Autonomous Power-Increase Operation for a Small Modular Reactor Based on Task Analysis with Proximal Policy Optimization

Hee-Jae Lee[a] and Jonghyun Kim[a]

[a]*Korea Advanced Institute of Science and Technology, 291 Daehak-ro, Yuseong-gu, Daejeon, 34141, Republic of Korea, jonghyun.kim@kaist.ac.kr*

Abstract: As small modular reactors move toward multi-module operation, the increased cognitive workload on operators-particularly power-increase operation. This study develops an autonomous algorithm to automate the power-increase operation. To achieve this, this study first performed a task analysis of the power-increase operation to identify candidate tasks for automation and to derive automation strategies. Based on the task analysis results, an algorithm was designed by combining a deep reinforcement learning-based system with a rule-based system, specifically proximal policy optimization and if-then rules, respectively. Experimental results demonstrated that the algorithm successfully achieved 100% reactor power while maintaining safety constraints, such as keeping the startup rate below 0.5 dpm. Additionally, the developed algorithm was implemented and visualized through a graphical user interface.

Keywords: Small Modular Reactor, Power-Increase Operation, Task Analysis, Proximal Policy Optimization, Operator Support

1. INTRODUCTION

Interest in small modular reactors (SMRs) has grown steadily because their compact, modular design offers enhanced passive safety and the flexibility to add capacity incrementally [1,2]. A defining feature of many SMR concepts is multi-module deployment, in which one control room oversees several modules simultaneously. While this arrangement improves economics, it also concentrates supervisory responsibility on fewer operators, raising concerns about how much workload a single operator can reasonably handle [3]. Lessons from other safety-critical domains, such as aviation, show that well-designed automation can relieve operator burden in tasks that require frequent and prolonged manual intervention [4-6].

Among the various plant operations, the power-increase operation is particularly demanding. Bringing a reactor from zero to full power involves coordinating control rods, boron concentration, and the turbine over an extended period. Throughout this period, the operator needs to monitor many parameters continuously and to make timely adjustments. To support the operator through this process, operating procedures lay out the operation as a sequence of discrete, step-by-step actions. These procedures cover the discrete actions well, but they cannot fully express the continuous, judgment-laden subtasks (e.g. boron dilution).

Deep reinforcement learning (DRL) offers a natural way to fill this gap. Through trial-and-error interaction with a simulator, an agent can acquire the adaptive decision-making ability that is traditionally dependent on operators. Accordingly, DRL has emerged as a promising route to operational autonomy for nuclear power plants [7]. Recent applications span safety-system actuation [8], power-increase operation in conventional reactors [9], reactor-core startup of SMRs [10], multi-objective plant operation [11], cold-shutdown control [12], and reactor control of boiling-water reactors [13]. The present work extends this line by task analysis and a robust-AI PPO formulation to the power-increase operation of an SMR. It

differs from [9], which targets a conventional reactor with a raw-signal PPO agent, and from [10], which addresses reactor-core startup. The present paper covers the 0–100 % power-up of an SMR and uses a trend-image (robust-AI) representation for the continuous subtask.

This paper presents an algorithm that carries out the power-increase operation autonomously for an SMR, utilizing an integral pressurized water reactor (iPWR) simulator [14]. This work first performs a task analysis of the operation to expose which subtasks can be automated and how. The resulting algorithm combines two complementary mechanisms: if-then rules for the discrete subtasks and a proximal-policy-optimization (PPO) agent for the continuous boron-dilution subtask. The remainder of the paper describes the task analysis (Section 2), the algorithm design (Section 3), and the experimental demonstration centered on a comparison of three boron-dilution controllers (Section 4), before concluding in Section 5. A complete treatment of the study is available in the authors' journal article [15].

2. TASK ANALYSIS OF THE POWER-INCREASE OPERATION

The iPWR simulator used in this study is a training simulator distributed by the IAEA. It can simulate reactor, control-rod, boron, and turbine subsystems that respond to operator actions in real time. Fig. 1 shows its operating interface.

The power-increase operation raises the reactor from 0 % to 100 % power. To understand it systematically, the procedure was examined through task analysis, decomposing the overall goal into ordered subtasks [16-18]. For every subtask, several attributes were decomposed: 1) the Action Verb, 2) the Monitoring Parameter, 3) the Expected Response, 4) the GUI Sheet, 5) the Manipulation flag, 6) the Control Means, and 7) the Task Type (discrete or continuous). In particular, the Task Type separates the subtasks into two: discrete and continuous. Discrete subtasks are triggered by specific conditions and map naturally onto explicit rules, such as withdrawing control-rod banks until the steps reaches 49 and pushing run-up button when synchronization light is on. The continuous subtask (e.g., boron dilution) is defined as one that requires adaptive decision-making rather than fixed trigger conditions.

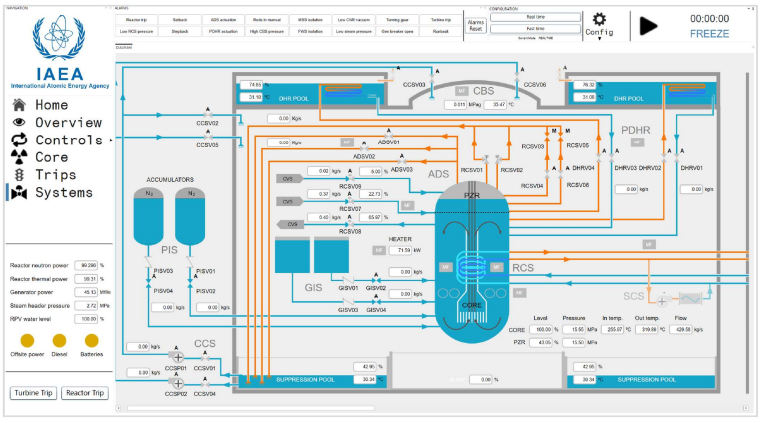


Fig. 1. Operating interface of the IAEA iPWR simulator.

Table 1 summarizes representative subtasks and the attributes derived from the analysis. This classification directly motivates the two-part algorithm of Section 3: discrete subtasks are delegated to a rule-based component, whereas the continuous boron-dilution subtask is handed to a learning-based component. Fig. 2 presents the hierarchical structure obtained from the decomposition.

Table 1: Representative subtasks of the power-increase operation and their attributes.

Subtask	Monitoring parameter	Control means	Task type
Control-rod withdrawal	Reactor power, SUR	Rod bank position	Discrete
Turbine run-up	Turbine speed	Turbine controller	Discrete
Generator synchronization	Grid/phase status	Synchronizer	Discrete
Boron dilution	Reactor power, boron conc., reactivity	Charging/dilution flow	Continuous
Turbine load increase	Turbine load demand	Load setpoint	Discrete

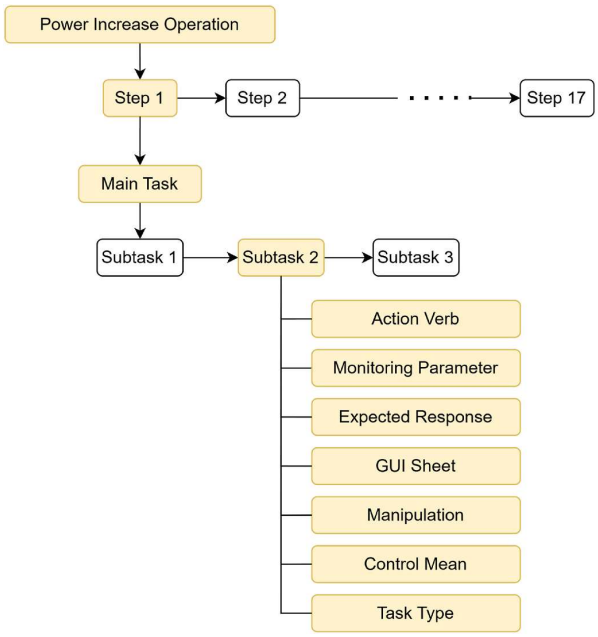


Fig. 2. Hierarchical structure used to decompose the power-increase operation (reproduced from [15]).

3. AUTONOMOUS ALGORITHM DESIGN

Guided by the task analysis, the autonomous algorithm is organized around a task manager that supervises execution and dispatches each subtask to the appropriate engine, as shown in Fig. 3. A task checker continuously identifies the current subtask; when that subtask is boron dilution, control passes to the deep-reinforcement-learning (DRL) engine, and otherwise to the rule-based engine. Both engines read plant parameters from the simulator and return control signals, so the operation proceeds without manual input while remaining transparent to a supervising operator.

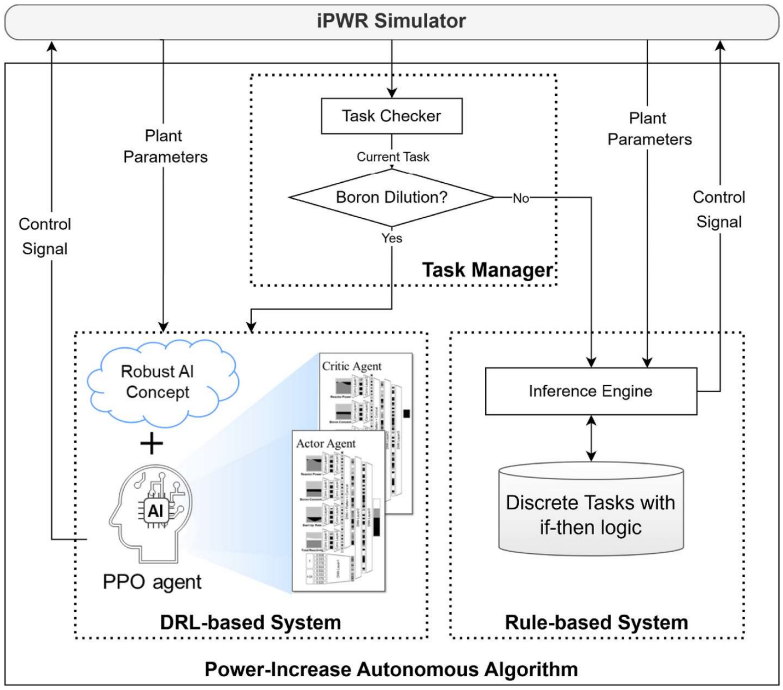


Fig. 3. Overview of the power-increase autonomous algorithm (reproduced from [15]).

3.1. Rule-Based Engine

The discrete subtasks identified in Section 2 are encoded as if-then rules. Each rule links a monitored condition to a control action, mirroring the procedural steps an operator would follow. For example, the engine advances a control-rod bank once the power and startup-rate conditions are satisfied, or it initiates turbine run-up and synchronization at the prescribed stage. These subtasks have clear triggering conditions. Explicit rules can therefore execute them reliably while remaining easy to inspect and verify.

3.2. Reinforcement-Learning Engine

Boron dilution is the continuous subtask and is delegated to a PPO agent [19,20]. PPO is an actor-critic method in which the actor proposes control actions and the critic estimates their long-term value; its clipped objective keeps each policy update small, which stabilizes training. The agent here adopts a robust-AI formulation, meaning that it consumes short-window trend images rather than instantaneous scalar readings so that point-wise input corruption has less effect on its decisions. Concretely, the recent trends of four key signals—reactor power, boron concentration, startup rate, and total reactivity—are turned into color-coded trend images, which a convolutional network with pooling layers compresses into features. PPO was chosen for its stable on-policy updates and its effectiveness in continuous-control tasks with shaped rewards. The boron-dilution subtask is continuous in effect, but it is actuated through a discrete set of dilution-rate primitives (e.g., −5 ppm, −2 ppm, −1 ppm, and hold). The actor therefore ends in a softmax layer that yields a probability distribution over this finite action set, while the critic produces a scalar value estimate. Fig. 4 illustrates the resulting architecture.

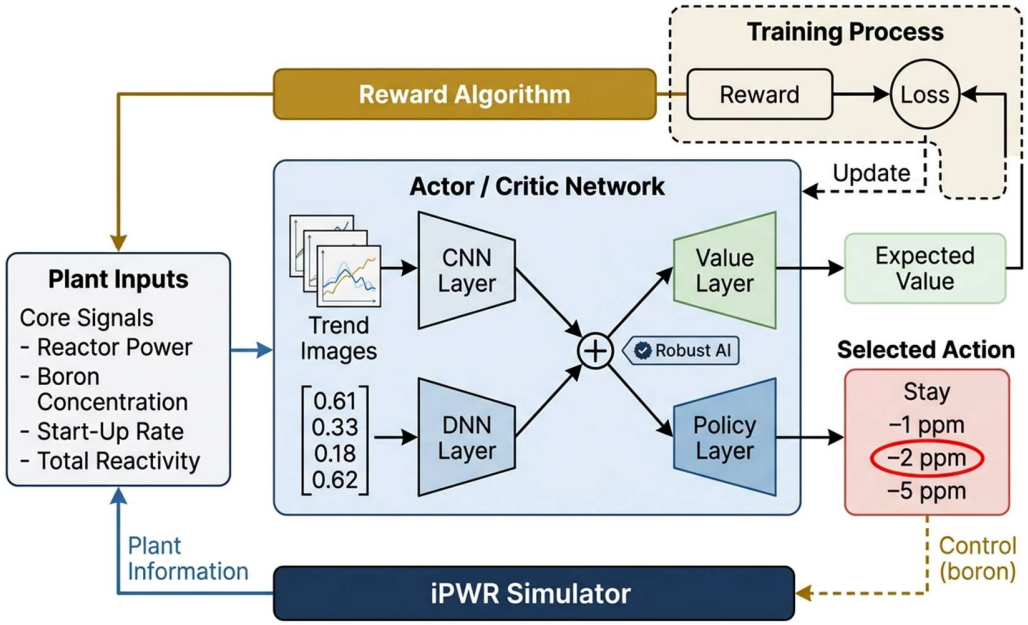


Fig. 4. Structure of the PPO agent incorporating the robust-AI concept for boron dilution (redrawn based on [15]).

A reward function shapes the agent toward smooth, safe dilution. It rewards progress toward the target power while penalizing actions that would push reactivity or the startup rate beyond their limits. A reactivity threshold near 100 pcm gates the dilution actions. Through this design the agent learns to dilute boron gradually rather than abruptly, which helps ensure stable convergence near the target power.

4. EXPERIMENTS AND RESULTS

The central result of this paper is a controlled comparison of three controllers for the boron-dilution subtask: a traditional if-then logic, a plain DNN-based PPO agent, and the proposed PPO agent with the robust-AI formulation. To set the comparison in context, this section first reports the training of the reinforcement-learning agent and the demonstration of the complete algorithm on a full power-increase operation from 0 % to 100 %; the comparison itself is then developed in Section 4.3.

4.1. Experimental Setup and Training

Training and testing were carried out on a workstation equipped with an NVIDIA GeForce RTX 3050 GPU for the neural-network computations and an AMD Ryzen 5 3600 CPU whose twelve threads drove a parallel simulation environment. As shown in Fig. 5, this environment ran several simulator instances concurrently and exchanged data with the agent over TCP/IP, so that state-action-reward samples could be collected far faster than with a single simulator. Each sample consisted of the agent's observed state, the action it chose, and the reward it received. These tuples were accumulated as training data, and the actor and critic networks then processed the batched states in parallel.

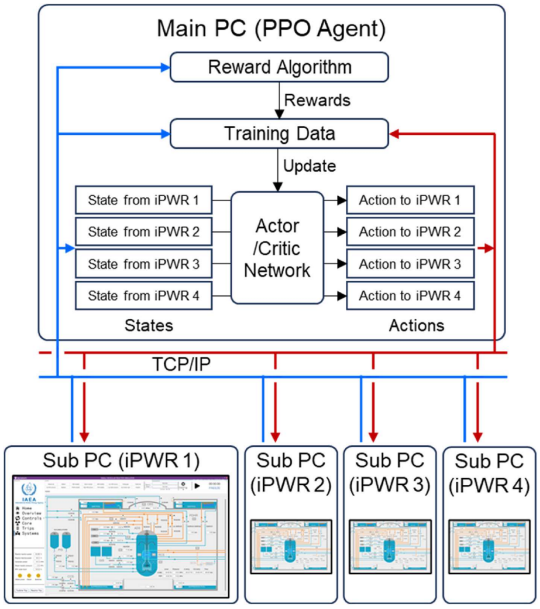


Fig. 5. Parallel training environment used for the PPO agent (reproduced from [15]).

Fig. 6 shows the cumulative reward earned by the agent in each episode over about 1,400 training episodes. The reward stayed low for the first few hundred episodes while the agent explored. It then rose sharply, and once it settled around the 1,000 mark the boron-dilution policy had effectively converged. As the policy matured, the agent issued fewer unnecessary control actions, which translated into progressively smoother and more stable power trajectories.

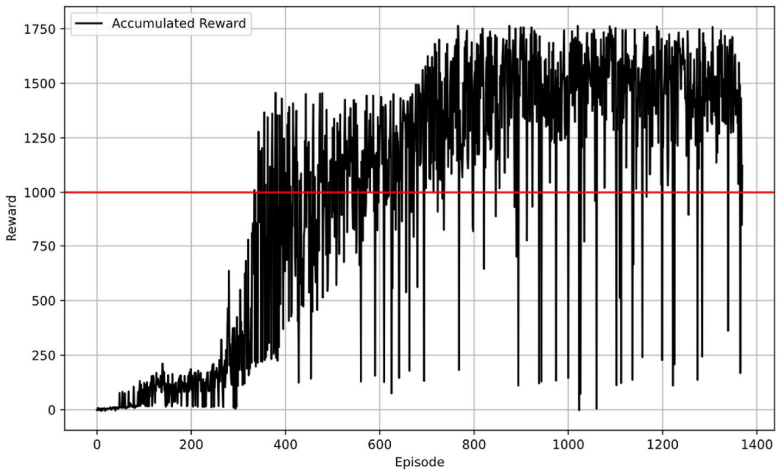


Fig. 6. Cumulative reward of the PPO agent per training episode (reproduced from [15]).

4.2. Execution of the Algorithm

In the demonstration, the operation unfolded in three consecutive phases that alternated between the two engines. In the first phase (0–1,169 s), the rule-based engine withdrew the control-rod banks to their target positions. This step lifted reactor power toward the level at which boron dilution would take over. Throughout this phase power stayed below 8 %, since dilution had not yet begun. In the second phase (1,169–18,604 s), the Robust AI-PPO agent assumed responsibility for boron dilution. It carried the

reactor toward its 8 % checkpoint while keeping the startup rate (SUR) below 0.5 dpm. In the third phase (18,604–27,847 s), the rule-based engine resumed and brought the plant to full power. It executed the remaining discrete actions—steam-flow adjustment, turbine run-up, generator synchronization, control-valve regulation, and the stepwise increase of turbine load. Reactor power climbed from 8 % to 100 %, and generator output reached 45 MWe. Fig. 7 plots reactor and generator power across the three phases. A comparison with the author-defined expected timeline yielded a Pearson correlation coefficient of 0.9978, which indicates that the operation followed the planned trajectory shape. Fig. 8 makes this match visible by overlaying the algorithm's reactor-power output on the expected timeline at each operation step.

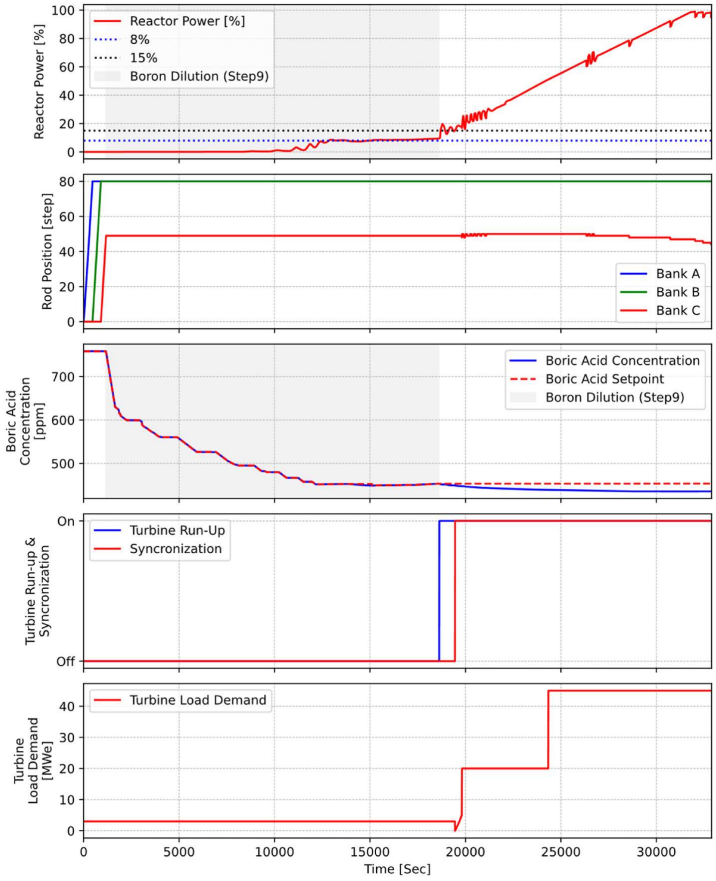


Fig. 7. Reactor and generator power during the autonomous power-increase operation (reproduced from [15]).

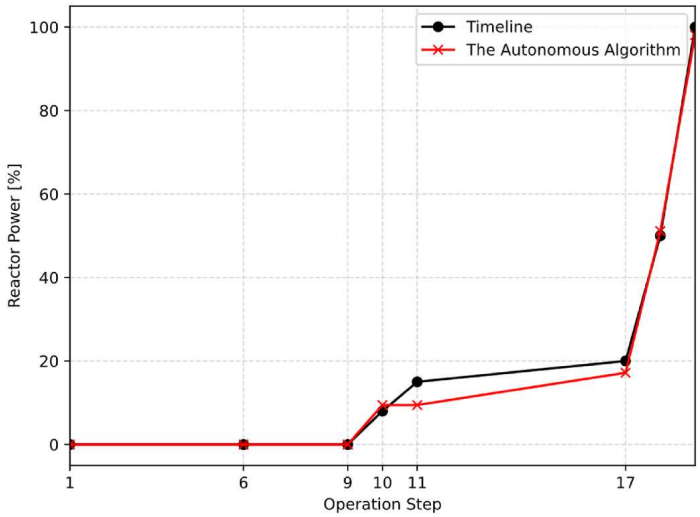


Fig. 8. Comparison of reactor power between the expected timeline (black) and the autonomous algorithm (red) at each operation step (reproduced from [15]).

4.3. Comparison of Three Control Approaches for Boron Dilution

To isolate the contribution of the proposed learning-based controller, the boron-dilution subtask was used as a common benchmark for three approaches: (i) a traditional if-then logic, (ii) a plain DNN-based PPO agent (DNN-PPO), and (iii) the proposed PPO agent with the robust-AI formulation (Robust AI-PPO). All three controllers operated on the same simulator from the same initial state. In every case the goal was to raise reactor power from 0 % to the interim 8 % target while staying within the operational limits. Their reactor-power trajectories are overlaid in Fig. 9. The 100 pcm reactivity gate is common to the if-then baseline and to the reward shaping used by both PPO variants. The comparison therefore isolates the action-selection mechanism, not the underlying safety envelope.

4.3.1. If-Then Logic

The if-then baseline reflects conventional procedural control: boron concentration is adjusted whenever the total reactivity drops below a fixed threshold of 100 pcm, triggering a dilution step. Every action is keyed to a specific condition, which keeps the controller straightforward to implement and to verify. In the experiment it reached the 8 % target quickly, at about 9,809 s, but its trajectory rose steeply and continued to overshoot, failing to settle near the target. Settling stably around 8 % would require expanding the rule set with gain-scheduled or transient-aware conditions. That path was not pursued here; the minimal one-rule logic is reported as a procedural baseline.

4.3.2. DNN-PPO

The DNN-PPO controller is a PPO agent whose actor and critic networks are built from fully connected layers that take the raw plant signals as numeric inputs. It is the natural learning-based counterpart to the if-then logic, allowing the controller to acquire its policy from interaction with the simulator rather than from hand-written rules. In the experiment it reached 8 % at about 12,161 s—slightly slower than the if-then baseline—but it too rose steeply and oscillated without converging stably near the target. The fully connected representation appears to lack the temporal context needed to taper dilution as power approaches the setpoint.

4.3.3. Robust AI-PPO

The Robust AI-PPO controller is the proposed approach. It augments the PPO agent with a representation that turns recent trends of the four key signals (reactor power, boron concentration, startup rate, and total reactivity) into color-coded trend images. A convolutional network then processes these images to extract temporal features, in parallel with a numeric vector path. This trend-based representation makes the controller sensitive to the rate and direction of change, not only to instantaneous values. In the experiment the agent approached 8 % more gradually, reaching the checkpoint at about 12,837 s. It then maintained reactor power within a ±1 % band around the target, a markedly tighter margin than either baseline. Above 8 % it adopted a "stay" behavior that simply held power steady, having learned that sustaining the target earned higher long-term reward. Notably, the agent acquired this stability from only the four trend inputs and the reward. It did not require the hand-engineered rule library the if-then approach would otherwise demand to cope with changing plant dynamics.

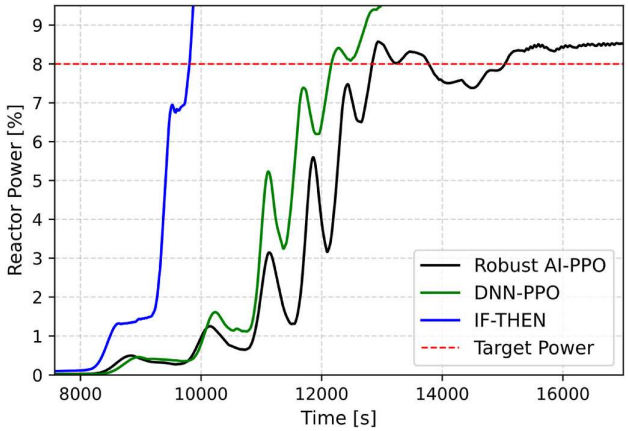


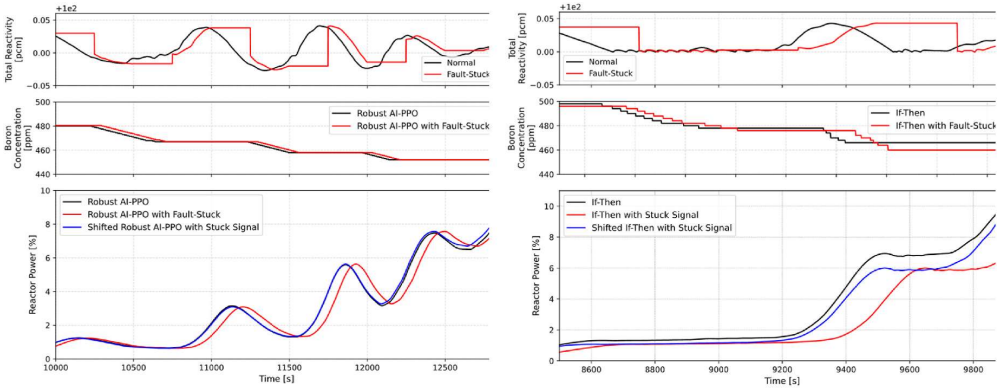
Fig. 9. Reactor-power trajectories during boron dilution for the three control approaches: Robust AI-PPO (black), DNN-PPO (green), and if-then logic (blue); the red dashed line marks the 8 % target power (reproduced from [15]).

Taken together, Fig. 9 highlights a consistent pattern. The if-then logic and DNN-PPO controllers prioritize speed of arrival at the target, and they pay for that speed with overshoot and unstable oscillations. The Robust AI-PPO trades a small amount of arrival time for the ability to converge and stay at the target. Numerically, the three controllers reached the 8 % checkpoint at about 9,809 s (if-then), 12,161 s (DNN-PPO), and 12,837 s (Robust AI-PPO). Only the Robust AI-PPO settled within a ±1 % band around the target. The advantage of the robust-AI formulation, then, is not raw speed but the smooth, stable behavior near the setpoint that is required for safe continuation of the power-increase operation.

4.3.4. Robustness Under a Fault Condition

To probe robustness, the Robust AI-PPO controller and the if-then baseline were exercised under a fault condition. The fault was produced by injecting a periodic stuck signal into the total-reactivity input, which is the key variable for dilution. DNN-PPO was excluded from this test because it had not settled even under nominal inputs, so the comparison here isolates the trend-image representation against the rule-based approach. Fig. 10 compares the Robust AI-PPO controller (a) with the if-then baseline (b) under normal and faulted inputs, showing total reactivity, boron concentration, and reactor power. The if-then logic, which keys its dilution actions directly to the corrupted reactivity reading, drove the boron concentration and reactor power onto trajectories visibly different from those of the normal case. The Robust AI-PPO controller, by contrast, showed only a slight delay relative to its own normal trajectory.

After accounting for the delay introduced by the corrupted input, the response shape was preserved. This pattern suggests graceful degradation rather than full immunity to the disturbance. The trend-image representation therefore confers a clear robustness advantage over the rule-based approach when sensor information is partially corrupted.



(a) (b)

Fig. 10. Performance under normal and fault (stuck-signal) conditions for (a) Robust AI-PPO and (b) if-then logic (reproduced from [15]).

4.4. Evaluation Against Design Objectives

The algorithm was designed to meet four objectives derived from the task analysis: (i) bring the reactor from 0 % to full power, (ii) maintain all operational safety limits throughout the run, (iii) produce smooth, stable boron dilution near the 8 % checkpoint, and (iv) tolerate plausible sensor faults on the key dilution input. Table 2 maps each objective to a quantitative or qualitative criterion and reports the corresponding outcome observed in the demonstration.

Design objective	Evaluation criterion	Result
(i) Complete the power-increase operation	Final reactor power reached	100 % reactor power; 45 MWe generator output
(ii) Safe operation	SUR < 0.5 dpm; no operational-limit violation	Satisfied across the 7 h 43 m run
(iii) Stable dilution near the 8 % checkpoint	Steady-state deviation from the 8 % target	Robust AI-PPO: ±1 % band (deviation 0.0121); if-then: deviation 0.3285 (≈27.15× larger); DNN-PPO: no stable settling
(iii′) Trajectory fidelity	Pearson r against the expected timeline	r = 0.9978
Time to checkpoint (informational)	Time to reach the 8 % target	If-then: 9,809 s; DNN-PPO: 12,161 s; Robust AI-PPO: 12,837 s
(iv) Robustness to stuck-signal fault	Trajectory-shape preservation under a corrupted reactivity input	Robust AI-PPO: graceful degradation; if-then: divergent response

Table 2: Evaluation of the algorithm against the four design objectives.

Quantitatively, the stability gap is sharp: the deviation from the 8 % target was 0.0121 for the Robust AI-PPO and 0.3285 for the if-then baseline, so the if-then deviation is about 27.15 times larger. The trajectory-fidelity Pearson coefficient of r = 0.9978 confirms a strong linear agreement with the expected timeline, although minor local mismatches remain (for example, the reference timeline calls for 15 % power at step 11 while the algorithm reaches 9.45 % at that point); the overall trend, direction, and relative magnitude are nonetheless highly consistent. For the fault criterion, the Robust AI-PPO's response under

the stuck-signal disturbance can be brought into close agreement with its nominal response by a single time-shift that aligns the peaks, indicating that the underlying control pattern is preserved—only delayed—rather than disrupted.

Qualitatively, the agent's behavior near the target resembles experienced operator practice: as power approaches 8 %, both the magnitude and the frequency of its dilution actions decrease, in keeping with the way operators taper boron adjustments to avoid overshoot. This adaptive, gradual style is a direct consequence of the trend-image representation, which encodes the rate and direction of change rather than instantaneous values, and of the reward shaping that discourages late-stage over-correction. Pairing this learned component with the rule-based engine for the discrete subtasks gives the algorithm both auditability (from the rules) and adaptiveness (from the policy). Combined with the quantitative outcomes in Table 2, these observations indicate that the four design objectives are met by the Robust AI-PPO controller. The if-then baseline and the DNN-PPO satisfy only the broader safety constraints (i) and (ii); neither meets the stability criterion (iii) at the 8 % checkpoint, and only the rule-based baseline was exposed to the fault test for (iv), where its response diverged from the nominal trajectory.

5. CONCLUSION

This paper described an algorithm that autonomously performs the power-increase operation of an SMR. A task analysis decomposed the operation and classified its subtasks as discrete or continuous, which directly shaped an algorithm that combines if-then rules with a PPO-based agent. On the iPWR simulator, the algorithm raised reactor power from 0 % to 100 % while keeping the startup rate below 0.5 dpm. Within this demonstration the boron-dilution subtask was used as a common benchmark for three controllers—a traditional if-then logic, a plain DNN-PPO agent, and the proposed Robust AI-PPO—and only the Robust AI-PPO settled within a ±1 % band of the 8 % checkpoint and degraded gracefully under a stuck-signal fault on its key input. The comparison suggests that the trend-image representation, rather than the choice of PPO alone, is responsible for this stability and robustness. By automating both the continuous and discrete subtasks of a long procedure, the approach provides a basis for future workload studies in multi-module SMR operation. Limitations include validation on a single iPWR simulator, a single fault scenario, and the absence of formal safety guarantees for the learned policy. Future work will extend the method to additional operations and to the coordinated supervision of multiple modules. Full methodological detail is reported in the authors' journal article [15].

ACKNOWLEDGEMENTS

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT)(No. RS-2025-02982991).

REFERENCES

[1] S. Choi, "Small modular reactors (SMRs): the case of the Republic of Korea," in Handbook of Small Modular Nuclear Reactors, Elsevier, pp. 425-465, (2021).

[2] H. O. Kang, B. J. Lee, and S. G. Lim, "Light water SMR development status in Korea," Nuclear Engineering and Design, vol. 419, 112966, (2024).

[3] J. Hartmann, J. Hyvarinen, and V. Rintala, "The operator and the seven small modular reactors," Nuclear Engineering and Design, vol. 418, 112929, (2024).

[4] E. L. Wiener and R. E. Curry, "Flight-deck automation: promises and problems," Ergonomics, vol. 23, no. 10, pp. 995-1011, (1980).

[5] C. E. Billings, "Human-Centered Aviation Automation: Principles and Guidelines," NASA Ames Research Center, NASA-TM-110381, (1996).

[6] R. E. Uhrig, "Opportunities for automation and control of the next generation of nuclear power plants," Nuclear Technology, vol. 88, no. 2, pp. 157-165, (1989).

[7] J. Kim, S. Lee, and P. H. Seong, "Autonomous Nuclear Power Plants with Artificial Intelligence," Springer, (2023).

[8] D. Lee, P. H. Seong, and J. Kim, "Autonomous operation algorithm for safety systems of nuclear power plants using long-short term memory and a function-based hierarchical framework," Annals of Nuclear Energy, vol. 119, pp. 287-299, (2018).

[9] D. Lee, A. M. Arigi, and J. Kim, "Algorithm for autonomous power-increase operation using deep reinforcement learning and a rule-based system," IEEE Access, vol. 8, pp. 196727-196746, (2020).

[10] S. J. Bae, H. H. Son, Y. Lee, and J. Yang, "Small modular reactor reinforcement learning framework: automating reactor core startup," Nuclear Engineering and Technology, vol. 57, no. 3, (2025).

[11] J. Bae, J. M. Kim, and S. J. Lee, "Deep reinforcement learning for a multi-objective operation in a nuclear power plant," Nuclear Engineering and Technology, vol. 55, no. 9, pp. 3277-3290, (2023).

[12] D. Lee, S. Koo, I. Jang, and J. Kim, "Comparison of deep reinforcement learning and PID controllers for automatic cold shutdown operation," Energies, vol. 15, no. 8, 2834, (2022).

[13] X. Chen and A. Ray, "Deep reinforcement learning control of a boiling water reactor," IEEE Transactions on Nuclear Science, vol. 69, no. 8, pp. 1820-1832, (2022).

[14] International Atomic Energy Agency, "Integral Pressurized Water Reactor Simulator Manual," IAEA-TCS-65, (2017).

[15] H.-J. Lee, D. Lee, and J. Kim, "Automating power-increase operation for small modular reactors based on task analysis with proximal policy optimization," Nuclear Engineering and Technology, vol. 58, 103767, (2026).

[16] J. Annett, "Hierarchical task analysis," in Handbook of Cognitive Task Design, CRC Press, pp. 17-36, (2003).

[17] N. A. Stanton, "Hierarchical task analysis: developments, applications, and extensions," Applied Ergonomics, vol. 37, no. 1, pp. 55-79, (2006).

[18] K. J. Vicente, "Cognitive Work Analysis: toward Safe, Productive, and Healthy Computer-based Work," CRC Press, (1999).

[19] R. S. Sutton and A. G. Barto, "Reinforcement Learning: An Introduction," 2nd ed., MIT Press, (2018).

[20] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," arXiv:1707.06347, (2017).