LLM-Based Integrated Control Agent System for Nuclear Reactor

Yoon Pyo Lee [a], Seung Geun Kim[a], Yonggyun Yu[a*]
[a]*Applied Artificial Intelligence Section, Korean Atomic Energy Research Institute, 111, Daedeok-daero 989 beon-gil, Daejeon 34057, South Korea*
[*]*Corresponding author: ygyu@kaeri.re.kr*

***Keywords :** nuclear reactor simulation, large language model, agent system, function calling, deep learning

1. Introduction

Nuclear energy plays a vital role in global energy demands due to its ability to provide a stable power supply and reduce greenhouse gas emissions. A nuclear power plant is a complex system that demands high levels of safety and operational continuity. Consequently, research has been conducted into autonomous operations to reduce the burden on operators and enhance overall system performance.

For instance, Lee et al. [1] proposed a technique that combines Deep Reinforcement Learning (DRL) with a rule-based system. Park et al. [2] applied DRL to automate the heat-up mode, which is still more manual than other modes. Additionally, Bae et al. [3] applied DRL to manage multiple objectives and devices during reactor operation.

There are also studies on using DRL for reactor diagnostics and predictions of nuclear reactors. Lee et al. [4] applied models based on Convolutional Neural Networks (CNNs) to detect abnormal reactor states. Y. H. Chae et al. [5] advanced this research using Graph Neural Networks (GNNs) for reactor diagnostics. J. Yang et al. [6] investigated predictions of reactor malfunctions using Long Short-Term Memory (LSTM) networks. In J. Bae et al. [7], DRL was utilized to predict key parameters in emergencies within a reactor.

However, these DRL models excel only at tasks for which they have been trained, limiting their capacity to handle multiple tasks with a single model. These fragmented DRL models can be integrated into a unified system using a Large Language Model (LLM). LLM is a model that demonstrates natural language understanding and generation capabilities by training on vast amounts of text data. LLM can perform tasks such as summarizing information based on provided text, engaging in dialogue, and assisting with document creation. By utilizing the function calling feature of LLM, it is possible to connect natural language requests with control functions. Function calling enables an LLM to act by invoking pre-written functions via natural language command. This integration enables the system to respond more diversely.

In this paper, we present an architecture that integrates LLM-based control with DRL models within a single framework. We utilized a light water cooled and moderated small integral pressurized water reactors (iPWR) simulator as distributed by the International Atomic Energy Agency (IAEA).

2. LLM-Based Integrated Control Agent System Development

2.1. Modules and Workflow

LLM Module receives natural language commands from users and invokes control functions through function calling. The Controller Module contains specific functions (e.g. reactor power demand setpoints, control rod adjustments). Diagnostic and Predictive Modules diagnose the state of the reactor and predict specific values using collected data. The Data Collection Module records real-time data every second in a CSV file after the reactor is started. The Data Collection Module operates without requiring user commands.

The basic workflow is as follows: 1) A user commands the LLM in natural language. 2) The LLM parses the command and through function calls, the corresponding function is executed in a control module. 3) Executed function from the Controller Module adjusts the settings. 4) The LLM reports back to the user in natural language.

When using a DRL model, 1) after the user commands, 2) the LLM utilizes function calling to execute the corresponding model function. 3) The DRL model is operated using a dataset made by the Data Collection Module. 4) And diagnosed, predicted data is reported by LLM to User. The workflow described can be observed in Figure 1.

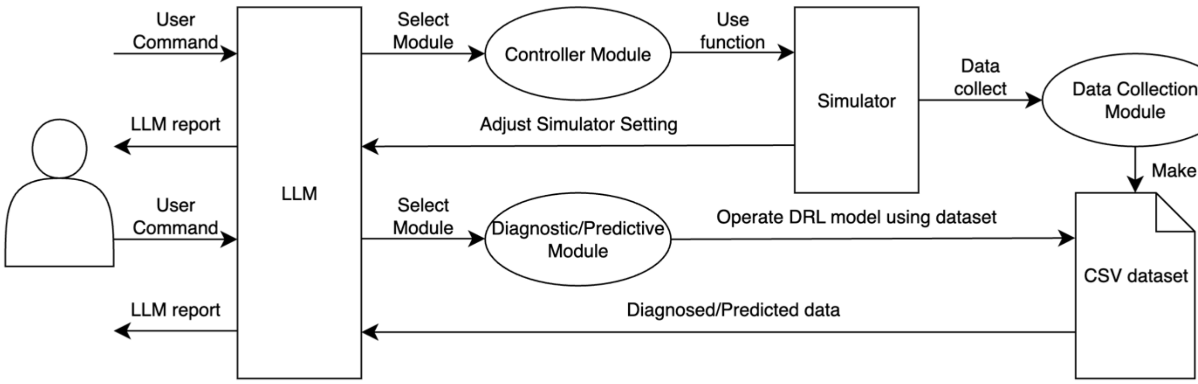

Fig. 1. System operation flow chart.

2.2. DRL Model Design

DRL models consist of the diagnostic model which requires snapshot data, and the predictive model which needs time-series data. The 62 variables and 11 scenarios were selected based on the simulator's Exercise Book.

The diagnostic model was developed using Multilayer Perceptron (MLP) to determine the current state of the simulator. The model correctly classified the No Malfunction case with an accuracy of 98.42% and it categorized the remaining classes woth an accuracy close to 100%.

The predictive model was developed using LSTM, with an input of 60 seconds for 62 variables, to predict the average coolant temperature over the next 140 seconds. The average coolant temperature was chosen as it is a crucial variable for safety, efficiency, and operational control. The results and predictions for each training session can be viewed in Figure 3. All predictions closely follow the overall trend with minimal error.

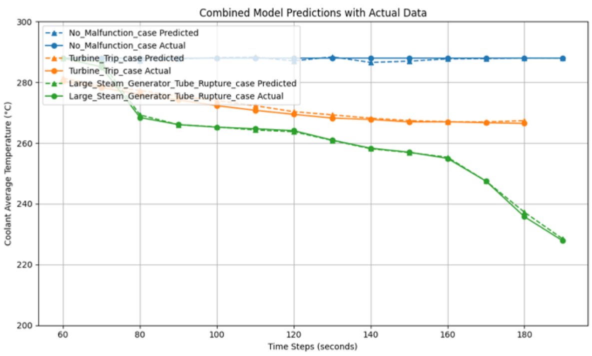

Fig. 3. Predicted values and errors for the next 140 seconds for the No Malfunction case.

2.3. Implementation of Function Call-based Control

In this iPWR simulator, variable values are located at specific addresses within a particular DLL file. The form of the read function takes the simulator's name, DLL file address, and variable address as inputs. The write function additionally accepts a new value to be written. By utilizing OpenAI's Swarm, roles are distributed to ensure more stable operation. Three agents are used, each serving distinct roles, a supervisor, a monitoring agent, and a scenario agent. The supervisor is configured to perform various actions such as heat up, cool down, and change power demand through logic functions that utilize basic functions. This setup enables it to execute multiple operations with a single command. The monitoring agent is composed of functions that provide information on the current state of the reactor. The scenario agent is designed to execute basic functions that operate the simulator in a sequence specified by procedures written in natural language.

3. Experiment and Results

3.1. Experiment

For our experiments, we utilized hardware comprising an Intel i7-13700 CPU and NVIDIA 4090 graphics card. The experiments were conducted: 1) heat-up, 2) procedure processing, and 3) autonomous decision-making. All experiments referenced the Exercise Book of the iPWR simulator.

Heat up: In this scenario, we will compare the performance of the system against human operation under identical conditions. This starts from a condition of 0% Beginning of life (BOL), in Natural Circulation (NC), and before withdrawing control rods. After fully drawing control rods A and B, withdraw control rod C until rod bank C reaches 49 steps. The control rods consist of A, B, and C, and the step scale ranges from 0 to 80, where 80 represents the fully withdrawn position. Then dilutes the boron concentration. Ensure that the SUR (Start Up Rate) does not exceed 0.5 dpm significantly, as this could lead to reactor shutdown. Finally, control rod C is gradually lifted until the neutron power reaches 8%.

Procedure processing: In this test, we will verify the system's ability to accurately perform function calling based on the given instructions. The scenario used in this experiment is the "Load Maneuvering (10%) in reactor leading mode" based on the Exercise Book. In brief, the scenario involves checking variables such as neutron power, start-up rate, and boron concentration.

After selecting the appropriate reactor mode, maneuvering the reactor power down to 90% and subsequently raising it back to 100%. The scenario starts differently from the heat-up scenario, beginning at a condition of 100% BOL, in NC. Figure 6. displays the actual experiment operation screen. Some steps of procedure and execution using function calling are observed.

Autonomous decision-making: This test will examine whether the system can autonomously make decisions when provided with current circumstances by the diagnostic model. The scenario begins similarly to the procedure processing scenario, starting from 100% BOL, NC. After loading the Large Main Steam Line Break with a severity of 10%, the system is commanded to 'Diagnose the reactor's condition and initiate a cool down if a shutdown is anticipated.' When diagnosing the simulator, it checks whether to initiate a cool down if a malfunction is detected.

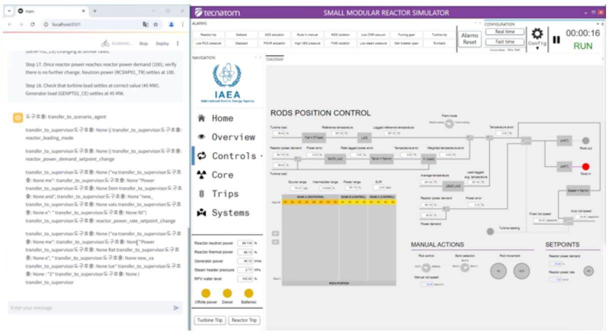

Fig. 4. The operation of procedure processing. Procedure steps 16, 17, and 18, as well as the execution of user commands by the LLM, are observed.

3.2. Results

The heat-up scenario experiment was conducted 10 times each by both a human and LLM. Figure 5. illustrates how long each instance of the experiment took. As shown in Figure 7, using the proposed system resulted in a 7.15% increase in speed compared to manual operation by humans. The variance is lower compared to human execution, the longest duration was around 1,700 seconds and the shortest was around 1,500 seconds, while all instances operated by the system were consistently around 1,500 seconds.

In procedure processing, after the procedure was inputted, the system asked the user for confirmation before executing each step, and upon receiving approval, carried out the actions according to the specified stages. The LLM occasionally displayed hallucinations, reporting actions it had not taken, such as checking variables or changing values. However, when explicitly instructed to execute the actions again, it performed the specified tasks. Figure 6. illustrates the neutron power over time during the system execution procedures. It followed the previously outlined scenario. Although functions for checking variables were insufficient to monitor all variables, it was observable that the output changed according to the scenario.

Figure 7. displays the neutron power over time and Figure 8. displays the chat screen during the last experiment. The first diagnosis was conducted without applying any malfunction. During this, the LLM identified it as No Malfunction and took no further action. During the malfunction state, the system diagnosed the issue and executed a cool down. When no action was taken, the nuclear power increased until a shutdown occurred. Conversely, when a cool-down was attempted, the neutron power decreased before the shutdown. Due to the limitations of the simulator, the cool-down was not successful, however, it is evident that the system recognized the need for a shutdown and attempted the cool-down.

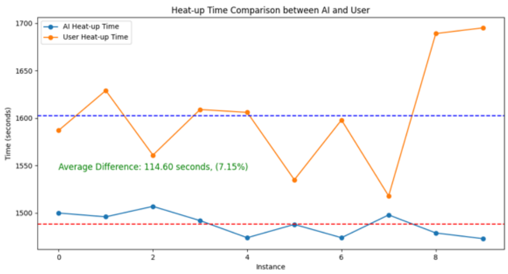

Fig. 5. The duration of each instance of the heat-up scenario execution and their average time.

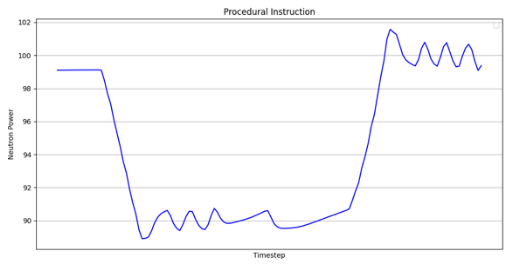

Fig. 6. The neutron power over time, when the system executes procedural instructions.

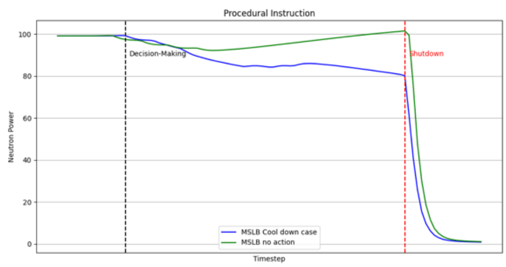

Fig. 7. Nuclear power over time when the decision-making scenario is executed by the system.

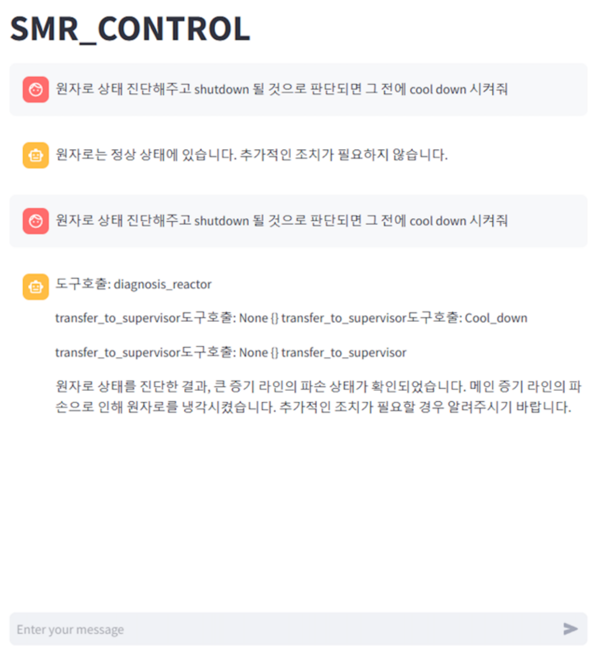

Fig. 8. The chat screen during the autonomous decision-making experiment.

4. Conclusions

This paper presents an integrated framework that utilizes the function calling capabilities of LLM to implement automatic control and employs DRL models for diagnostics and predictions. As discussed earlier, compared to human operation, the experimental results showed control stability and the ability to predict subsequent variables with an error of less than 1%. The capability to issue in natural language, along with the use of procedural instructions, demonstrated improvements in user convenience. Additionally, it was confirmed that autonomous decision-making was possible when information was provided through the DRL model.

While the current system has demonstrated advancements, there are several areas where further improvements and research are necessary. The limitation arises from the use of a training simulator, which constrains the complexity of scenarios that can be generated. As a result, it limits the system's ability to showcase its full potential in handling more complex, real-world situations. Due to the lack of implementation of all necessary functions, it was challenging to determine if the LLM performed all steps perfectly. Occasionally, it exhibited hallucinations, highlighting the need for a system to prevent such occurrences. The scenarios predicted were also limited and the predicted data generated was not utilized effectively. Moreover, there were limited aspects of the system that could be controlled, further restricting the demonstration of its capabilities.

Future research directions will explore various multimodal data sources, including video, audio, and textual documents, to develop more sophisticated autonomous control systems. Integrating these diverse data types aims to enhance the accuracy and reliability of predictive models and to expand the system's capabilities in complex environments.

ACKNOWLEDGEMENTS

This work was supported in part by Korea Atomic Energy Research Institute R&D Program under Grant KAERI-524540-25.

REFERENCES

[1] 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. 2020. doi: 10.1109/ACCESS.2020.3034218, 2020.
[2] J. Park, T. Kim, and S. Koo, Control automation in the heat-up mode of a nuclear power plant using reinforcement learning, Progress in Nuclear Energy, Vol.145, no. 104107, 2022.
[3] J. Bae, J. M. Kim, S. J. Lee, Deep reinforcement learning for a multi-objective operation in a nuclear power plant, Nuclear Engineering and Technology, Vol.55, pp. 3277-3290, 2023.
[4] G. Lee, J. Lee, and C. Lee, A convolutional neural network model for abnormality diagnosis in a nuclear power plant, Applied Soft Computing Journal, Vol 99, no. 106874, 2021.
[5] Y. H. Chae, C. Lee, C. Lee, S. M. Han, and P. H. Seong, Graph neural network based multiple accident diagnosis in nuclear power plant: Data optimization to represent the system configuration, Nuclear Engineering and Technology, Vol 54, pp. 2859-2870, 2022.
[6] J. Yang, and J. Kim, An accident diagnosis algorithm using long short-term memory, Nuclear Engineering and Technology, Vol 50, pp. 582-588, 2018
[7] J. Bae, G. Kim, S. J. Lee, Real-time prediction of nuclear power plant parameter trends following operator actions, Expert Systems With Applications, Vol 186, no. 115848, 2021.