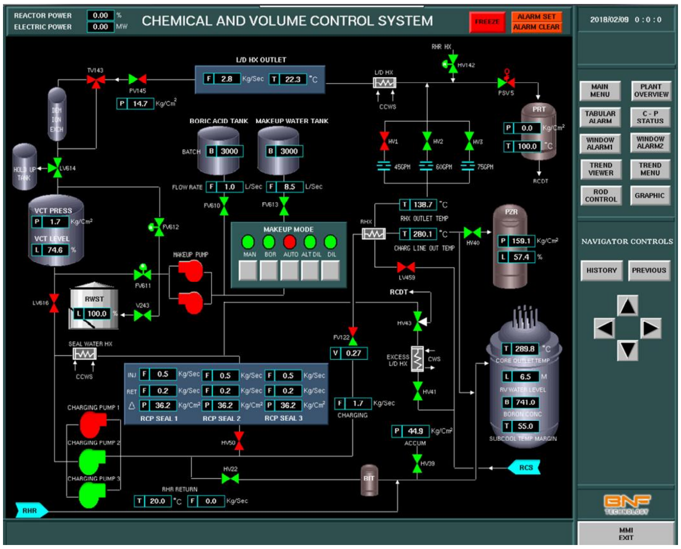


Fig. 13. Chemical and volume control system (CVCS) in the CNS

Fig. 14 shows the multi-CNS environment for training and validating the DRL-based controller. Two desktop computers were used to construct the multi-training environment. A DRL-based controller is installed on the main computer. CNSs were installed on a subcomputer with Intel Core(TM) i7-8700K and 16 GB of memory. Twenty CNSs were simultaneously simulated. One of the local networks is connected to a CNS simulation through user datagram protocol (UDP) communication. The global network was trained on two Nvidia Geforce GTX1080Ti graphics cards, whereas the SAC training algorithm was trained using a 10 CPU core on Intel CoreX i7-7820X. The DRL-based controller was programmed using Python. PyTorch, which is a well-known Python machine-learning library, was used to develop a DRL-based controller.

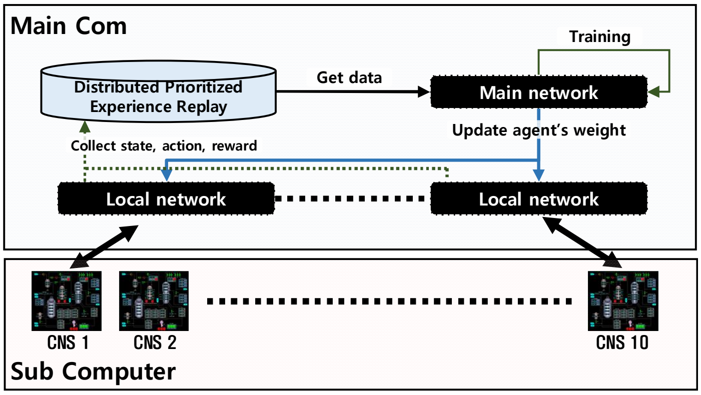


Fig. 14. Multi-CNS environment for training and validating the developed DRL-based controller

To achieve acceptable performance of the proposed DRL-based controller, it was trained until it reached a stable training state. DRL-based pressure and level controllers are trained through many episodes, each of which is completed if at least one controller reaches the termination condition. All controllers stop training when the average maximum probability converges to a certain value, or when the value stabilizes. The average maximum probability is the mean value of the probability of the actions selected by the DRL controller in one step. In one step, the DRL-based controller learns using 256 sample data from the DPER. In this study, the experimental results considering the entire operation time confirmed that operational goals could be reached when 256 data points were sampled. If more (512) or less (128) than this, learning fails. The average maximum probability refers to the degree to which the DRL controller completes the training. If the average maximum probability is higher than the previous step, it implies that the DRL controller selects actions that are more likely to succeed. Fig. 15 shows the trend of the average maximum probability per step over time. Fig. 16 shows the trend in the rewards per episode. The y-axis represents the total reward earned in each episode.

The pressure and level controllers reached a stable value after 2000 episodes. Approximately 84 h of training were required until the DRL-based controllers learned how to adjust the charging, letdown, and spray valves to achieve the operational goal. At approximately 500 episodes, the rewards reached 400 (pressure controller) and 275 (level controller).

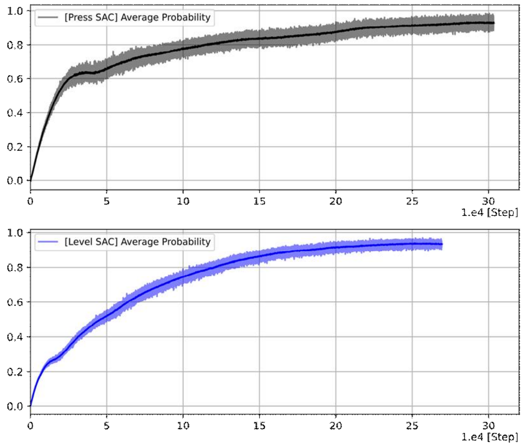


Fig. 15. Average maximum probability per episode

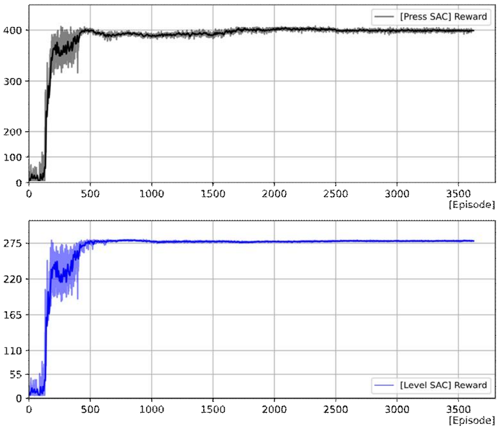


Fig. 16. Reward per episode

4. Development of a PID-based Controller

A PID-based controller was designed as shown in Fig. 17. The PID-based controller should manage five components (charging, letdown, spray valves, and back-up and proportional heaters) to achieve two operational goals (pressurizer pressure and level). If-then logic was applied to control the two heaters. Therefore, three PID controllers are developed for the three valves.

In general, a PID controller is applied to a single-input, single-output system without considering the disturbance and nonlinearity of the system [65]. Thus, three PID controllers must be developed to adjust the charging, letdown, and spray valves. In Fig. 17, PID Controllers 1 and 2 adjust the letdown and spray valves for pressurizer pressure, whereas PID Controller 3 manages the charging valve at the pressurizer level. Because the spray valve can be operated after pressurizer bubble creation, a condition switch was added to avoid unnecessary operations. The operational goal of PID controllers 1 and 2 was to regulate the pressurizer pressure within a specified pressure ($r_p(t)$). The pressure deviation error ($e_p(t)$) between the actual pressure value ($y_p(t)$) and pressure set-point ($r_p(t)$) is commonly used in PID controllers 1 and 2. PID controller 3 controls the charging valve to satisfy the pressurizer level.

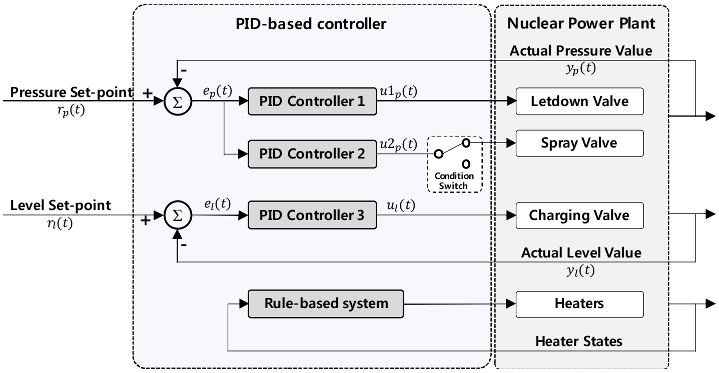


Fig. 17. PID-based system block diagram

a. Background of PID controller

The PID controller is based on classical optimal control theory that uses a control loop feedback mechanism to control the process variables [66]. PID controllers are typically used in industrial control applications to regulate temperature, flow, pressure, speed, and other process variables. To increase plant performance and safety, a PID controller is also one of the most commonly used process controllers in NPPs [67]. The PID controller in Fig 18 aims to minimize the cost function comprising three terms: current error with the proportional term, past errors with the integral term, and future errors with the derivation term. Its control output $u(t)$ is linearly obtained by combining the proportional, integral, and differential of the error $e(t)$, between the set value $r(t)$, and the actual value $y(t)$, thus realizing the control of the controlled object. Among them, proportional regulation can accelerate the system response speed, integral regulation can eliminate the steady-state error of the system, and differential regulation can realize advanced control of the system [48]. However, the regulation performance of classical PID controllers includes the regulation time, overshoot, and system stability [68].

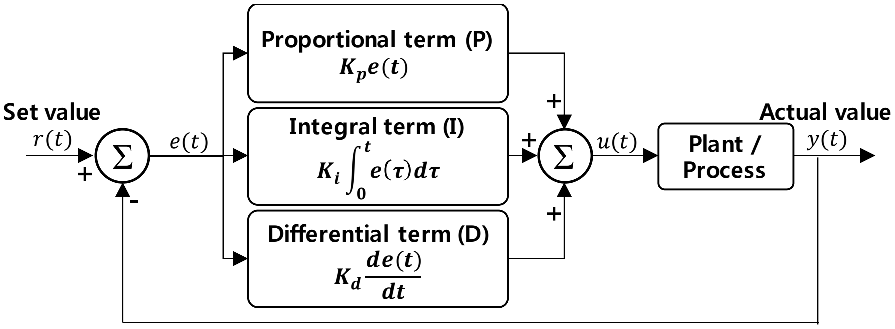


Fig. 18. Block diagram of the PID controller principle

b. PID-based controller tuning using Ziegler-Nichols rule and DRL algorithm

This study applies the Ziegler-Nichols closed-loop tuning method and DRL tuning method to achieve an acceptable performance of PID-based controllers. As a traditional tuning method, the Ziegler-Nichols method is well known as a suitable tool for nuclear power plants whose mathematical models are unknown or difficult to obtain [69]. Despite many design methods for PID controllers, the Ziegler-Nichols rule is one of the most widely used design methods in the literature [70, 71]. In addition, the Ziegler-Nichols tuning method is used for automatic control in Korean NPPs [67].

According to the Ziegler-Nichols tuning method, a PID controller is tuned by first setting it to the P-only mode, which means that the integral gain (K_i) and derivative gain (K_d) are set to zero. The proportional gain (K_p) increases until the ultimate gain (K_u), where the system starts to oscillate, and an ultimate oscillation period (T_u), as shown in Fig 19 Then, K_p, K_i, and K_d were then approximated using Table 5 [14].

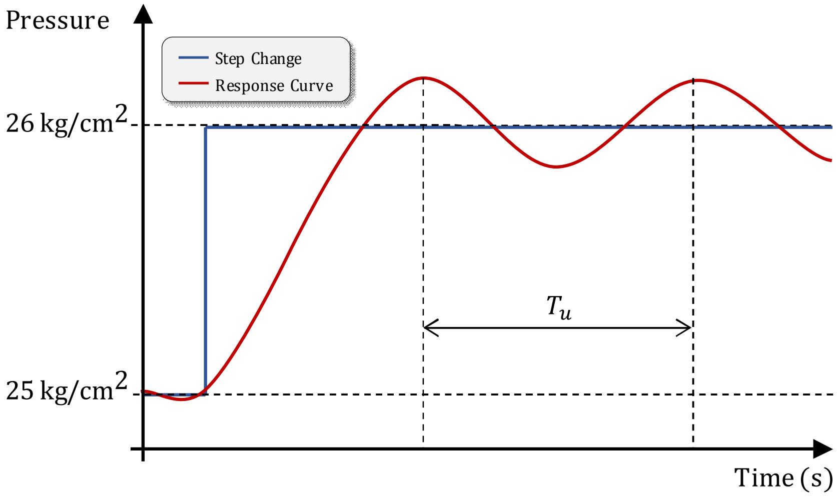


Fig. 19. The change of pressure about step response

Table 5. Ziegler-Nichols formula for PID controller tuning rules

Controller	K_p	K_i	K_d
P	$0.50K_u$	0	0
PI	$0.45K_u$	$0.54K_u/T_u$	0
PID	$0.60K_u$	$1.20K_u/T_u$	$3K_uT_u/40$

＊ $K_u = K_p$

As an alternative for the intelligent tuning methods, the DRL-based tuning was applied. This uses a DRL approach to obtaining the gains of controllers (Kp, Ki, and Kd). Fig. 20 shows the process of the DRL-based tuning method. At the first step, the method initializes the gains as Kp=0.1, Ki=0, and Kd=0. Then, in the second step, the policy network with simple DNN layers generates the gains, and the Q-network generates the expected reward by using initialized gains. The third step applies the gains to the PID controller and runs the CNS with the controller. In the fourth step, the cumulative rewards resulting from the simulation are calculated by using Equations (1) and (2). The fifth step calculates the loss value by the deviation between the cumulative rewards and the reward expected from the Q-network. Then, the policy and Q-network weights are updated by using the loss value in the sixth step. The seventh step evaluates whether the cumulative reward reaches a stable training state. If it is evaluated to be satisfactory, the generated gains are finally selected as the final gains of the controller.

The PID controllers for the letdown, spray, and charging valves were tuned by using DRL-based tuning algorithm. Fig. 21 shows the history of cumulative reward per episode. The y-axis represents the total reward earned in each episode. Each PID controller is tuned until it reaches a stable training state.

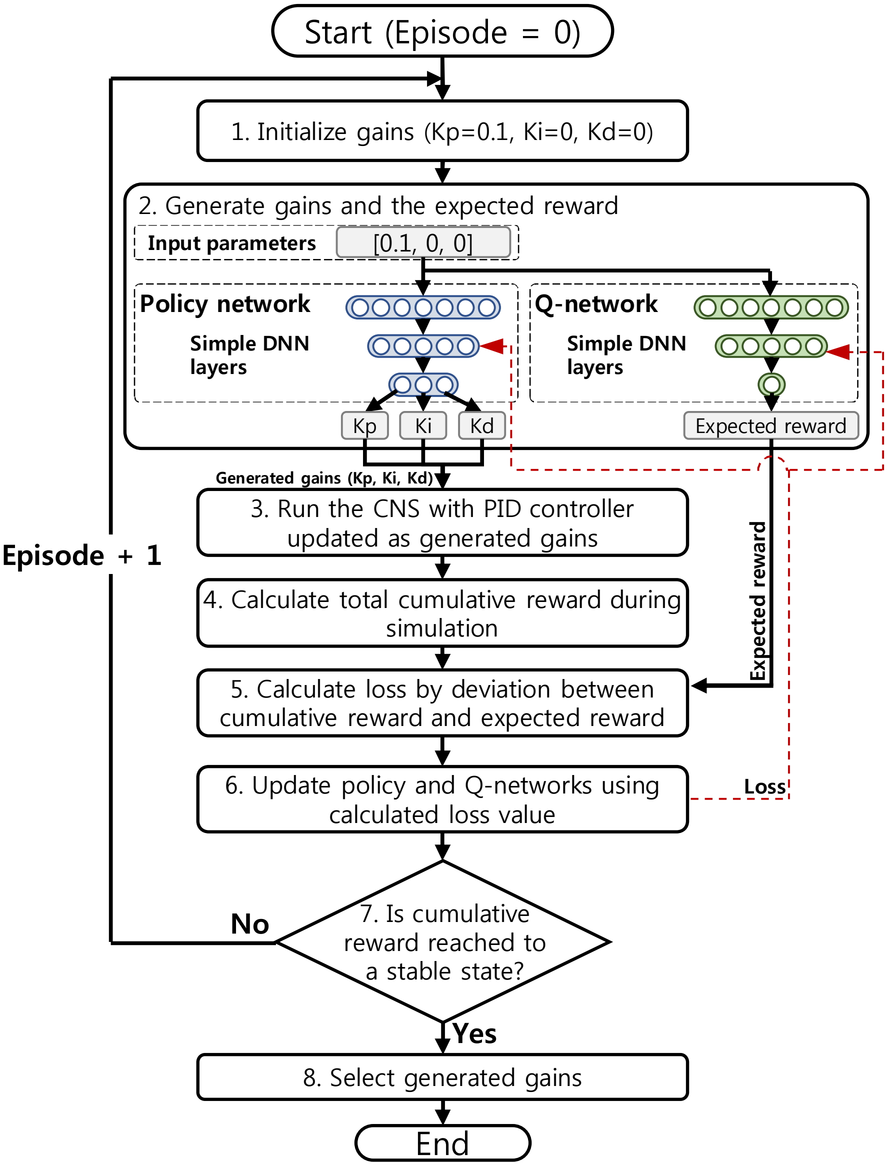

Start (Episode = 0)
1. Initialize gains (Kp=0.1, Ki=0, Kd=0)
2. Generate gains and the expected reward
Input parameters
[0.1, 0, 0]
Policy network
Simple DNN layers
Q-network
Simple DNN layers
Kp
Ki
Kd
Expected reward
Generated gains (Kp, Ki, Kd)
3. Run the CNS with PID controller updated as generated gains
4. Calculate total cumulative reward during simulation
Expected reward
5. Calculate loss by deviation between cumulative reward and expected reward
6. Update policy and Q-networks using calculated loss value
Loss
Episode + 1
No
7. Is cumulative reward reached to a stable state?
Yes
8. Select generated gains
End

Fig. 20. Flowchart of DRL tuning algorithm

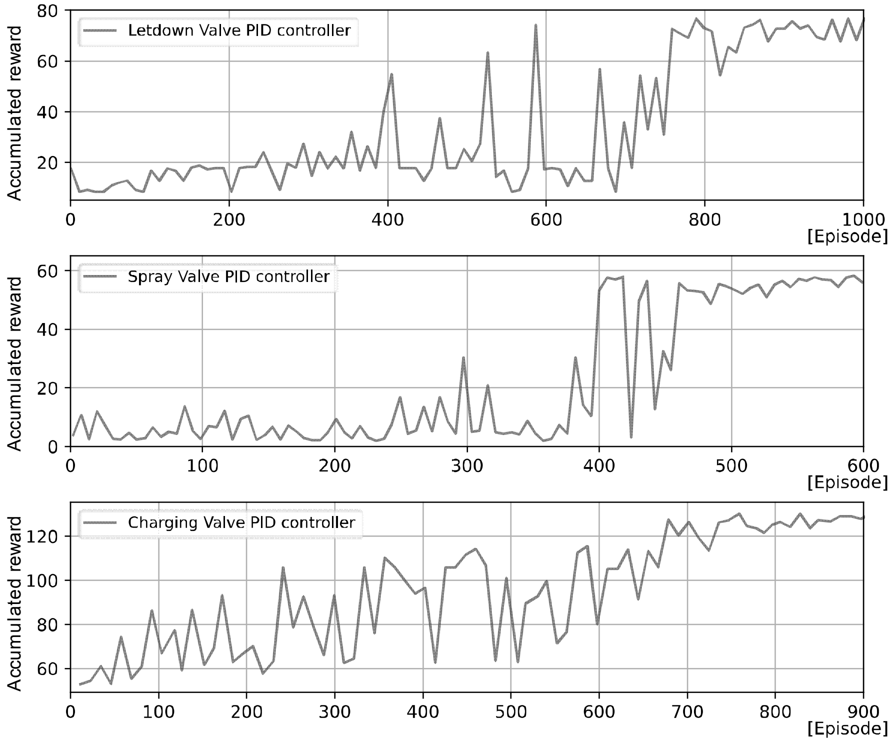


Fig. 21. Reward per episode

Table 6 shows the tuning results for the PID controllers by using the Ziegler-Nichols and DRD-based tuning method. Fig. 22 also compares the performances of the different tuning results for the pressure and level of pressurizer. Because the pressure is managed by the letdown and spray valves, the letdown valve controller was first tuned and then the spray valve was tuned later. The comparison indicates that the DRL-based tuning shows better performances in time and accuracy than the Ziegler-Nichols method.

Table 6. Tuning results based on Ziegler-Nichols method and DRL tuning method

Controller	Initial parameter	Tuned parameter (Ziegler-Nichols method)	Tuned parameter (DRL method)

Letdown valve	Kp=0.2
	Ki=0
	Kd=0
		Tu=60 sec
		Kp=0.12
		Ki=0.004
		Kd=0.9
			Kp=1.487
			Ki=1.155
			Kd=0.106

Spray valve	Kp=0.1
	Ki=0
	Kd=0
		Tu=70 sec
		Kp=0.06
		Ki=0.001714
		Kd=0.525
			Kp=1.657
			Ki=0.198
			Kd=0.078

Charging valve	Kp=0.1
	Ki=0
	Kd=0
		Tu=300 sec
		Kp=0.06
		Ki=0.004
		Kd=2.25
			Kp=0.833
			Ki=2.522
			Kd=0.105

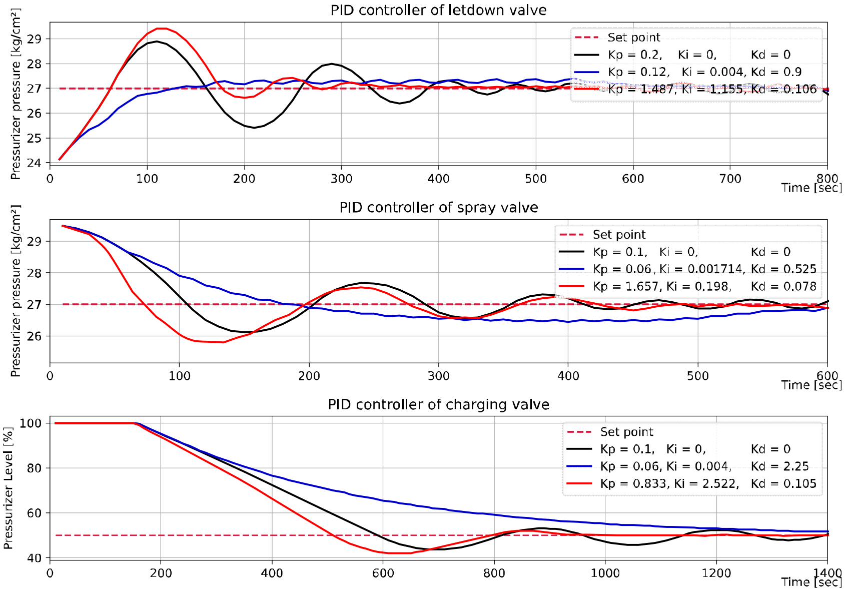


Fig. 22. Comparison of performances for the different tuning methods

5. Comparison of performances between DRL-based and PID-based controllers

A comparison of the performance of the developed DRL-based and PID-based controllers was conducted for the automation of the cold shutdown operation. The data were sampled from the simulator per second and the components could be manipulated every 10 seconds, which is considered enough time period for the pressure and level of the pressurizer to change. The data sampling frequency was chosen by taking into account the computation time (0.5 seconds) of the simulator and the time transmitted to the controller (0.1 milliseconds).

Fig. 23 shows the comparison of the performances in controlling the pressurizer pressure by the DRL-based and PID controllers. As shown in Fig 24, the DRL-tuned PID controller shows smaller accumulated error than the PID controllers tuned by the ZN and DRL-based controller. The comparison for the pressurizer level also shows similar results as shown in Fig. 25 and 26. The DRL-tuned PID controller shows the smallest error in the level. For the time to reach the desired state, it appears that the DRL-tuned PID controller is faster than the other controllers.

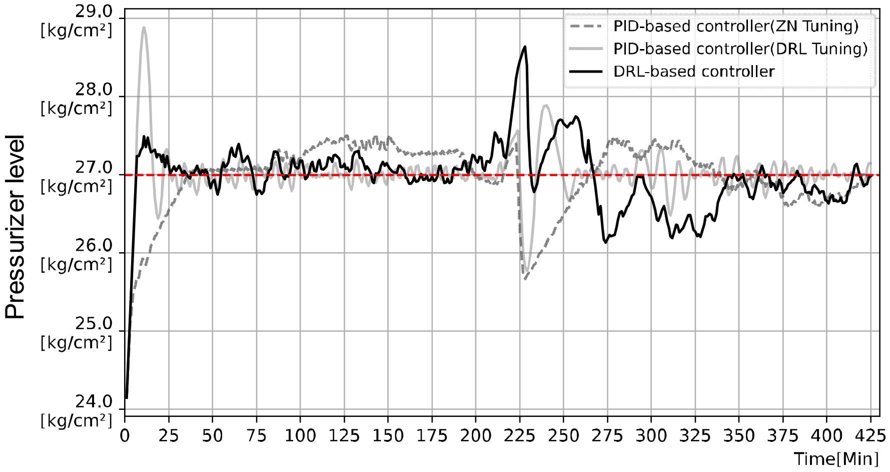


Fig. 23. Comparisons of controllers for the pressurizer pressure

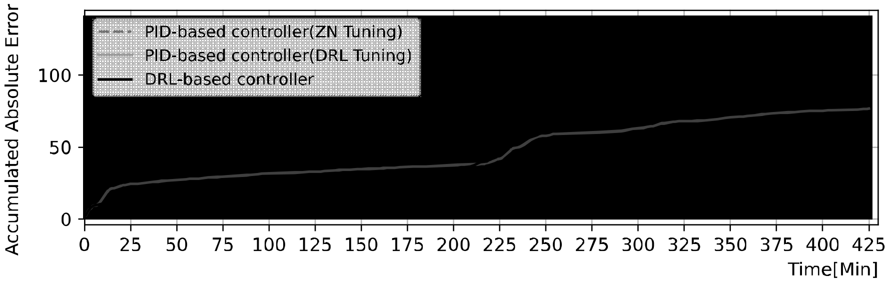


Fig. 24. Accumulated error for the pressure with the reference of 27 kg/cm^2

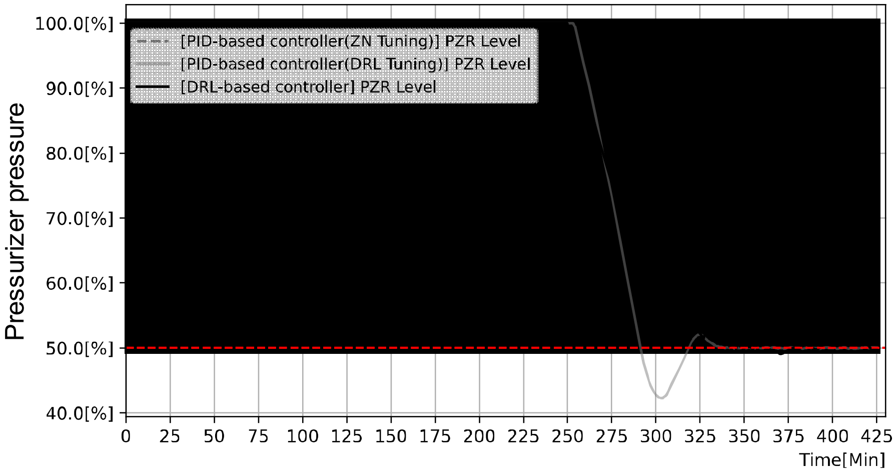


Fig. 25. Comparisons of controllers for the pressurizer level

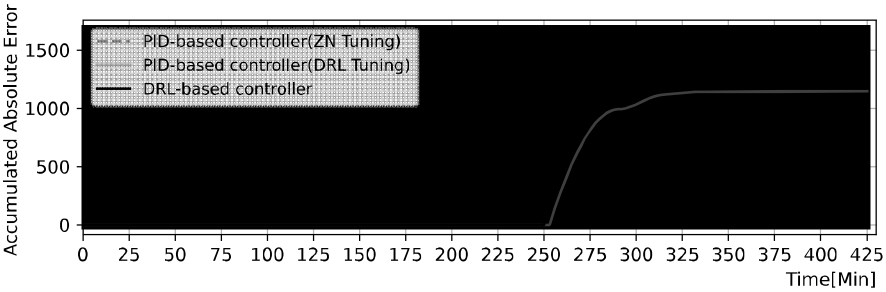


Fig. 26. Accumulated error for the level with the set-point of 50%

C. Power Increase Operation

1. Overview of the power-increase operation

To increase the power from 2% to 100%, two GOPs should be applied in the reference plant, namely, "Power operation greater than 2%" and "Secondary system heat-up and start-up", as presented in Fig. 6. The instructions for increasing the plant load from 2% to 100% are provided in the "Power operation greater than 2%" GOP, while the procedure "Secondary system heat-up and start-up" procedure describes the steps that are necessary for aligning and starting the secondary systems. These GOPs require the operators to operate components, such as the rod controller, turbine load controller, feedwater pumps, condenser pumps, steam generator feedwater valves, and synchronizer, based on the planned rate of power increase. Fig. 27 presents a simplified schematic diagram of the components that are related to the power-increase operation, and the operation’s initial and final conditions are presented in Table 7.

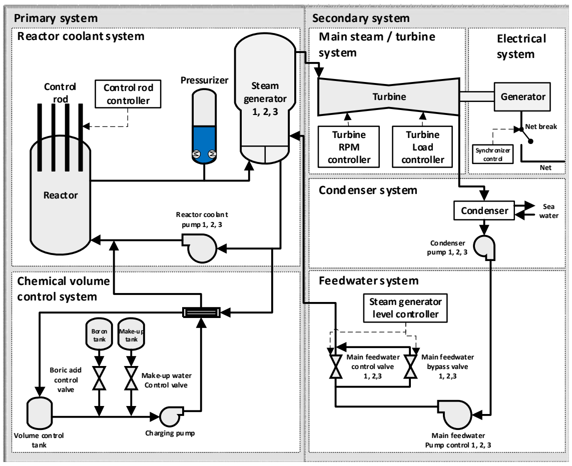


Fig. 27. Simplified schematic diagram of related components

Table 7. Initial and final conditions of the power-increase operation

Major parameter	**Initial condition**	**Final condition**
Reactor power	2%	100%
Electric power	0 MWe	900 MWe
Reactor coolant system (RCS) average temperature	294 ℃	306 ℃
Turbine revolutions per minute (RPM)	0	1800 RPM
Turbine load setpoint	0 MWe	900 MWe
Turbine load rate setpoint	0 MWe/min	2 MWe/min
Boron concentration	637 ppm	457 ppm
Rod position	211 Step (A Bank) 95 Step (B Bank) 0 Step (C Bank) 0 Step (D Bank)	228 Step (A Bank) 228 Step (B Bank) 228 Step (C Bank) 220 Step (D Bank)
Rod controller	Manual	Auto
Steam generator controller	Manual	Auto
Feedwater pump 1	On	On
Feedwater pump 2	Off	On
Feedwater pump 3	Off	On
Condenser pump 1	On	On
Condenser pump 2	Off	On
Condenser pump 3	Off	On
Synchronous connection	Disconnected	Connected

The operators’ tasks in the applicable procedures can be divided into 1) primary system control and 2) secondary system control. When conducing primary system control, the operators withdraw the control rods (reactor coolant system, Fig. 27) and manipulate the boron concentration (chemical volume control system, Fig. 27). At the beginning of the operation for stably increasing the power to 2%, the operators withdraw all control rods to the 100% position, which is the final condition, as specified in Table 7, and subsequently increase the boron concentration to maintain the reactor power at 2%. Once all the control rods have been withdrawn, the operators do not manipulate them further, and they reduce the boron concentration to increase the power from 2% to 100% by increasing the volume of the water from the make-up tank.

The rate of power increase (percent power per hour) is determined by considering the reactor cooling system (RCS) average temperature and the reference temperature. The reference temperature is the desired RCS temperature, which is predefined based on the current turbine load, while the RCS average temperature is the actual temperature in the primary side [72]. According to the procedure, during the power increase from 2 to 100%, the difference between the reference temperature and the RCS average temperature should be maintained within ± 1 ℃. This is only a recommendation and is not mandatory.

Operators must control several components of the secondary system. First, they increase the turbine speed to 1800 revolutions per minute (RPM) using the turbine RPM controller (the main steam/turbine system in Fig. 27). When the turbine and the reactor power reach 1800 RPM and 15%, respectively, the operators close the breaker to connect the generator to the grid and synchronize the frequencies (the electrical system in Fig.

27). In addition, the operators increase the turbine load setpoint, start the feedwater pumps, and start the condenser pumps concurrently with the reactor power increase in the primary system. The primary and secondary systems must be controlled harmoniously to avoid a reactor trip.

2. Task analysis of the power-increase operation

Based on a review of the "Power operation greater than 2%" and "Secondary plant heat-up and start-up" procedures, a task analysis was conducted to identify the tasks that should be automated by the algorithm that is proposed in this study. As presented in Table 8, this analysis identified a total of 21 control actions that are performed by the operators according to these procedures. Only the control-related actions were extracted for the development of the algorithm, although the procedures also provide monitoring actions, e.g., “confirm the RCS temperature is above 200 ℃.”

These actions were also categorized into three task types: Decision Making, Continuous Control, and Discrete Control. Decision Making task determines the rate of power increase; the subsequent control actions depend on this rate, although it does not include any control action. The continuous controls in this study adjust component states over a range to realize specified target values for the given parameters, and the rules that govern the necessary adjustments cannot be described with simple logic. For example, the operators adjust the RCS boron concentration to manipulate the power level. In contrast, a discrete control involves the direct setting of a target value based on a binary condition, as in if-then logic. An example of a discrete control is as follows: if the power level is 10%, then the turbine is set to 1800 RPM. The next section proposes an algorithm that can perform these actions.

Table 8. Operational tasks for increasing the reactor power

Step	Task Type	Action
1	Decision Making	Determine the rate of power increase in %/h
2	Continuous Control	Withdraw all control rods to the position of 100% reactor power while maintaining the reactor power at 2% through boration.
3	Continuous Control	If all the control rods are withdrawn, increase the reactor power from 2% to 6% – 10% by reducing the boron concentration.
4	Discrete Control	If the reactor power is 10%, the turbine RPM setpoint is 1800 RPM.
5	Discrete Control	If the reactor power exceeds 10%, the acceleration setpoint is 2 MWe/min.
6	Continuous Control	Adjust the boron concentration to increase the reactor power from 10% to 20%.
7	Discrete Control	If the reactor power is between 10% and 20%, the load setpoint is 100 MWe.
8	Discrete Control	If the turbine RPM is 1800 RPM and the reactor power exceeds 15%, push the net-breaker.
9	Discrete Control	If the reactor power is 20%, start condenser pump #2.
10	Continuous Control	Adjust the boron concentration to increase the reactor power from 20% to 100%.
11	Discrete Control	If the reactor power is between 20% and 30%, the load setpoint is 200 MWe.
12	Discrete Control	If the reactor power is between 30% and 40%, the load setpoint is 300 MWe.
13	Discrete Control	If the reactor power is 40%, start main feedwater pump #2.
14	Discrete Control	If the reactor power is between 40% and 50%, the load setpoint is 400 MWe.
15	Discrete Control	If the reactor power is between 50% and 60%, the load setpoint is 500 MWe.
16	Discrete Control	If the reactor power is 50%, start condenser pump #3.
17	Discrete Control	If the reactor power is between 60% and 70%, the load setpoint is 600 MWe.
18	Discrete Control	If the reactor power is between 70% and 80%, the load setpoint is 700 MWe.
19	Discrete Control	If the reactor power is 80%, start main feedwater pump #3.
20	Discrete Control	If the reactor power is between 80% and 90%, the load setpoint is 800 MWe.
21	Discrete Control	If the reactor power is between 90% and 100%, the load setpoint is 900 MWe.

3. Timeline of the power-increase operation

The timeline of the power-increase operation was analyzed to develop a normative operational strategy. This analysis considered the GOP's operational rules and the practical operational practices, which were determined from an interview with a senior reactor operator who works at a reference plant. Fig. 27 presents the timeline that was developed, which associates the desired operations with the reactor and electric powers, RCS temperatures and their differences from the reference temperature, and the control of related systems, such as the steam generator (SG) level, control rods, turbines, valves, and pumps.

The power-increase operation is divided into two operational ranges: 1) maintaining the reactor power at 2% and 2) increasing the reactor power from 2% to 100%. The objective of the first operational range is to adjust the positions of all control rods (Fig. 28 (d)) to 100% while maintaining the reactor power at 2% (Fig. 28 (a)); the average temperature is also maintained because it depends on the reactor power (Fig. 28 (b)). As the control rods are withdrawn, the reactor power increases, and increasing the boron concentration in the RCS reduces the reactor power. To maintain the reactor power at 2%, a boric acid-water solution is injected into the RCS, as illustrated in Fig. 28 (c).

The objective of the second operational range is to increase the reactor power from 2% to 100%, as represented by the red line in Fig. 28 (a). The operators determine the rate of the power increase (%/h); the power is increased by reducing the boron concentration in the RCS using make-up water (Fig. 28 (c)). The electric power is also increased to 100% by following a load setpoint that is increased stepwise. The RCS average temperature increases from 294 ℃ to 306 ℃, as illustrated in Fig. 28 (b). The difference between the RCS average temperature and the reference temperature should be maintained within ± 1 ℃, as represented by the gray area in Fig. 28 (b). This condition is applied after the start of the electrical power generation because the reference temperature is calculated based on the electrical power.

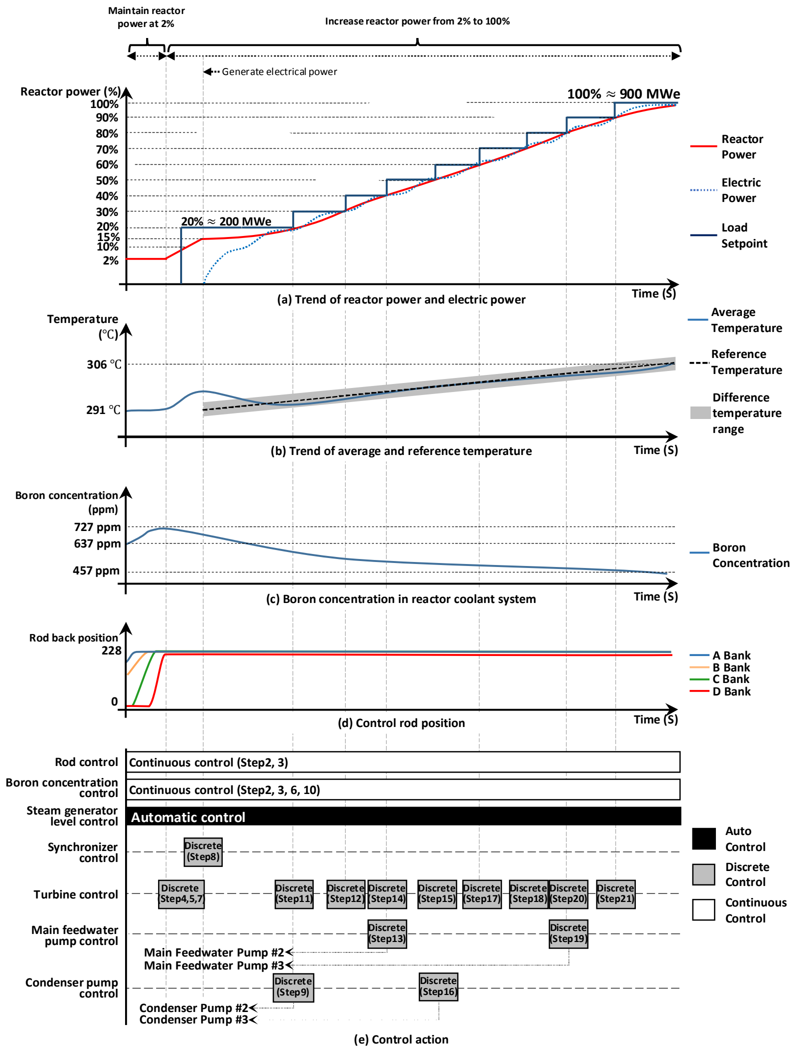


Fig. 28. Timeline for increasing the reactor power from 2% to 100%

To increase the reactor power, the operators manipulate seven systems, as illustrated in Fig. 28 (e). As described in Table 8, they withdraw the control rods and manipulate the boron concentration continuously, which corresponds to Steps 2, 3, 6, and 10. At 10% reactor power, in Steps 4, 5, and 7, the turbine RPM, acceleration setpoint, and load setpoint are adjusted to 1800 RPM, 2 MWe/min, and 100 MWe, respectively. Subsequently, the operators adjust the load setpoint with every 10% increase in the reactor power (Steps 11, 12, 14, 15, 17, 18, 20, and 21). At 15% reactor power, the plant and the grid are synchronized (Step 8). At 20% reactor power, condenser pump #2 is started (Step 9); condenser pump #1 is already running. Condenser pump #3 is started at 50% reactor power (Step 16). Main feedwater pumps #2 and #3 are started at reactor powers of 40% (Step 13) and 80% (Step 19), respectively; main feedwater pump #1 is already running. This study applies the pre-established automatic control algorithm for the SG level control.

4. Development of an algorithm for power-increase control

This paper presents an algorithm that employs a rule-based system and deep reinforcement learning to facilitate the autonomous increase of NPP power from 2% to 100% by controlling several systems. Fig. 29 illustrates the structure of the proposed algorithm, which consists of two modules: 1) a discrete control module and 2) a continuous control module. The discrete control module directs the synchronization, turbine, main feedwater pump, and condenser pump controls, for which rule-based systems can be developed based on the operating procedures.

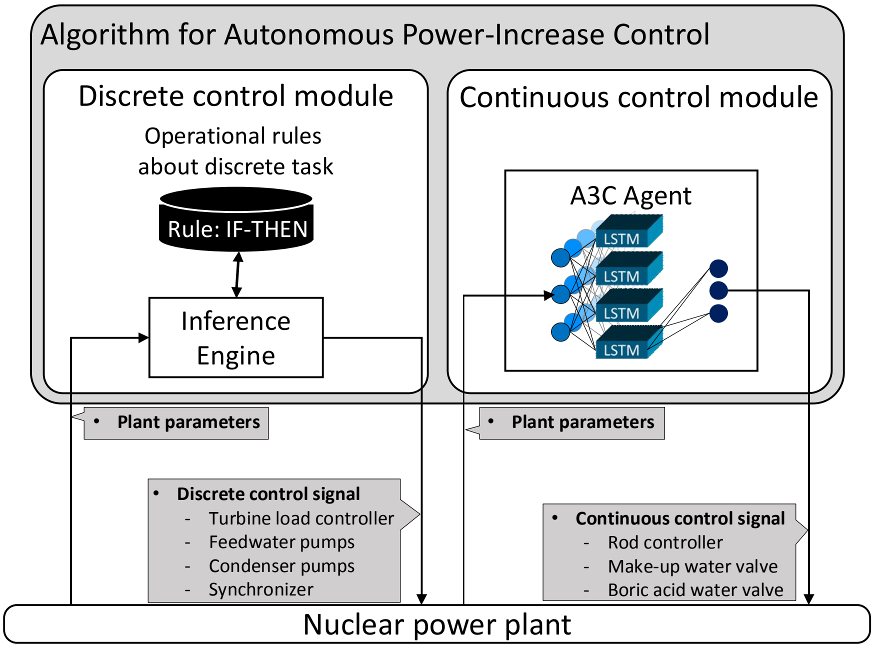


Fig. 29. Overview of the algorithm for the power-increase operation

The continuous control module dictates the adjustment of the control rods and the RCS boron concentration. The associated procedures do not specify rules for the operators; e.g., they do not specify the number of steps in which the control rod should be withdrawn or the volumes of make-up or boric acid water that should be added. The procedures specify only the objective of the control activity, e.g., "increase the power to 20% by altering the control rod position or RCS boron concentration.“

Deep reinforcement learning was deemed suitable for use as the continuous control module. A neural network and a training algorithm are selected by considering the characteristics of the operational steps in NPPs. The types of control for NPPs are regulatory control (e.g., adjustment of valve position) and discrete control (e.g., on/off control). For discrete control, the set-point and operating conditions are specified in detail in the operating procedure. Operators can conduct discrete control according to rules that are specified in the operating procedures. In contrast, only operational target values are provided for regulatory control. Accordingly, regulatory control is based on the operator's experience, which includes monitoring previous and current plant conditions. The target of the continuous control module is the requlatory control. Thus, this study attempted to implement controls in accordance with the operator's behavioral pattern through trial and error using a long short-term memory (LSTM) and an asynchronous advantage actor-critic (A3C) algorithm.

(1) This study used a LSTM network, a kind of recurrent neural network (RNN), by considering the characteristics of the plant parameters. The trends of the plant parameters are well known to be the same as that of time series data. To extract and analyze meaningful information, e.g., the timing of an AI agent's action, from time-series data, it is important to identify the correlations between previous and current data. The output of an LSTM can be calculated by considering previous data, in contrast to other neuronal networks such as convolutional neural networks and vanilla feedforward neural networks.

Moreover, LSTM not only stores the values that are calculated from the previous time data in the LSTM cell but also considers previously saved values when calculating the next time data. The author's previous studies showed that the LSTM can support well the operation of nuclear systems [5, 57] as well as the diagnosis of events [56, 73]. Moreover, to better support the selection of the LSTM neural network, this study compares the performance of other neural networks such as deep neural network (DNN), convolutional neural network (CNN), LSTM, and C-LSTM(CNN + LSTM).

(2) An asynchronous advantage actor-critic (A3C) algorithm was quickly trained in the specified domain. The A3C algorithm is well known for fast training due to parallel actor-learners that are based on the central processing unit's (CPU's) multiple threads and the asynchronous network update. This study used a nuclear simulator to test and train an AI agent. This simulator does not recommend calculation acceleration with a stable calculation performance. As a result, the AI agent takes more than 14 hours per episode to train the entire power increase operation. To solve this problem, we not only built multiple environments but also applied a parallel training algorithm, namely, A3C.

The goal of the continuous control module is to select actions necessary to meet the operational goals of the sequential plant states. The continuous control module with A3C algorithm can find an operational path in parallel. An operational path is a set of actions for controlling a component to achieve flexible operating goals that are assigned by the operators. A reward algorithm was developed for training the agent, and an LSTM network was used for selecting the actions necessary to meet the operational goals of the sequential plant states.

a. Design of the discrete control module using if-then logic

A rule basis for discrete control was developed for the synchronizer, turbine, main feedwater pump, and condenser pump controls by transforming the operating procedures into if-then rules, which are presented in Table 8. The tasks that are identified as discrete controls in Table 9 were analyzed and categorized into four functions based on the controlled system, and the applicable rules were extracted from the procedures' task instructions. The inputs and outputs that were required for the module to control the tasks were identified. An input is a plant parameter that must be obtained to correctly determine the control action that is needed for accomplishing a task, while an output is the control action that will be performed as a result.

Table 9. Discrete control module if-then rules for increasing the reactor power from 2% to 100%

Function	**Rule Number(s)**	**If-then Rule**	**Input(s)**	**Output(s)**
Synchronizer control	1	If the turbine RPM is 1800 RPM and the reactor power is greater than 15%, push the net-breaker button.	Reactor power, Turbine RPM	Net-breaker button control
Turbine control	2	If the reactor power is 10%, the turbine RPM setpoint is 1800 RPM.	Reactor power, Turbine RPM	Turbine RPM setpoint control
	3	If the reactor power is greater than 10%, the acceleration setpoint is 2 MWe/min.	Turbine acceleration	Turbine acceleration setpoint control
	4	If the reactor power is between 10% and 20%, the load setpoint is 100 MWe.	Reactor power, Load setpoint	Load setpoint control
	5 - 11	...	...	...
	12	If the reactor power is between 90% and 100%, the load setpoint is 900 MWe.	Reactor power, Load setpoint	Load setpoint control
Main feedwater pump control	13	If the reactor power is 40% and the state of the main feedwater pump 1 is “activated,” start main feedwater pump 2.	Reactor power, Main feedwater pumps 1 and 2 states	Main feedwater pump 2 control
	14	If the reactor power is 80% and the state of main feedwater pump 2 is "activated," start main feedwater pump 3.	Reactor power, Main feedwater pumps 2 and 3 states	Main feedwater pump 3 control
Condenser pump control	15	If the reactor power is 20% and the state of condenser pump 1 is "activated," start condenser pump 2.	Reactor power, Condenser pumps 1 and 2 states	Condenser pump 2 control
	16	If the reactor power is 50% and the state of condenser pump 2 is "activated," start condenser pump 3.	Reactor power, Condenser pumps 2 and 3 states	Condenser pump 3 control

b. Design of the continuous control module using SAC

The A3C agent for continuous control aims at managing the reactor power by manipulating the control rods and boron concentration, and, if fully trained, can manage the reactor power based on a specified rate of power increase and the obtained plant parameters. The A3C agent's strategies relate to three operational strategies: increase power, decrease power, and stay.

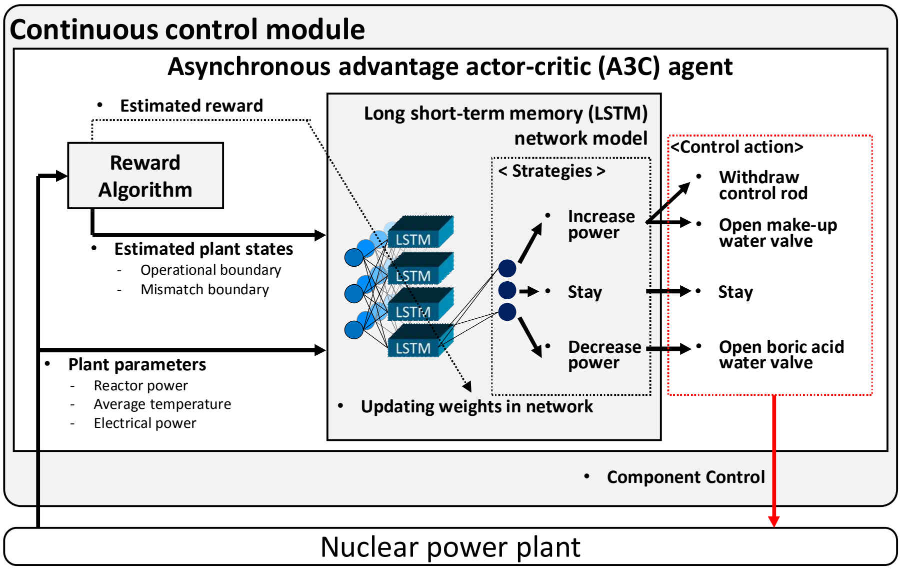


Fig. 30. Overview of the continuous control module

Fig. 30 illustrates the overall structure of the A3C agent for continuous control, which consists of a reward algorithm and an LSTM network model. The reward algorithm evaluates the obtained plant parameters to determine whether and the degree to which the prior operation or action of the A3C agent was successful, and this reward is used to update the weights in the LSTM network model. The LSTM network model generates an operational strategy using the obtained and evaluated plant parameters. Then, the A3C agent selects the option that is associated with the highest probability value from among the available outputs of the LSTM network: increase, decrease, or stay.

The operational strategies comprise the control actions that are required for realizing the objective of each strategy. For example, for the "stay" strategy, the A3C agent stops manipulating components, and the boric acid water valve is opened to increase the boron concentration and, therefore, decrease the reactor power. The strategies for "power increase" consist of two control actions; the A3C agent withdraws the control rods and changes the control action to the opening of the make-up water control valve to reduce the boron concentration.

c. Design of the reward algorithm

In DRLs, the reward is an essential element that is used to update the weights of the A3C agent; learning by the agent is associated with updating the weights of the network to maximize the accumulative reward [74]. The reward algorithm evaluates the agent’s behavior based on a specified state in the environment to determine the reward. Therefore, the reward algorithm guides the agent to obtain a high accumulative reward in the target domain [75]. To find the best operational path, the use of operational guidelines or boundaries is a suggested for designing a reward algorithm [76]. Furthermore, if the operational goal is more than one, like in the multi-objective problems, Garduno-Ramirez and Lee [77] proposed defining the upper and lower boundaries for each operational goal. In this study, the specified operational objectives were used to design the reward algorithm for increasing the reactor power.

This study proposes a reward algorithm that is designed for training the proposed A3C agent to increase the reactor power. It has two reward criteria, which are based on the reactor power and the average temperature. Fig. 31 presents the criteria for providing a reward via the proposed reward algorithm. The first reward criterion is related to the reactor power. As illustrated in Fig. 31, two bandwidths were applied. While maintaining the reactor power at 2% (the blue area in Fig. 31), the reward boundary was defined as ± 1% of the reactor power, namely, 1% to 3%. During the power increase after reaching 2% reactor power, the bandwidth was determined by the following linear equations that were based on the pre-determined rate of power increase (the red area in Fig. 31). The upper boundary was 3% at 2% reactor power and 110% at 100% reactor power, while the lower boundary was 1% at 2% reactor power and 90% at 100% reactor power.

$$End of\ Operation\ Time\ (t_{100}) = t_2 + \frac{100-2}{\mathrm{Pr}} \tag{8}$$

$$Upper\,Boundary = \begin{cases} 3 & (t_2 \geq t) \\ \dfrac{100-3}{t_{100}-t_2}(t-t_2)+3 & (t_{100} \geq t > t_2) \\ 110 & (t > t_{100}) \end{cases} \tag{9}$$

$$Lower\,Boundary = \begin{cases} 1 & (t_2 \geq t) \\ \dfrac{100-3}{t_{100}-t_2}(t-t_2)+1 & (t_{100} \geq t > t_2) \\ 90 & (t > t_{100}) \end{cases} \tag{10}$$

Pr : Predefined Rate of Power Increase (%/h)
t : Time
t_2 : Time at All Rods 100% Withdrawal
t_{100} : End of Operation Time

The power reward was calculated as the difference between the current power at time t and the most desirable power, which was the predefined power at that time and is represented by the dashed line in the center of the reward boundary in Fig. 31. The power reward was calculated via Equation (11) by using a normalized value of the distance. The reward was maximal, namely, 1, when the current power was equal to the predefined power, while it was 0 when the current power was located on the upper or lower boundary. For instance, at t = 8 h in Fig. 31, when the reactor power increased from 2% at 5 h to 100% at 103 h at a 1%/h rate of increase, the reactor power, the predetermined power that was based on the rate of power increase, and the upper boundary were 6%, 4.99%, and 6.27%, respectively. The resulting reward was 0.21 by R=1 - (6 - 4.99)/(6.27 - 4.99). Similarly, at t = 10 h and P = 5.6%, the reward was 0.04, as presented in Fig. 31.

$$Power\,Reward\,(0\sim 1)=\begin{cases}0 & (P>R_{up})\\ 1-\dfrac{P-R_{mp}}{R_{up}-R_{mp}} & (R_{up}\geq P>P_{mp})\\ 1 & (P=R_{mp})\\ 1-\dfrac{R_{mp}-P}{R_{mp}-R_{lp}} & (R_{mp}>P\geq R_{lp})\\ 0 & (P<R_{lp})\end{cases}\qquad(11)$$

P : Current Power at Time t (%)
R_{mp} : Middle of Power Reward Boundary,
i.e., Pre-determined Power at Time t
R_{up} : Upper Power Reward Boundary
R_{lp} : Lower Power Reward Boundary

If the power moved outside the boundary, the training was terminated. In addition, the agent stopped the training when it realized the objective of the operation, namely, when the reactor power was 100%.

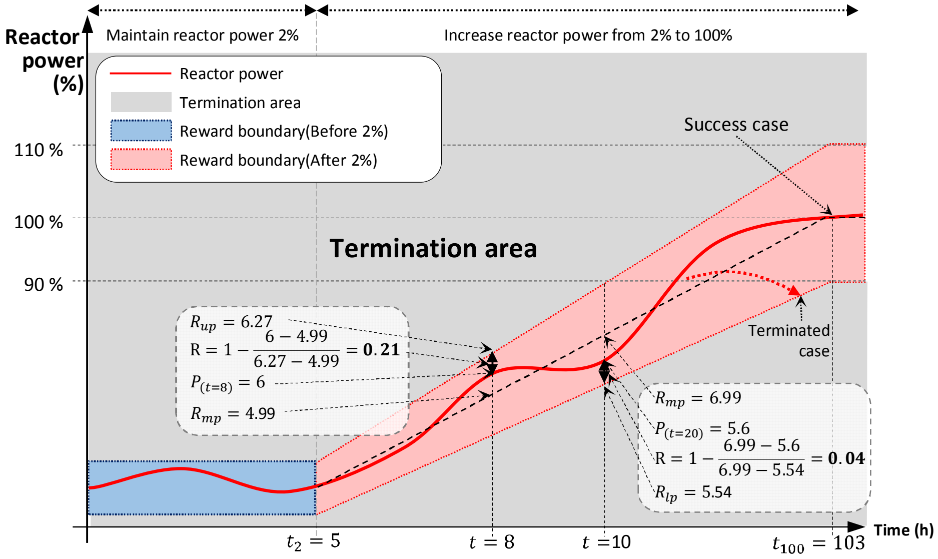


Fig. 31. Power reward for the A3C agent

The second reward criterion relates to the difference between the average temperature and the reference RCS temperature that is provided by the GOP. This reward represents that the rule that the average RCS temperature should be controlled by the agent to within ± 1 ℃" of the reference RCS temperature (the gray area in Fig. 32). Since the reference temperature is calculated based on the current turbine load (MWe), the upper and lower limits of this reward boundary are calculated after the electrical power generation has begun.

Similar to the power reward, the temperature reward was also calculated via Equation 12 based on the difference between the current temperature at time t and the most desirable temperature, namely, the reference temperature. The maximal reward, namely, 1, was obtained when the average RCS temperature was equal to the reference temperature. In contrast to the power reward, if the average RCS temperature moved outside the boundary, the training was not terminated; instead, the reward had a negative value that was proportional to the distance from the closest boundary, with -1 being the lowest possible value.

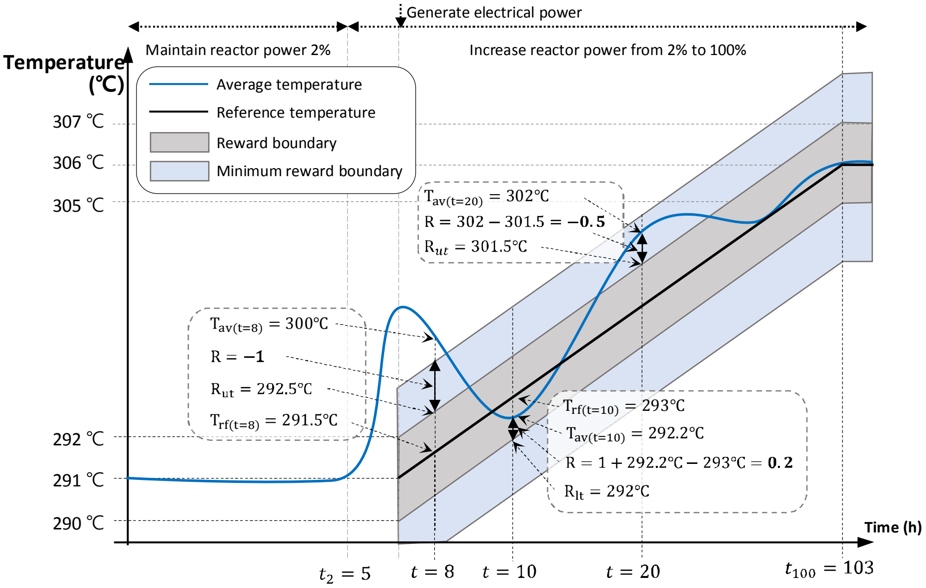


Fig. 32. Temperature reward for the A3C agent

As shown in Fig. 32, when the average RCS temperature was between the upper and lower boundaries, a positive reward was returned and was inversely proportional to the distance from the reference temperature (as shown at t = 10 h in Fig. 32). Outside this boundary and up to a difference of ± 2 ℃, a negative reward was given proportional to the distance to the closest boundary (as shown at t = 20 h in Fig. 32). If the temperature difference was greater than 2 ℃, the reward was -1.

The total reward was calculated as the arithmetic mean of the power and temperature rewards, as expressed in Equation 13. The agent conducted the training to obtain the largest total reward for each episode and, in the process, was incentivized to shift the reactor power and the average RCS temperature to the middle values of the reward boundaries. The episode continued until the reactor power reached 100% or moved outside the reward boundary.

$$Temperature\,Reward(-1\sim 1)=\begin{cases}-1 & (R_{ut}+1<T_{av})\\ -T_{av}+R_{ut} & (R_{ut}+1\geq T_{av}>R_{ut})\\ 1-T_{av}+T_{rf} & (R_{ut}\geq T_{av}>T_{rf})\\ 1 & (T_{av}=T_{rf})\\ 1+T_{av}-T_{rf} & (T_{rf}>T_{av}\geq R_{lt})\\ T_{av}-R_{lt} & (R_{lt}-1\leq T_{av}<R_{lt})\\ -1 & (R_{lt}-1>T_{av})\end{cases} \tag{12}$$

(13)

T : Average RCS Temperature at Time t

T_{rf} : Middle of Temperature Reward Boundary, i.e., Reference Temperature at Time t

R_{ut} : Upper Temperature Reward Boundary (T_{rf}+ 1) at Time t

R_{lt} : Lower Temperature Reward Boundary (T_{rf}−1) at Time t

d. LSTM network modeling

Fig. 33 illustrates the proposed LSTM network of the continuous control module's A3C agent for producing an operational strategy (increase, decrease, or stay). The final control action of the continuous control module is selected based on the reactor power and the operational strategy. Each operational strategy maps to the required control action. For example, the decrease strategy is mapped to the opening of the boric acid water valve. If the output strategy of the LSTM network is "stay," the A3C agent does not control the component. In the increase strategy, the A3C agent selects a control according to the current operational objective:

- Withdraw the control rod (when maintaining the reactor power at 2%) or

- Open the make-up water valve (when increasing the reactor power from 2% to 100%).

The proposed LSTM network model consists of an input layer, an LSTM layer, and an output layer. The sizes of the input and output layers can be defined based on the numbers of plant parameters and control actions, respectively. The number of LSTM cells is determined by the time window.

The input layer of the investigated LSTM network had a 10-step time window, which considered the trends of plant parameters by exploiting the collected historical data. The historical data were sampled from the simulator every 30 s to optimize the dataset size; the trends that were observed when the data were collected every second did not differ significantly. The A3C agent used the current and previous states as a two-dimensional array and as a training dataset, which included the plant parameters for 300 s. At each time window, the LSTM network used eight input parameters, namely, four plant parameters (reactor power, average temperature, reference temperature, and electric power) and four variables that represented the distances of the current power and average RCS temperature from their upper and lower boundaries.

At the LSTM network's output layer, the probability of each operational strategy was generated using a softmax function, which can map a network's output to a probability distribution between 0 and 1; the sum of the generated output values is one. If the A3C agent selected the strategy with the highest probability among the operational strategies, it received a large reward or realized the operational objective. Finally, the A3C agent selected a control action based on the selected operational strategy. The detailed structure and hyperparameters of the LSTM network were determined as illustrated in Fig. 33 through an experimental optimization.

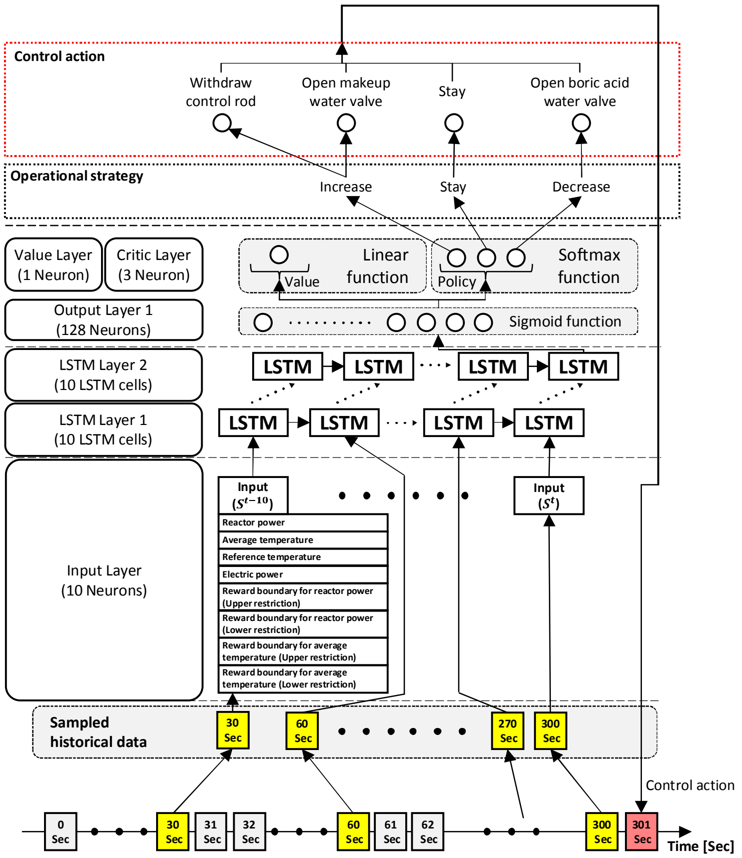


Fig. 33 The structure of the LSTM network for the A3C agent

5. Experiments

a. Training environment

CNS was also used as a real-time testbed for training and validating the proposed autonomous power increase algorithm. Fig. 34 shows the A3C agent training environment structure, which consists of four desktop computers-one main computer and three sub-computers. One main agent and sixty local agents for implementing the proposed algorithm were installed on the main computer. The CNS was installed on the three sub-computers. Each sub-computer could run 20 CNS simulations at a time; therefore, a total of 60 simulations could be conducted simultaneously. The A3C global network was trained, while the A3C training algorithm was trained using 60 threads of CPUs. The A3C agent was developed based on the Python programming language with the TensorFlow and Keras machine learning libraries

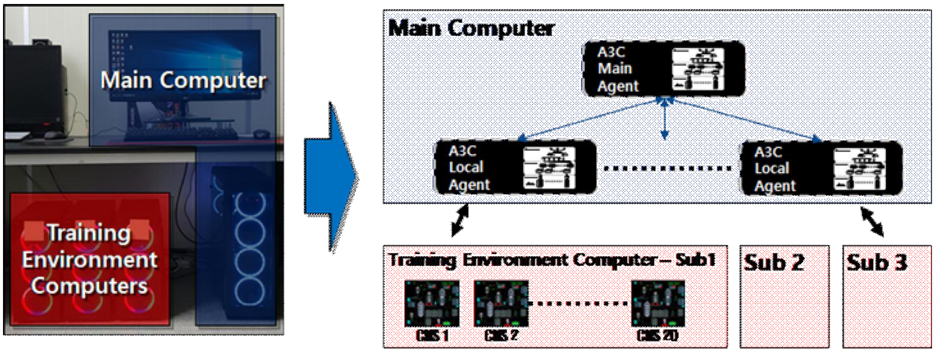


Fig. 34. Structure of the training environment for the A3C agent

b. Training and stability for the entire power-increase operation

For a complete (from 0% to 100%) power-increase operation at a rate of 3%/h, the A3C agent was trained in 8800 episodes. The A3C agent training was complete when the average maximum probability converged to a specified value or when the value became stable.

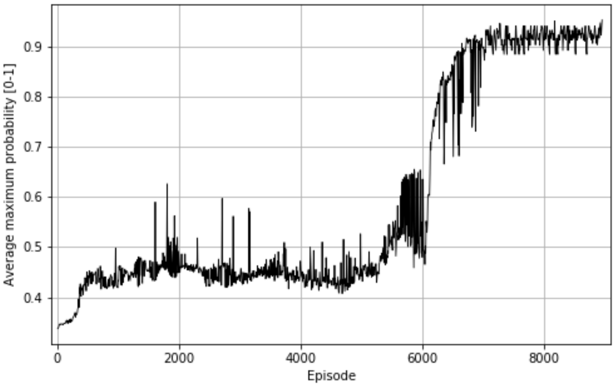


Fig. 35 Average maximum probability per episode for the A3C LSTM network

Fig. 35 presents the trend in the average maximum output probability per episode over time. The A3C network approached a stable probability (larger than 0.9) after approximately 7500 episodes. Fig. 36 shows the trend of the rewards that were obtained by the A3C agent as the number of episodes increased. In one episode, the theoretical maximum cumulative reward during the entire power-increase operation was 4800 (the green dashed line in Fig. 36); this is because the largest reward for a training dataset was 1, and the total number of datasets that increased the reactor power to 100% over 144 000 s at the rate of 3%/h, plus an additional margin of 4000 s, was 4800. The maximum practicably feasible reward for power-increase operation success was observed to be 3000.

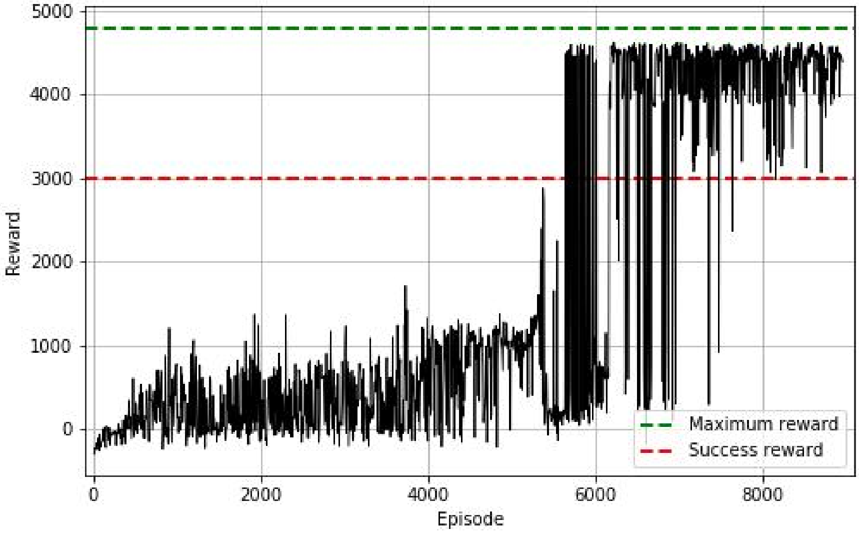


Fig. 36. Rewards obtained by the A3C LSTM network

This study identifies a network that can be quickly trained in the specified domain since the A3C network requires more than 14 hours per episode to train the entire power increase operation. In this study, the considered networks are the deep neural network (DNN), convolutional neural network (CNN), LSTM, and C-LSTM (CNN + LSTM). DNN is a typical feed-forward neural network that contains many hidden layers of nonlinear hidden units and a very large output layer. In CNN, the hidden layers have fewer connections and parameters because filters that perform convolution operations are utilized. CNN has been demonstrated to outperform DNN in feature extraction from input data. LSTM can calculate time-sequential input data for units that are called constant error carousels. It can facilitate the memorization of important events or long-term data. C-LSTM is a combined model of CNN and LSTM. This network has been proposed for extracting features of data and for handling time-sequential data.

Table. 10 Architectures of the compared networks

Network	Network layer	Layer type	Time-sequence	Node	Parameter
DNN	Common	Input layer	-	8	0
		Dense	-	32	224
		Dense	-	64	2112
		Dense	-	70	4550
	Actor	Dense	-	64	4544
		Output layer	-	3	195
	Critic	Dense	-	32	2272
		Output layer	-	1	33
CNN	Common	Input layer	10	8	0
		Conv1D	10	10	190
		Max pooling	3	10	0
		Flatten	-	30	0
		Dense	-	64	1984
		Dense	-	70	4550
	Actor	Dense	-	64	4544
		Output layer	-	3	195
	Critic	Dense	-	32	2272
		Output layer	-	1	33
LSTM	Common	Input layer	10	8	0
		LSTM	-	32	4992
		Dense	-	64	2112
	Actor	Dense	-	64	4160
		Output layer	-	3	195
	Critic	Dense	-	32	2080
		Output layer	-	1	33
C-LSTM	Common	Input layer	10	8	0
		Conv1D	10	10	190
		Max pooling	3	10	0
		LSTM	-	32	5504
		Dense	-	60	1900
	Actor	Dense	-	64	3904
		Output layer	-	3	195
	Critic	Dense	-	32	1952
		Output layer	-	1	33

To train these networks under the same conditions, they should have the same number of parameters. The parameters at each layer of the network model are arranged with a normal distribution (mean = 0.0 and standard deviation = 1.0), which supports stable training under the same conditions. Table 10 describes the architectures of the networks that are used in the A3C algorithm for the experiment. Each network consists of three layers: common, actor, and critic. The actor and critic layers are linked to the common layer.

Before training on the entire power increase operation, the A3C agent is trained between 2% and 15% power to identify the optimal network. Each network has been trained by 6500 episodes. Fig. 37 shows the trend of the duration of each network versus the number of episodes. Each line represents the average duration over 10 episodes. The agent's objective is to increase the power within the operational boundary, which is the power reward boundary in this paper, for 600 seconds. For strict comparison of these networks, an operation with a duration of less than 600 seconds is regarded as a failed operation. These networks are trained until the average duration is 600 seconds. In Fig. 37, the LSTM network is the best performing network as it realized an average duration of 600 seconds in 6500 episodes. The second-best performing network is CNN, which realized a duration of approximately 400 seconds in 6500 episodes. C-LSTM and DNN show poor performance (durations of less than 250 seconds). The results of this experiment demonstrate that the LSTM network can realize the operational objective in fewer training episodes than the other networks.

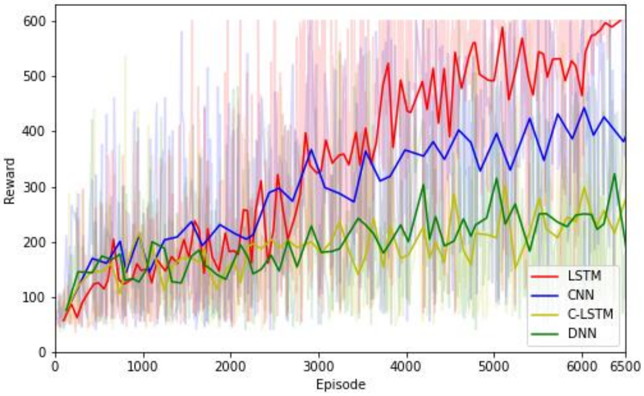


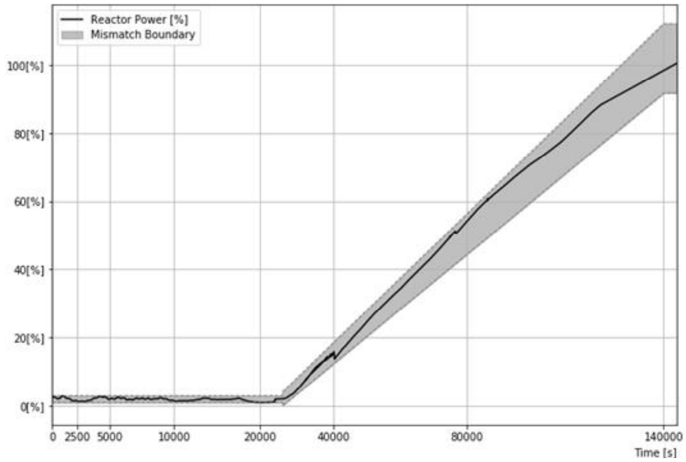
Fig. 37. Average duration of each network

c. Experimental results

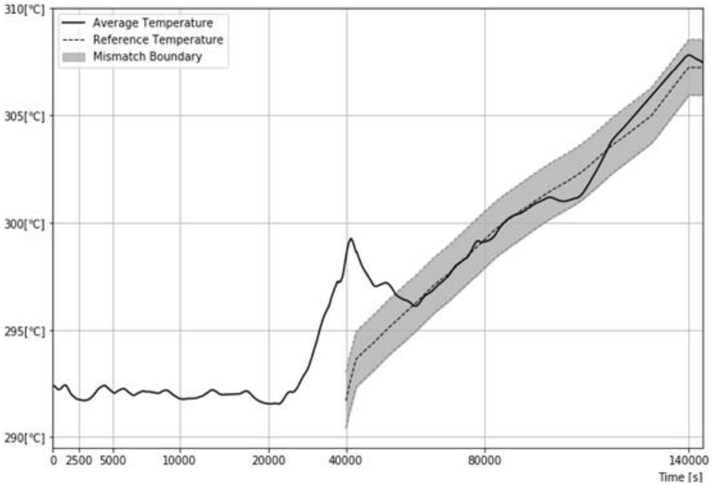
After the algorithm for autonomous power increase control was trained, an experiment was conducted to demonstrate that the proposed algorithm could autonomously increase the power at a specified rate. The continuous control module was implemented using an A3C and an LSTM network, while the discrete control module was implemented with a rule-based system. Fig. 38 (a-h) presents the experimental results for a 3.0%/h rate of power increase, which demonstrate that the proposed algorithm can increase the power at the intended rate within the operational boundary (Fig. 38 (a)). In addition, Fig. 38 (b) shows that the proposed algorithm managed the average temperature within the mismatch boundary from the reference temperature over the reactor power of 30% and could effectively restore an increased or decreased average temperature to within the mismatch operation range. The changes in the average temperature that were observed at approximately 40 000 s were due to connecting to the grid and starting a condenser pump, which impacted the overall plant state.

The continuous control module also managed the boron concentration during the power increase; the results are presented in Fig. 38 (c) and (d). To maintain the power at 2%, the boron concentration was increased to compensate for the effect of the control rod withdrawal, which occurred at approximately 22 000 s, as shown in Fig. 38 (e). Then, the controller decreased the boron concentration by increasing the volume of the make-up water to increase the reactor power from 2% to 100%.

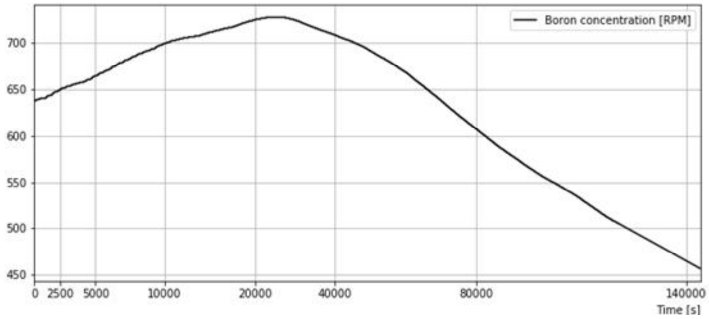
The discrete control model operated the system's synchronous connection to connect to the electrical grid at a reactor power of 15%. The discrete control module also selected the turbine load (Fig. 38 (f)) and RPM setpoints (Fig. 38 (g)) based on the reactor power. Additional actions that were performed by the discrete control module during the power-increase operation are presented in Fig. 38 (h) and include starting feedwater pumps 2 and 3 and condenser pumps 2 and 3 to circulate feedwater in the secondary part of the plant. The control module started these pumps in sequence according to the general operating procedure.



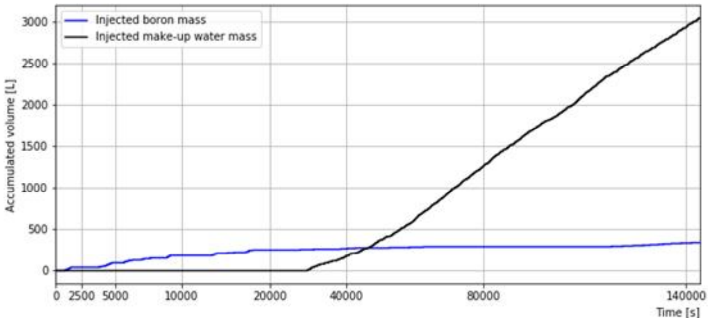
(a) Simulation results: Reactor power



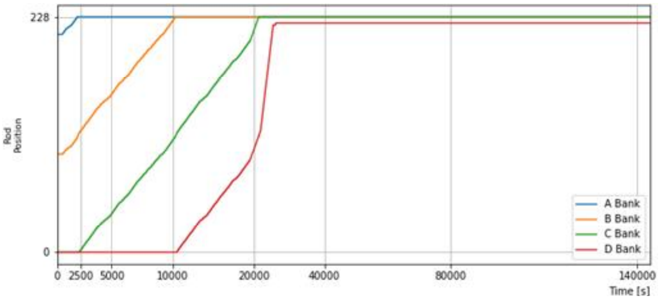
(b) Simulation results: Average and reference temperatures



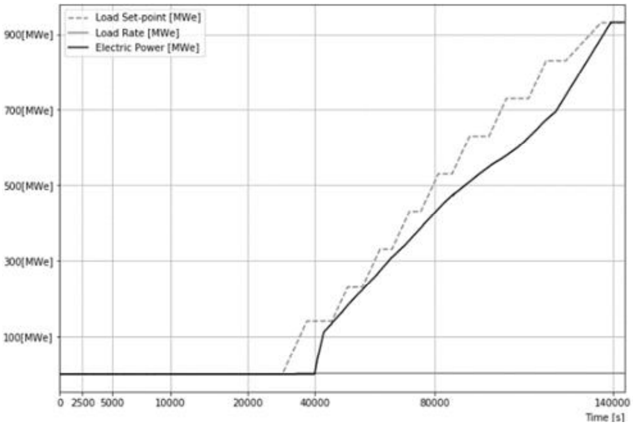
(c) Simulation results: RCS boron concentration



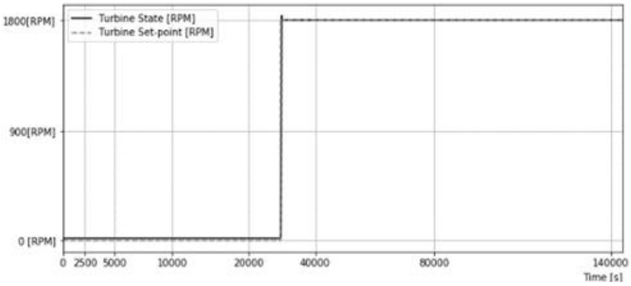
(d) Simulation results: Injected masses of boron and make-up water



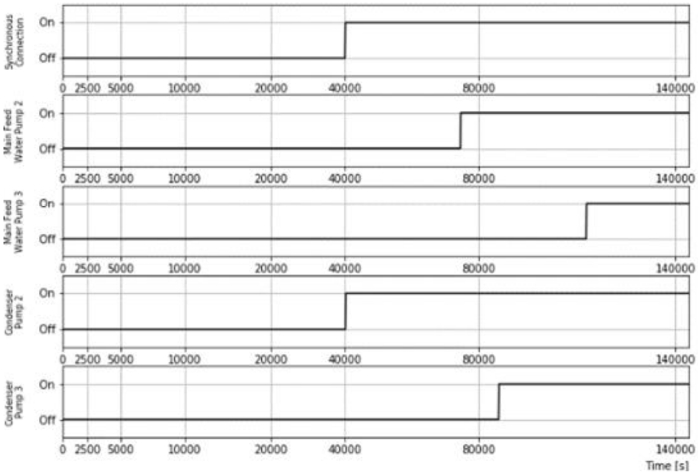
(e) Simulation results: Injected masses of boron and make-up water



(f) Simulation results: Turbine load and electric power



(g) Simulation results: Turbine RPM and turbine RPM setpoint



(h) Simulation results: Pump and synchronous control signals

Fig. 38. Simulation results for a 3%/h autonomous power-increase operation

IV. Emergency Operation

This section aims to develop an emergency operation agent that can reduce the primary pressure and temperature safely until the shutdown cooling entry condition after reactor trip caused by the loss of coolant accident (LOCA) in NPPs. The suggested agent uses Soft Actor-Critic (SAC) algorithm and a deep neural network. SAC is a DRL method that optimizes a stochastic policy in an off-policy way. This suggested algorithm has also proven its data efficiency and learning stability as well as hyper-parameter robustness. In order to identify the agent's inputs/outputs, the functional recovery procedures (FRPs) are analyzed through an abstraction decomposition space (ADS). ADS can help to represent the entire target domain and draw constraints on the given domain through a step-down decomposition. Based on identified constraints, this study designs reward algorithms for providing training directions for the agent. The test results using a compact nuclear simulator (CNS) indicates that the suggested emergency operation agent can control the components to comply with identified constraints until the shutdown cooling entry condition.

A. Emergency operation analysis

NPP operating strategies during emergency situations were analyzed to develop an autonomous operating agent. FRPs were analyzed for identifying operation goals and criteria, required systems and components, and success paths to mitigate the emergency. Then, the identified information was mapped into the table of ADS. As a result of ADS, the tasks of the agent and reward criteria were defined.

1. Emergency operation analysis based on FRP

Based on the FRP, this study identified the goal and criteria of each safety function, system, and component required in the emergency operation. The emergency operating procedure in Korean NPPs can be divided into the event-based procedure (optimal recovery procedure) and symptom-based procedure (functional recovery procedure) [78]. Optimal recovery procedure (ORP) is designed to cover specific design basis accidents (DBAs), such as loss of coolant accident (LOCA) and steam generator tube rupture (SGTR). On the other hand, FRP is focused on the recovery of safety functions. FRP provides operator actions for events in which a diagnosis is impossible, or any ORP is unavailable. The actions of FRPs are to ensure that safety functions are placed in a stable, safe condition. Fig. 39 shows the flow of the emergency operation strategy.

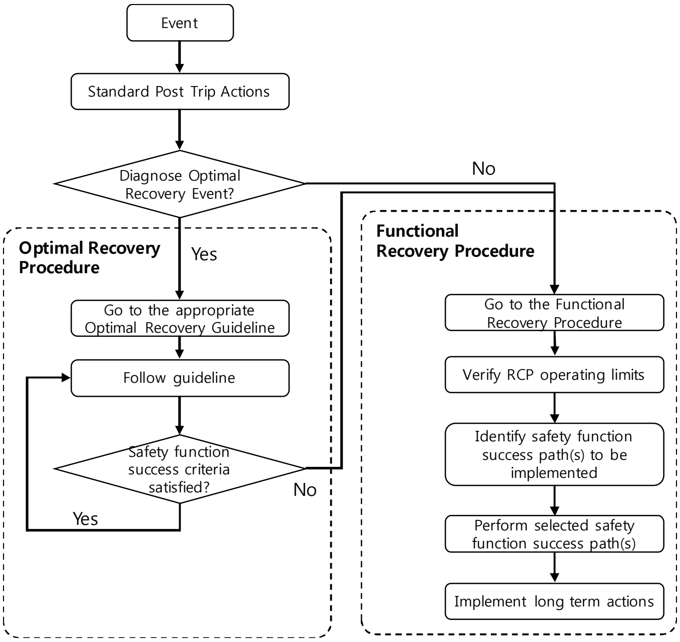

Event
Standard Post Trip Actions
Diagnose Optimal Recovery Event?
No
Yes
Optimal Recovery Procedure
Go to the appropriate Optimal Recovery Guideline
Follow guideline
Safety function success criteria satisfied?
No
Yes
Functional Recovery Procedure
Go to the Functional Recovery Procedure
Verify RCP operating limits
Identify safety function success path(s) to be implemented
Perform selected safety function success path(s)
Implement long term actions

Fig. 39. Strategy flow chart for emergency operation

This study analyzed safety functions, the required systems, and components. Table 11 shows the nine safety functions and their purposes. Table 12 represents the safety systems and components designed to satisfy RCS inventory control function in Korean NPPs [78].

Table 11. Nine safety functions

No	Safety function	Purpose
1	Reactivity control	Shut reactor down to reduce heat production
2	Reactor coolant system (RCS) inventory control	Maintain volume or mass of reactor coolant system
3	RCS pressure control	Maintain pressure of reactor coolant system
4	RCS heat removal	Transfer heat out of coolant system medium
5	Core heat removal	Transfer heat from core to a coolant
6	Containment isolation	Close valves penetrating containment
7	Containment pressure and temperature control	Keep from damaging containment
8	Hydrogen control	Control hydrogen concentration
9	Maintenance of vital auxiliaries	Maintain operability of systems needed to support safety systems

Table 12. Safety systems and components designed for safety RCS pressure control

System	Component
Safety depressurization and vent system	Power-operated relief valve (PORV)
Pressurizer (PZR) pressure control system	PZR spray valve, PZR heater

2. Work domain analysis by using abstraction decomposition space

The ADS is used to systematically identify the systems and components that the agent is required to manipulate. The operational goal and constraints during the emergency operation were also analyzed to design the agent's reward algorithm.

ADS can analyze the given work domain as the abstraction level and decomposition space. The abstraction level is a hierarchical structure that consists of functional purpose, abstraction function, generalized function, and physical function. These levels are connected with mean-end links that show how-what-why relationships between levels. On the other hand, the decomposition space is typically divided into a whole system, sub-system, and component. It can represent the entire domain under examination, stepping down through the spaces of detail to a component space.

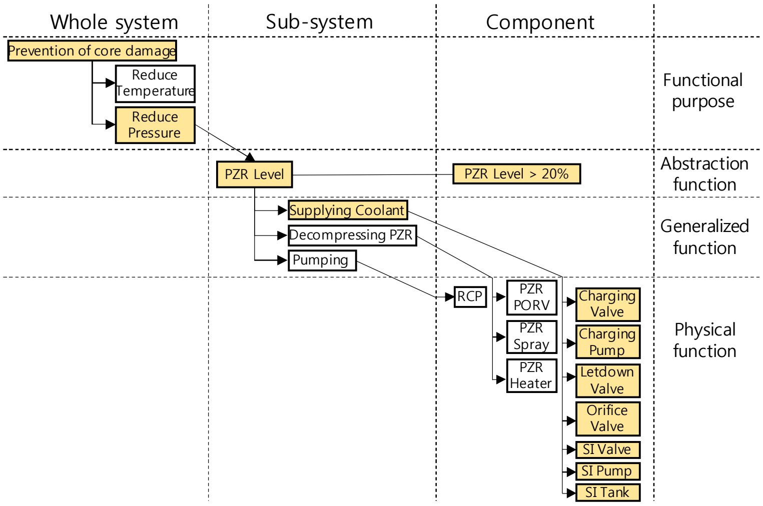


Fig. 40. An example of ADS to reduce the pressure

Fig. 40 illustrates an example of ADS for controlling the pressure of the reactor and cooling system. The functional purpose is considered as the objective of the systems and components. The functional purpose was defined as reducing pressure and temperature to prevent core damage. At the lowest level of physical function, target systems and components to be controlled are identified.

Table 13. Required physical parameters and its success criteria in abstraction function level

Physical Parameter	**Success Criteria**
PZR pressure	Pressure < 29.5kg/cm^2
	Pressure within P-T curve boundary
PZR level	20% < Level < 76%
RCS average temperature	170 C < Average temperature
	Temperature within P-T curve boundary
	55 °C/hour < Cooling rate
S/G Pressure	Pressure < 88.2kg/cm^2
S/G level	6% < Narrow level < 50%

The abstraction function represents the basic principles such as flow, mass, temperature, and level. These principles should be fully considered as the means to achieve the ends specified in the functional purpose level. Table 13 shows physical parameters and its success criteria condition based on the FRP. For instance, the pressure of the pressurizer (PZR) has a success criterion of the RCS pressure control function, i.e., the PZR pressure should be below 29.5kg/cm2, which is the shutdown operation entry condition, and stay within the pressure-temperature curve (P-T Curve) boundary as shown in Fig. 41.

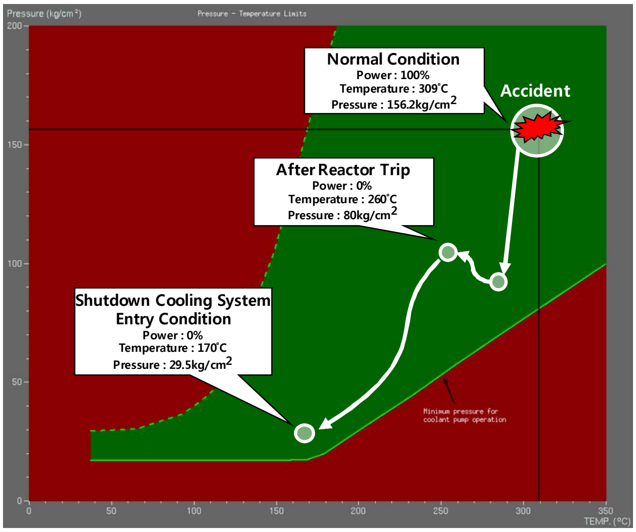


Fig. 41. P-T curve boundary and trajectory of the change of the pressure and temperature

The generalized function represents operation functions that can directly or indirectly affect the basic principle defined in the abstraction function. This function can be defined as the systematic process in relation to physical parameters. For example, PZR level is affected by decompressing PZR, and pumping and suppling the coolant. These system processes are related to the purpose of the system in the safety functions.

The physical functions are defined as the components that can achieve each systematic process, i.e., the generalized function such as pumping and suppling the coolant. Table 14 shows the components required to supply coolant to PZR. These components affect in finally satisfying the PZR level.

Table 14. Components required to supply coolant to PZR

Component
SI valve, SI pump, Charging valve, Letdown valve, Orifice valve

This study classified these components into continuous control and discrete control according to the control type. As shown in Table 15, the components required to reduce pressure and temperature are divided into two control types. The continuous controls adjust component states to satisfy specified target values of given parameters, and the rules that govern the necessary adjustments cannot be described with simple logic, i.e., cool the temperature within P-T curve adjusting the position of the steam dump valve. In contrast, a discrete control involves the direct setting of a target state, i.e., if the pressure is below 97kg/cm2, the RCP is switched off.

Table 15. Control type of components

Control type	Component
Continuous control	PZR spray valve, SI pump, SI valve, aux feedwater valve, steam dump valve
Discrete control	PZR heater, charging valve, letdown valve, orifice valve, aux feedwater pump, main feedwater pump, reactor coolant pump

B. Development of Emergency Operation Algorithm

This study developed an autonomous operation agent for emergency situations. This algorithm employs a rule-based system and Soft Actor-Critic (SAC), a kind of DRL. Fig. 42 illustrates the structure of the proposed algorithm, which consists of 1) discrete control module and 2) continuous control module.

The discrete control module controls components (discrete control type) described in Table 15., i.e., the PZR heater, charging valve, letdown valve, orifice valve, aux feedwater pump, main feedwater pump, and reactor coolant pump. In addition, the continuous control module adjusts components (continuous control type) such as the PZR spray valve, aux feedwater valve, and steam dump valve. In particular, this module focuses on operational tasks where the procedure does not provide the target status of components but the goal value of the parameter that should be achieved by the component, e.g., PZR pressure controlled by the spray. Although the continuous control includes mostly control valves, some components that have discrete states may be involved in this control. For instance, the SI pump and valves with only discrete states are categorized into the continuous control because those components are used to achieve the goal of PZR level parameter.

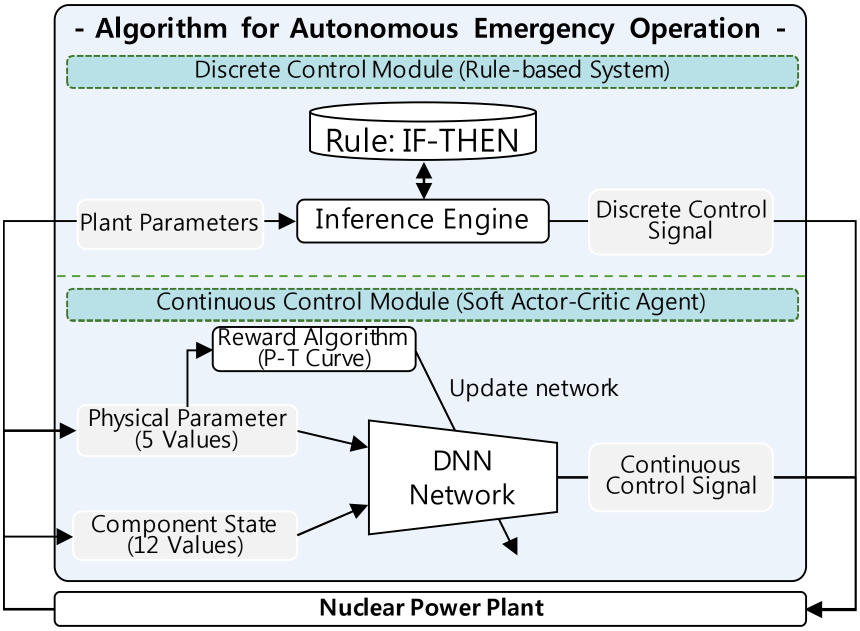
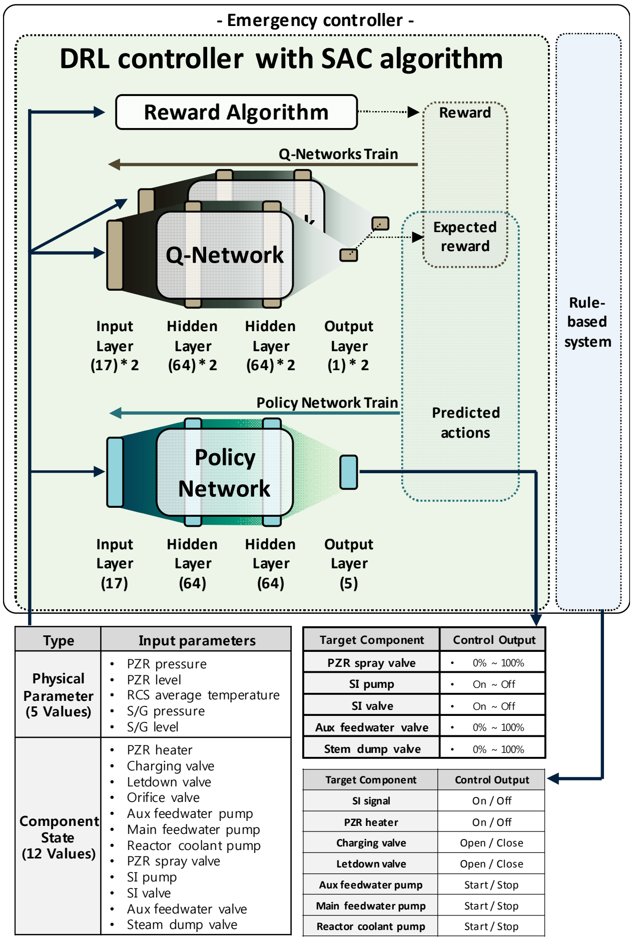

- Algorithm for Autonomous Emergency Operation -
Discrete Control Module (Rule-based System)
Rule: IF-THEN
Plant Parameters
Inference Engine
Discrete Control Signal
Continuous Control Module (Soft Actor-Critic Agent)
Reward Algorithm (P-T Curve)
Update network
Physical Parameter (5 Values)
DNN Network
Continuous Control Signal
Component State (12 Values)
Nuclear Power Plant

Fig. 42. Overview of the algorithm to reduce the primary pressure and temperature during emergency operation

Appropriate methods were selected by considering the characteristics of each control type in NPPs. A rule-based system was adopted to implement the discrete control because the specific rules can be developed from the operating procedures. On the other hand, reinforcement learning was applied to implement continuous control because it is difficult to define specific rules, i.e., how much the valve should be opened or closed. Reinforcement learning is similar to how actual operators learn and gain experiences in real operations or training for continuous control.

For the continuous control, a SAC-based algorithm and a DNN were used. Fig. 43 shows the structure of the SAC agent. As a training algorithm, the SAC was applied. The SAC agent can find the policy to explore more widely while giving up on clearly unpromising avenues. The policy can capture multiple operational paths of near-optimal behavior. The Q-values can optimize its behavior selected from the policy by considering the actual and expected rewards [10]. DNN was used to capture an action that can achieve the operational goal.



Type	Input parameters
Physical Parameter (5 Values)	• PZR pressure • PZR level • RCS average temperature • S/G pressure • S/G level
Component State (12 Values)	• PZR heater • Charging valve • Letdown valve • Orifice valve • Aux feedwater pump • Main feedwater pump • Reactor coolant pump • PZR spray valve • SI pump • SI valve • Aux feedwater valve • Steam dump valve

Target Component	Control Output
PZR spray valve	• 0% ~ 100%
SI pump	• On ~ Off
SI valve	• On ~ Off
Aux feedwater valve	• 0% ~ 100%
Stem dump valve	• 0% ~ 100%

Target Component	Control Output
SI signal	On / Off
PZR heater	On / Off
Charging valve	Open / Close
Letdown valve	Open / Close
Aux feedwater pump	Start / Stop
Main feedwater pump	Start / Stop
Reactor coolant pump	Start / Stop

Fig. 43. Structure of SAC agent

In DRLs, the reward is an essential element that updates the weights of the SAC agent; learning by the agent involves updating the weights of the network to maximize the accumulative reward. This study suggests a reward algorithm to reduce the pressure and temperature of the reactor and cooling system down to the shutdown cooling system entry condition. The reward was developed by the success criteria resulting from the analysis of the operating procedure. The reward was calculated as shown in Equations. 14 to 17. The SAC agent, who interacts with the simulator every second, gets a total reward calculated by the equations below based on the collected power plant's physical parameters. The range of expected total reward per second is (-inf ~ 0). If the reward is close to zero, it means that the agent satisfies the success criteria

$$T_{cooling} = T_{stable} - 55 \times (t - t_{trip})/3600 \tag{14}$$

$$T_{dist} = | T_t - T_{cooling} | \tag{15}$$

$$P_{dist} = | P_t - P_{cooling} | \tag{16}$$

(17)

$T_{cooling}$: Calculated Cooling Temperature
T_{stable} : Stable Temperature After Reactor Trip (260℃)
T_t : Temperature (time = t [sec])
$P_{cooling}$: Pressure of Shutdown Cooling Entry Condition
P_t : Pressure (time = t [sec])
t : Current Time [sec]
t_{trip}: Reactor Trip Time [sec]
T_{dist} : Distance between T_t and $T_{cooling}$
P_{dist} : Distance between P_t and $P_{cooling}$

The SAC agent interacts with simulators until the temperature or pressure moves outside the P-T curve boundary (operation failure) or the agent approaches the shutdown cooling entry condition (operation success). If the interaction is complete, the simulator's condition returns to initial operating conditions. This process is defined as one episode. Therefore, the SAC agent is trained through numerous episodes until the cumulative rewards of episodes converge.

C. Training and Stability

To complete the emergency operation, the SAC agent was trained for more than 800 episodes. The SAC agent training is stopped when the average reward becomes saturated stably. Fig. 44 shows the trend of the rewards that the SAC agent obtained. In one episode, the theoretically maximum cumulative reward during the entire emergency operation was 0 (the green dashed line in Fig. 44). For a cumulative reward in one episode to be zero, the SAC agent should get zero as a reward every second. However, since the pressure cannot be the same as the pressure of shutdown cooling entry condition at the beginning of the operation, the maximum cumulative reward should be selected through experimental observation. The practicably feasible maximum reward for the emergency operation success was observed to be over -65.

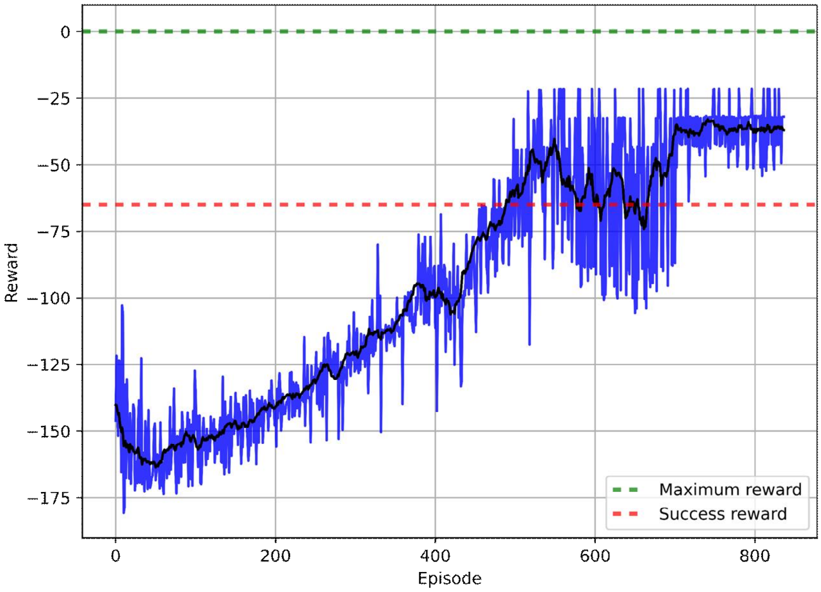


Fig. 44. Reward obtained by the SAC agent

1. Experiment result

After the proposed algorithm was trained successfully, an experiment was conducted to demonstrate that the proposed algorithm can autonomously cool down the reactor in the LOCA scenario and satisfy the operation constraints, i.e., within the P-T curve boundary with the cooling rate (55 °C/hour). As shown in Fig. 45, the proposed algorithm can reduce the pressure and temperature within operational criteria down to the entry condition of shutdown cooling.

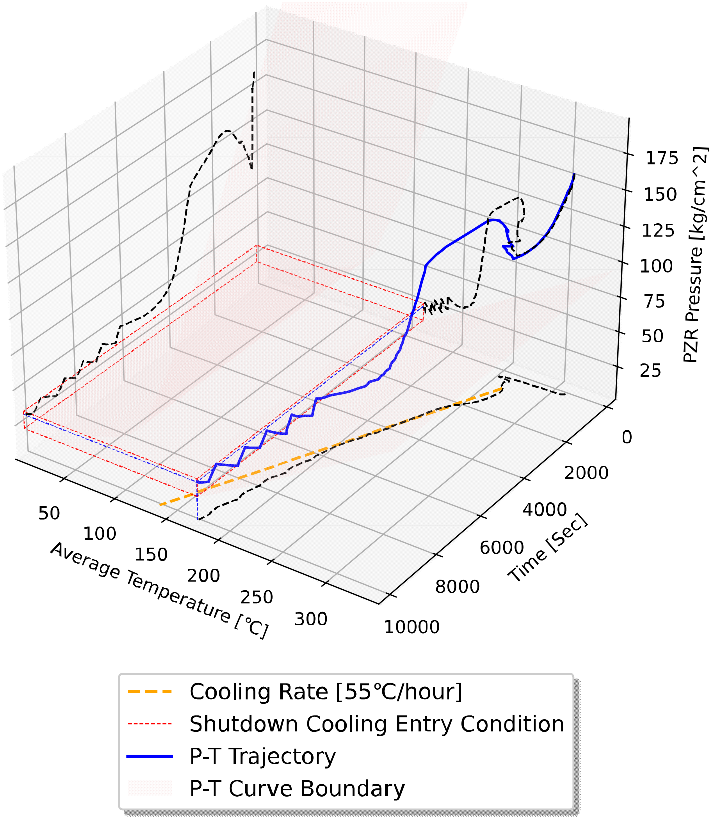


Fig. 45. Simulation results for autonomous emergency operation

V.Discussion

A. Bubble Creation Operation

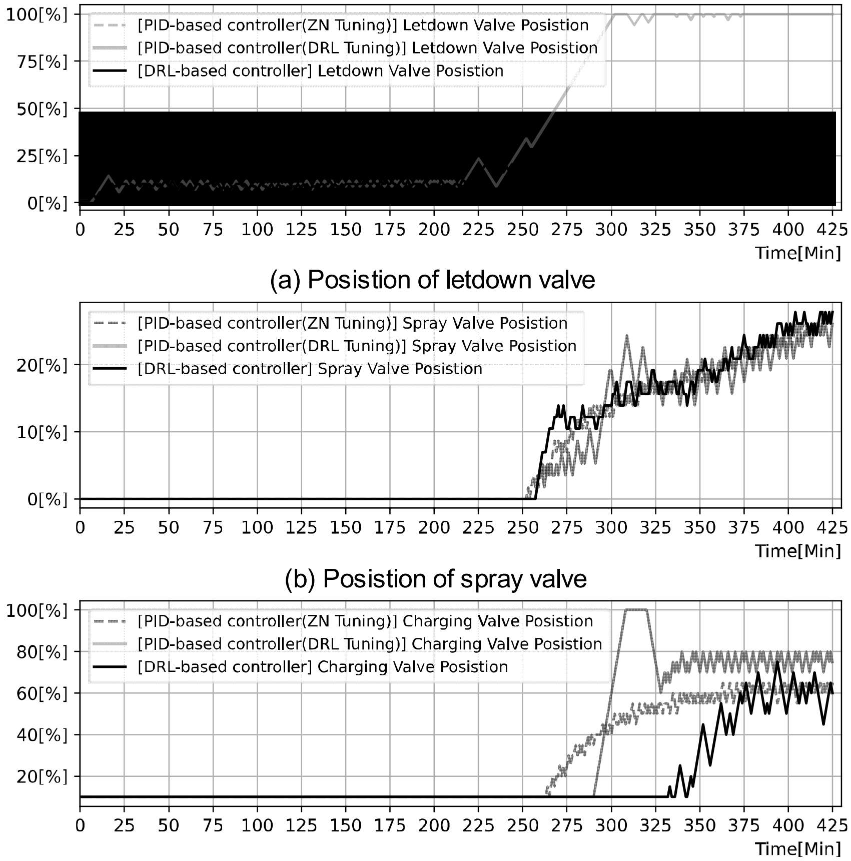
This section discusses some interesting findings from this comparison study. The DRL-tuned PID controller exhibited best performances in terms of error and time.

Table 16 compares the DRL- and PID-based controllers in terms of the average deviation error from the target value of the parameters and the time taken to reach the target value. For the pressurizer pressure and level, the DRL-tuned PID controller generally exhibited the smallest error and fastest reaching time than both the ZN-tuned PID controller and the DRL-based controller.

Table 16. Comparison result of operational performances

Performance		**PID-based controller**	**DRL-based controller**
Pressurizer Pressure	Average deviation error from 27 kg/cm^2	±0.3248 kg/cm^2 (ZN) ±0.1805 kg/cm^2 (DRL)	±0.2816 kg/cm^2
	Reaching time to 27 kg/cm^2	32 minutes (ZN) 10 minutes (DRL)	10 minutes
Pressurizer Level	Average deviation error from 50%	±9.56% (ZN) ±6.55% (DRL)	±8.79%
	Reaching time to 50%	+ 144 minutes (ZN) + 38 minutes (DRL)	+ 93 minutes

Although PID-based controllers are dedicated to one component, DRL-based controllers manage the parameters and control multiple components simultaneously. To control the pressurizer pressure to the desired value, three PID-based controllers were designed for three components: the charging valve, letdown valve, and spray. The controllers opened these values and the spray simultaneously to reduce pressure, as shown in Fig 46. On the other hand, two DRL-based controllers were developed for the control of pressure, not the control of components. Thus, the DRL-based controllers manipulate the three components in an interactive manner. For instance, as shown in Fig 46, the DRL-based controllers closed the letdown value at approximately 260 min and instead maintained the charging valve closed, while the PID-based controllers consistently opened these valves at the same time.



(a) Posistion of letdown valve

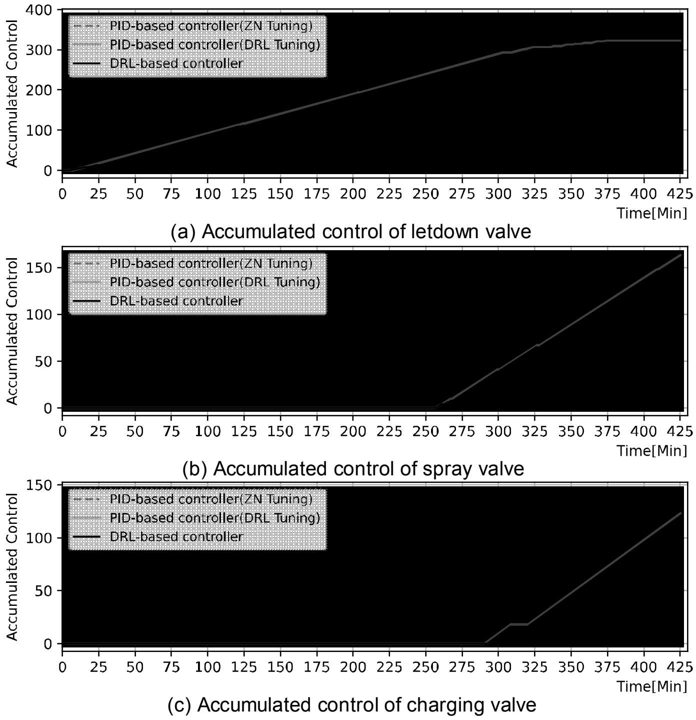
(b) Posistion of spray valve

(c) Posistion of charging valve

Fig. 46. Positions of charging, spray, and letdown valve

PID-based controllers manipulate components more frequently than DRL-based controllers. Fig 47 shows a comparison of the number of manipulations for the three components. As shown in the figure, PID-based controllers control the components more frequently than DRL-based controllers. This may be related to the second finding described above. DRL-based controllers work interactively and can satisfy the operational goal with fewer manipulations.

Less frequent manipulation is desirable in NPPs. First, frequent manipulation is likely to lead to component failures. From the perspective of probabilistic safety assessment, once a component starts to work (e.g., open/close or start/stop)., the probability of failing to work increases. Second, frequent manipulation accelerates the aging or fatigue of components. Thus, the replacement period is shortened because of aging.



(a) Accumulated control of letdown valve

(b) Accumulated control of spray valve

(c) Accumulated control of charging valve

Fig. 47. Total manipulation of letdown/spray/charging valve during cold shutdown operation

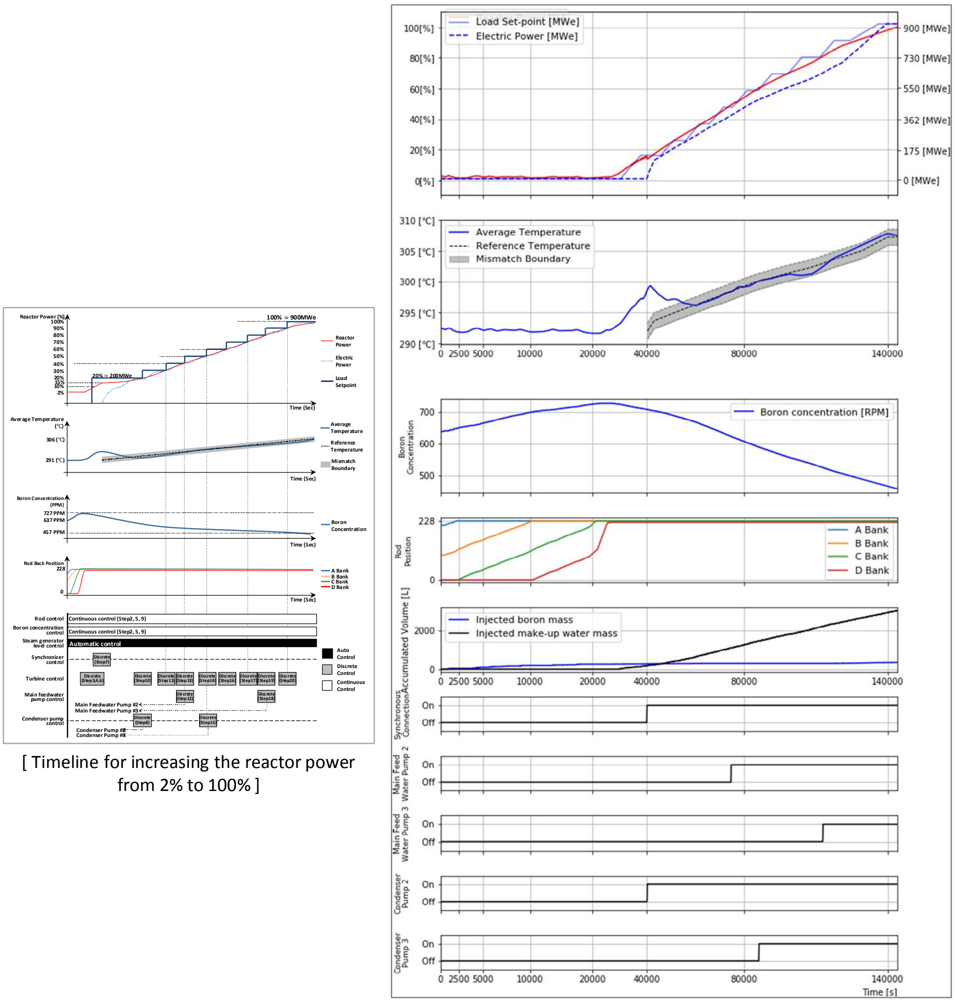
B. Power Increase Operation

The experimental results demonstrated that the proposed algorithm successfully controlled the components to increase the reactor power and generate electrical power at the intended rate of power increase. The performance of this algorithm was also compared with that of the established operation strategy. According to Fig. 48, the proposed algorithm had a pattern of operation that was nearly identical to that of the established operational strategy. Therefore, it is concluded that the proposed algorithm, which combines a rule-based system and reinforcement learning, can successfully control the complicated power-increase operation.

In this algorithm, the discrete control module operated the synchronizer controller, turbine controller, main feedwater pump, and condenser pump according to the operational steps that are clearly stated in the GOPs.

The continuous control module adjusted the valves to manage the boron concentration and manipulated the rod controller. The continuous control module can provide experiential control of these inputs, thereby gradually affecting the power increase, based on the parameter trends, the predetermined rate of power increase, and the current operational boundaries. In addition, the results demonstrate that the continuous control module effectively managed the boron concentration such that the difference between the average temperature and the reference temperature was maintained within ± 1 ℃. Since this rule is not mandated in the GOPs, the control module allowed average temperatures that were outside the mismatch boundary.

However, based on interviews with senior operators who work at the reference plant, this restriction can be satisfied after the reactor power reaches 30%; in the earlier stages of the power-increase operation, the start-up of large components results in system disturbances that complicate temperature control. Therefore, these results demonstrate that the A3C agent in the continuous control module can effectively conduct experience-based control after training with the simulator and the discrete control module can control components according to rules that are based on the operating procedures.



[Timeline for increasing the reactor power from 2% to 100%]

[Experimental result of the power increase at 3%/h]

Fig. 48. Comparison between the existing operational strategy and the simulation results

Several aspects should be further considered regarding the practical application of this algorithm:

1) Since the power-increase operation is only a small part of the overall plant operation, to cover the entire plant operation, the proposed reward algorithm should be changed according to the operation objectives, strategies, operational methods, and required procedural steps for each operating range. Moreover, the AI agent should be capable of selecting and controlling an operating strategy based on the context.

2) To further improve the safety of NPPs, an AI agent requires additional functions (e.g., fault detection, diagnosis, forecasting the status of the plant, identifying the possible control options, and recommending the best option) to address emergency, abnormal, and normal situations.

3) The signal noise in a plant should be an additional consideration; signals in NPPs contain noise, while the simulator does not. Therefore, a technique that can mitigate the signal noise, e.g., signal validation or noise tolerance, must be developed.

4) Another issue is the differences in behavior between the simulator model and actual power plants, which mandates a thorough validation of the practical application.

C. Emergency Operation

Licensing is one of the unsolved issues for the DRL-based controllers. The application of NPPs requires proven technologies. In particular, for safety-critical systems, controllers need to be approved by regulations. PID-based controllers have been acknowledged as a proven technology, because they have been used in NPPs for decades. However, it is common knowledge that AI technologies have not been sufficiently proven. Therefore, solving the licensing issue is the largest problem for applying DRL-based controllers to NPPs.

Even though the licensing issue is beyond the scope of this study, it is worth investigating some approaches to proving AI. The first is the use of an explainable AI called XAI. XAI can show how the AI produces the result and makes the AI closer to a whitebox. The second is the application of the software development process. The software used in the safety-critical system of NPPs should follow a very strict development process recommended by various standards, such as IEEE Standards 1012 [79] and 7-4.3.2 [80]. Because AI-based controllers can also be regarded as software, they are considered to apply the software development process to them.

VI. Conclusion

This study developed a intelligent controller for an autonomous operation in NPPs during start-up and emergency. The controller was focused on conducting high-level operations that are similarly performed to the current operation strategy. To manipulate components similarly to operators, controllers currently aimed to automate manual controls.

First, this study compares the performance of DRL- and PID-based controllers in the cold shutdown operation of NPPs. This study conducted a task analysis for the bubble creation operation based on the operating procedures. Subsequently, PID- and DRL-based controllers were developed to satisfy the operational goal of the operation. The PID-based controllers were tuned by using the Ziegler-Nichols and DRD-based tuning method. This study compared the performances of the controllers. In general, the DRL-tuned PID controller exhibited the smallest error and fastest reaching time than both the ZN-tuned PID controller and the DRL-based controller. Finally, we presented some interesting findings.

Second, this study proposed an algorithm for the power-increase operation. The power increase algorithm was also designed through an analysis of the current operational strategy, which considered the operation staffing and operating procedures. To train the continuous control, the proposed algorithm used an A3C agent and an LSTM network and applied a rule-based system for the discrete control components. The CNS was used to determine whether the proposed algorithm could effectively and autonomously control the power-increase operation at a 3%/h rate of power increase. Based on the simulation results, the power increase algorithm was proven capable of identifying an acceptable operation path for increasing the reactor power from 2% to 100% at a specified rate of power increase.

Third, this study proposed an algorithm for an autonomous emergency operation that uses AI techniques. The emergency operation algorithm was developed through a domain analysis based on the FRPs using ADS. The proposed algorithm used a SAC agent and a DNN network for the continuous control and applied a rule-based system for the discrete control. A compact nuclear simulator was used to train and test the algorithm. Based on the simulation results, this algorithm reached the shutdown operation entry condition, according to the cooling rate (55 °C/hour).

These three studies was shown that the validation results showed that the autonomous operation algorithm can mange the NPPs according to given operational goals. The suggested approach seems to be applicable to other operational modes in NPPs, if the reward algorithm is adjusted according to the operation objectives, strategies, methods, and required procedure steps for each operating range.

Future studies may suggest developing an agent that can select and control a contextual operating strategy, either in the entire operation range or in part. Future studies may also consider emergency as well as abnormal situations during power-increasing operation. More so, to realize a fully automated NPP, an autonomous control system should be capable of: automatic operation of the NPP, fault detection, diagnosis (identifying the causes of component failures or incidents), simulation, forecasting the status of the plant, identifying the possible control options, and recommending the best option for optimizing the plant performance. This autonomous control is expected to be a key technology in small modular reactors that are under development.

REFERENCES

1. Wood, R.T., et al. Autonomous control capabilities for space reactor power systems. in AIP Conference Proceedings. 2004. American Institute of Physics.
2. Lu, C., et al. Nuclear power plants with artificial intelligence in industry 4.0 era: Top-level design and current applications-A systemic review. IEEE Access, 2020. 8: pp. 194315-194332.
3. Choi, S.S., et al. Development strategies of an intelligent human-machine interface for next generation nuclear power plants. IEEE Transactions on Nuclear Science, 1996. 43(3): pp. 2096-2114.
4. Kim, J., et al. Conceptual design of autonomous emergency operation system for nuclear power plants and its prototype. Nuclear Engineering and Technology, 2020. 52(2): pp. 308-322.
5. Lee, D., and Kim, J. Autonomous algorithm for start-up operation of nuclear power plants by using LSTM. in International Conference on Applied Human Factors and Ergonomics. 2018. Orlando, Florida, USA: Springer.
6. Li, Y., Deep reinforcement learning: An overview. arXiv preprint arXiv:1701.07274, 2017.
7. Lee, S.J., and Seong, P.H. Development of automated operating procedure system using fuzzy colored petri nets for nuclear power plants. Annals of nuclear energy, 2004. 31(8): pp. 849-869.
8. Kima, Y., and Park, J. Envisioning human-automation interactions for responding emergency situations of NPPs: a viewpoint from human-computer interaction, in Transactions of the Korean Nuclear Society Autumn Meeting. 2018: Yeosu.
9. Kim, A.R., et al. Study on the identification of main drivers affecting the performance of human operators during low power and shutdown operation. Annals of Nuclear Energy, 2016. 92: pp. 447-455.

10. Haarnoja, T., et al. Soft actor-critic algorithms and applications. arXiv preprint arXiv:1812.05905, 2018.
11. Viitala, A., et al. Learning to drive small scale cars from scratch. arXiv preprint arXiv:2008.00715, 2020.
12. Bhalla, S., et al. M. Deep multi agent reinforcement learning for autonomous driving. in Canadian Conference on Artificial Intelligence. 2020. Springer.
13. Yu, C., et al. Distributed multiagent coordinated learning for autonomous driving in highways based on dynamic coordination graphs. IEEE Transactions on Intelligent Transportation Systems, 2019. 21(2): pp. 735-748.
14. Ziegler, J.G. and Nichols, N.B. Optimum settings for automatic controllers. trans. ASME, 1942. 64(11).
15. Cohen, G., Theoretical consideration of retarded control. Trans. Asme, 1953. 75: pp. 827-834.
16. Astrom, K.J. and Hagglund, T. Automatic tuning of simple regulators with specifications on phase and amplitude margins. Automatica, 1984. 20(5): pp. 645-651.
17. Malwatkar, G., et al. Tuning PID controllers for higher-order oscillatory systems with improved performance. ISA transactions, 2009. 48(3): pp. 347-353.
18. Izci, D., et al. HHO algorithm based PID controller design for aircraft pitch angle control system. in 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA). 2020. IEEE.
19. Eker, E., et al., A new fusion of ASO with SA algorithm and its applications to MLP training and DC motor speed control. Arabian Journal for Science and Engineering, 2021. 46(4): pp. 3889-3911.
20. Solihin, M.I., et al., Tuning of PID controller using particle swarm optimization (PSO). in Proceeding of the international conference on advanced science, engineering and information technology. 2011.
21. Carlucho, I., et al., An adaptive deep reinforcement learning approach for

MIMO PID control of mobile robots. ISA transactions, 2020. 102: pp. 280-294.

22. Kim, J.T., et al., Development of advanced I&C in nuclear power plants: ADIOS and ASICS. Nuclear Engineering and Design, 2001. 207(1): pp. 105-119.
23. Wei, T., et al., Deep reinforcement learning for building HVAC control. in Proceedings of the 54th Annual Design Automation Conference. 2017.
24. Goodfellow, I., et al., Deep learning. 2016: MIT press.
25. Schmidhuber, J., Deep learning in neural networks: An overview. Neural networks, 2015. 61: pp. 85-117.
26. Vecerik, M., et al., A practical approach to insertion with variable socket position using deep reinforcement learning. in International Conference on Robotics and Automation (ICRA). 2019. IEEE.
27. Kazmi, H., et al., Gigawatt-hour scale savings on a budget of zero: Deep reinforcement learning based optimal control of hot water systems. Energy, 2018. 144: pp. 159-168.
28. Kang, C., et al., An automatic algorithm of identifying vulnerable spots of internet data center power systems based on reinforcement learning. International Journal of Electrical Power & Energy Systems, 2020. 121: pp. 106145.
29. Rocchetta, R., et al., A reinforcement learning framework for optimal operation and maintenance of power grids. Applied energy, 2019. 241: pp. 291-301.
30. Du, G., et al., Deep reinforcement learning based energy management for a hybrid electric vehicle. Energy, 2020: pp. 117591.
31. Zhou, S., et al., Combined heat and power system intelligent economic dispatch: A deep reinforcement learning approach. International Journal of Electrical Power & Energy Systems, 2020. 120: pp. 106016.
32. Palanisamy, P., Multi-agent connected autonomous driving using deep

reinforcement learning. in International Joint Conference on Neural Networks (IJCNN). 2020. IEEE.

33. Kiran, B.R., et al., Deep reinforcement learning for autonomous driving: A survey. IEEE Transactions on Intelligent Transportation Systems, 2021.
34. Yang, Z., et al., Deep-reinforcement-learning-based energy management strategy for supercapacitor energy storage systems in urban rail transit. IEEE Transactions on Intelligent Transportation Systems, 2020.
35. Saenz-Aguirre, A., et al., Artificial neural network based reinforcement learning for wind turbine yaw control. Energies, 2019. 12(3): pp. 436.
36. Genders, W. and Razavi, S., Policy analysis of adaptive traffic signal control using reinforcement learning. Journal of Computing in Civil Engineering, 2020. 34(1): pp. 04019046.
37. Dong, Z., et al., Multilayer perception based reinforcement learning supervisory control of energy systems with application to a nuclear steam supply system. Applied Energy, 2020. 259: pp. 114193.
38. Park, J., et al., Providing support to operators for monitoring safety functions using reinforcement learning. Progress in Nuclear Energy, 2020. 118: pp. 103123.
39. Bennett, S., The past of PID controllers. Annual reviews in control, 2001. 25: pp. 43-53.
40. Mousakazemi, S.M.H., Computational effort comparison of genetic algorithm and particle swarm optimization algorithms for the proportional-integral-derivative controller tuning of a pressurized water nuclear reactor. Annals of Nuclear Energy, 2020. 136: pp. 107019.
41. Mousakazemi, S.M.H., Control of a PWR nuclear reactor core power using scheduled PID controller with GA, based on two-point kinetics model and adaptive disturbance rejection system. Annals of Nuclear Energy, 2019. 129: pp. 487-502.
42. Zhang, B., et al., Novel fuzzy logic based coordinated control for multi-unit

small modular reactor. Annals of Nuclear Energy, 2019. 124: pp. 211-222.

43. Upadhyaya, B.R., et al., Autonomous control of space reactor systems. 2007, Univ. of Tennessee, Knoxville, TN (United States).

44. Na, M.G. and Upadhyaya, B.R., A neuro-fuzzy controller for axial power distribution an nuclear reactors. IEEE Transactions on Nuclear Science, 1998. 45(1): pp. 59-67.

45. Rojas-Ramirez, E., et al., A stable adaptive fuzzy control scheme for tracking an optimal power profile in a research nuclear reactor. Annals of Nuclear Energy, 2013. 58: pp. 238-245.

46. Arab-Alibeik, H. and Setayeshi, S., Adaptive control of a PWR core power using neural networks. Annals of Nuclear Energy, 2005. 32(6): pp. 588-605.

47. Jiang, Q., et al., Study on switching control of PWR core power with a fuzzy multimodel. Annals of Nuclear Energy, 2020. 145: pp. 107611.

48. Zeng, W., et al., A fuzzy-PID composite controller for core power control of liquid molten salt reactor. Annals of Nuclear Energy, 2020. 139: pp. 107234.

49. Belles, R. and Muhlheim, M.D., Licensing Challenges Associated with Autonomous Control. 2018, Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States).

50. Basher, H., et al., Autonomous Control of Nuclear Power Plants. 2003: United States. Department of Energy.

51. Mnih, V., et al., Asynchronous methods for deep reinforcement learning. in International conference on machine learning. 2016.

52. Kapturowski, S., et al., Recurrent experience replay in distributed reinforcement learning. in International conference on learning representations. 2018.

53. Horgan, D., et al., Distributed prioritized experience replay. arXiv preprint arXiv:1803.00933, 2018.

54. Lee, D., et al., Algorithm for Autonomous Power-Increase Operation Using Deep Reinforcement Learning and a Rule-Based System. IEEE Access, 2020.

8: pp. 196727-196746.

55. Kim, H., et al., Development of a diagnostic algorithm for abnormal situations using long short-term memory and variational autoencoder. Annals of Nuclear Energy, 2021. 153: pp. 108077.
56. Yang, J. and Kim, J., Accident diagnosis algorithm with untrained accident identification during power-increasing operation. Reliability Engineering & System Safety, 2020. 202: pp. 107032.
57. Lee, D., et al., Autonomous operation algorithm for safety systems of nuclear power plants by using long-short term memory and function-based hierarchical framework. Annals of Nuclear Energy, 2018. 119: pp. 287-299.
58. Guo, X., et al., Deep learning for reward design to improve monte carlo tree search in atari games. arXiv preprint arXiv:1604.07095, 2016.
59. Aggarwal, M., et al., Improving search through a3c reinforcement learning based conversational agent. in International Conference on Computational Science. 2018. Springer.
60. Hochreiter, S. and Schmidhuber, J., Long short-term memory. Neural computation, 1997. 9(8): pp. 1735-1780.
61. Seker, S., et al., Elman's recurrent neural network applications to condition monitoring in nuclear power plant and rotating machinery. Engineering applications of artificial intelligence, 2003. 16(7-8): pp. 647-656.
62. Gers, F.A., et al., Learning to forget: Continual prediction with LSTM. Neural computation, 2000. 12(10): pp. 2451-2471.
63. Trinh, T.T., et al., Bird detection near wind turbines from high-resolution video using lstm networks. in World Wind Energy Conference (WWEC). 2016.
64. KAERI, Advanced Compact Nuclear Simulator Textbook. 1990: Nuclear Training Center in Korea Atomic Energy Research.
65. Pongfai, J., et al., Optimal PID controller autotuning design for MIMO nonlinear systems based on the adaptive SLP algorithm. International Journal

of Control, Automation and Systems, 2021. 19(1): pp. 392-403.

66. Willis, M., Proportional-integral-derivative control. Dept. of Chemical and Process Engineering University of Newcastle, 1999.
67. Sung, C.H. and Min, M.G., A Method of Tuning Optimization for PID Controller in Nuclear Power Plants. Transactions of the Korean Society of Pressure Vessels and Piping, 2014. 10(1): pp. 1-6.
68. Zeng, W., et al., A functional variable universe fuzzy PID controller for load following operation of PWR with the multiple model. Annals of Nuclear Energy, 2020. 140: pp. 107174.
69. Korkmaz, M., et al., Design and performance comparison of variable parameter nonlinear PID controller and genetic algorithm based PID controller. in 2012 International Symposium on Innovations in Intelligent Systems and Applications. 2012. IEEE.
70. Mutlag, A.H., et al., Optimum PID controller for airplane wing tires based on gravitational search algorithm. Bulletin of Electrical Engineering and Informatics, 2021. 10(4): pp. 1905-1913.
71. Ogata, K., Modern control engineering. 2010: Prentice hall.
72. Na, M.G., et al., A model predictive controller for nuclear reactor power. Nuclear Engineering and Technology, 2003. 35(5): pp. 399-411.
73. Yang, J. and Kim, J., An accident diagnosis algorithm using long short-term memory. Nuclear Engineering and Technology, 2018. 50(4): pp. 582-588.
74. Ng, A.Y., et al., Autonomous inverted helicopter flight via reinforcement learning, in Experimental robotics IX. 2006, Springer. pp. 363-372.
75. Guo, X., Deep learning and reward design for reinforcement learning (Doctoral dissertation). 2017.
76. Chen, T., et al., Gradient band-based adversarial training for generalized attack immunity of A3C path finding. arXiv preprint arXiv:1807.06752, 2018.
77. Garduno-Ramirez, R. and Lee, K.Y., Multiobjective optimal power plant operation through coordinate control with pressure set point scheduling. IEEE

Transactions on energy conversion, 2001. 16(2): pp. 115-122.

78. KHNP, APR1400 design description. 2014, Korea Hydro & Nuclear Power CO., LTD.

79. IEEE Standard for System, Software, and Hardware Verification and Validation. 2017, IEEE Computer Society. pp. 1-206.

80. IEEE Standard Criteria for Programmable Digital Devices in Safety Systems of Nuclear Power Generating Stations. 2016, IEEE Power and Energy Society. pp. 1-86.