Tri Dasa Mega Vol. 26 No. 3 (2024) 107-114

Jurnal Teknologi Reaktor Nuklir

Tri Dasa Mega

Journal homepage: jurnal.batan.go.id/index.php/tridam

Data Visualization in The Human-Machine Interface of Reactor Protection System Simulator

Tulis Jojok Suryono[1*], Nuri Trianti[1], Sigit Santoso[1], Sudarno[1], Kiswanta[1], Restu Maerani[1], Medila Kusriyanto[2]

[1]Research Center for Nuclear Reactor Technology, Research Organization for Nuclear Energy, National Research and Innovation Agency (BRIN) Gd. 720, KST BJ Habibie, Serpong, 15313, Indonesia

[2]Department of Electrical Engineering, Faculty of Industrial Engineering, Indonesia Islamic University, Jl. Kaliurang km 14.5, Sleman, Yogyakarta, 55584, Indonesia

ARTICLE INFO

Article history:

Received: July 16th, 2024
Received in revised form: August 8th, 2024
Accepted: August 14th, 2024

Keywords:

Data visualization
Reactor protection system
Python programming language
Human-machine interface
High-temperature gas-cooled reactor

A B S T R A C T

The Reactor Protection System (RPS) is critical to the operation of nuclear power plants as it monitors essential reactor parameters and initiates automated shutdowns when necessary. The human-machine interface (HMI) of the RPS is pivotal for enabling operators to efficiently monitor, analyze, and respond to complex data. This study aims to simulate the signal or data flow in the RPS of HTR-10, a high-temperature gas-cooled reactor (HTGR) with a thermal power output of 10 MWth, which contributes to deeper understanding of RPS functionality and to enhance the awareness of operators regarding the plant parameters status. The HMI panels, as well as sensor input data, were generated using Python programming language. The HMI of RPS successfully and comprehensively displays the values of key sensor inputs and their trip setpoints (neutron flux, helium temperature, primary coolant pressure) both on panels and real-time graphs. Moreover, it also shows the reactor status (normal or trip) based on the existence of an initiation trip signal in the RPS. Alarm panels are generated when the reactor is tripped. The RPS simulator can be used to guide future enhancements in reactor safety systems.

1. INTRODUCTION

Nuclear power plants are equipped with Reactor Protection Systems (RPS) to ensure reactor safety by preventing unsafe conditions and mitigating potential accidents [1]. The RPS continuously monitors reactor parameters and initiates protective actions if the parameters exceed the predefined safety limit, either by shutting down the reactor or initiating other safety measures to limit the impact [2, 3, 4].

Operators of NPP monitor and control the reactor through the human-machine interface (HMI) of a main control room. The HMI serves as the basis for human physical and cognitive processes within the plant, playing a critical role in ensuring safe and efficient plant operation [5, 6]. Reference [5] also emphasizes function analysis in HMI design, which identifies necessary information and controls for operational objectives that ensure safe and effective tasks and execution.

In addition, operator awareness of the reactor's operational state is increased by HMI's real-time data visualizations, warnings, and status indicators [7]. HMI helps operators make prompt and accurate decisions by displaying complicated data in an easy and accessible way, particularly in unusual

*Corresponding author.
E-mail: tuli002@brin.go.id
DOI: 10.55981/tdm.2024.7073

situations, thus enhancing situation awareness [8, 9]. Operators learn to respond effectively to alarms, diagnose issues, and take corrective actions, which leads to more effective and error-free operations due to ergonomic and user-friendly HMI design [10].

Modeling and simulation provide a comprehensive approach for RPS evaluation, enabling the assessment of the entire system response under diverse operating scenarios. Developing a detailed model of the RPS signal flow allows engineers to simulate its behavior during normal operation and anticipated transients, offering valuable insights into several crucial aspects of RPS performance.

The purpose of this study is to simulate the signal or data flow within an RPS into the graphical user interface (GUI) of HMI so that operators or users understand the functionality of the RPS, increase their awareness of the plant parameters status, and help them making decision and taking action based on the alarm generated on the HMI to mitigate the impact of the anomalies. The HMI was designed following the requirements. The RPS model and the sensor inputs used are for HTR-10, which has 3 channel redundancy and 2 out of 3 (2oo3) voting logic to generate the actuation trip signal. The GUI of the HMI, as well as the generation of sensor input signals, presentation of sensor inputs on the panel and real-time graphs, and generation of alarm, were created using Python programming language.

2. RPS DATA FLOW

The signal flow in a RPS is a critical aspect of ensuring safe operation of a nuclear power plant. The RPS is designed to monitor the reactor's status and execute protective actions when necessary to prevent unsafe conditions. Figure 1 provides a detailed breakdown of the signal flow process in an RPS.

2.1. Sensor Inputs

The RPS relies on various sensors to continuously monitor key reactor parameters. These sensors detect abnormal conditions that might indicate a potential hazard. The main types of sensors include neutron flux sensors, temperature sensors, and pressure sensors [11, 12].

2.2. Data Acquisition and Preprocessing

The signals from the sensors are transmitted to the data acquisition system. This system performs initial preprocessing tasks such as filtering noise, signal conditioning, and converting analog signals to digital formats for further processing.

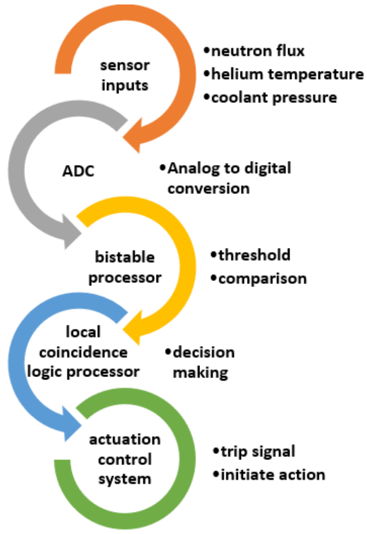


Fig. 1. Data flow in RPS [13]

2.3. Signal Processing and Control Logic

The preprocessed signals are then fed into the control logic units of the RPS. These units are responsible for interpreting the sensor data and making decisions based on predefined algorithms and setpoints. The control logic typically involves comparing sensor readings against predefined setpoints to detect any deviation from normal operating conditions. Moreover, utilizing multiple sensors and redundancy checks to ensure accurate detection and to mitigate false alarms. Voting logic helps determine the correct action when different sensors provide conflicting information.

2.4. Decision-Making and Initiation of Protective Actions

Once the control logic determines that a protective action is necessary, it generates initiation signals. These signals can trigger a variety of protective actions, including reactor scram, coolant system activation, pressure relief, and alarm systems.

2.5. Actuator Responses

The initiation signals are sent to actuators, which execute the physical protective actions. Actuators in an RPS include control rod drive mechanisms, valve actuators, and pump motors.

2.6. Feedback and Monitoring

After the protective actions are initiated, the system continuously monitors the reactor parameters to assess the effectiveness of the response. The feedback loop involves continuous sensor monitoring, control logic updates, and operator intervention.

3. METHODOLOGY

The methodology for data visualization in the HMI of RPS simulator involves several key steps: HMI design requirements, modeling the HMI of RPS, HMI of RPS development using Python programming language, integrating and testing the models, and analyzing the simulation results. These steps are summarized in Fig. 2.

4. RESULTS AND DISCUSSION

4.1. HMI Design Requirements

One of the requirements that is commonly used in HMI design is NUREG-700, Revision 3: "Human-System Interface Design Review Guidelines" [14], which provides comprehensive guidelines for designing and evaluating human-system interfaces in nuclear power plants. Specific parts of NUREG-7000 that are directly relevant to the HMI design requirements for an RPS simulator are provided in Table 1.

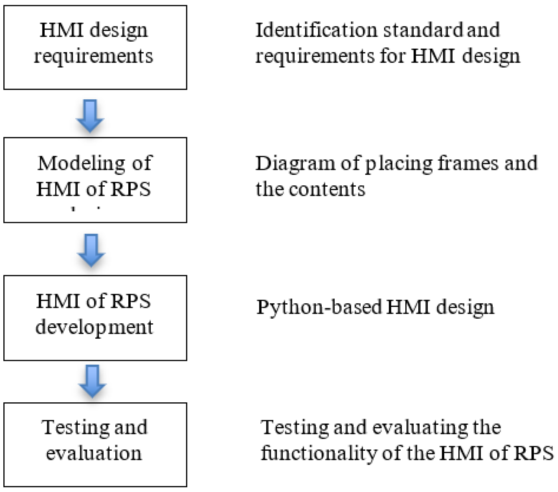


Fig. 2. HMI data visualization of RPS design methodology

Table 1. HMI of RPS simulator requirements

No	Part of NUREG-700	Description
1	Information display	Display information clearly and consistently to support operator tasks, and the use of appropriate coding techniques (e.g., color, symbols).
2	Use of color	Effective use of color to convey information and to avoid misinterpretation. It is important for indicating alarms and trip signals.
3	Parameters	Ensures that all critical parameters (like neutron flux, helium temperature, and coolant pressure) are displayed accurately and timely.
4	Status indication	ensuring that the current state of the system is conveyed to the operator.
5	Alarm presentation	alarms should be presented to the operator, including visual and auditory alerts
6	Real-time monitoring	system provides real-time monitoring capabilities for critical parameters.
7	Control mechanism	Design of control mechanisms (e.g., start/stop buttons, emergency shutdown) to ensure they are easily accessible and operable.

4.2. Modeling the HMI Design

Based on the requirements information gathered in the previous step, the layout of the HMI of RPS is created as can be seen in Fig. 3. Focus of the HMI design is how to visualize the flow of RPS data so that it can make it easier for operator or user to read and to understand. The HMI design layout consists of several frames:

- Alarm panel. This panel provides information on trip status and variable inputs that cause the trip.
- Sensor inputs and setpoint panel. It will display the value of sensor inputs (neutron flux, helium temperature, primary coolant pressure) and the trip setpoints.
- Logic processing and trip signal panel. It shows the existence of the initiation trip signal and the trip status.
- Control panel. It contains a START button to start the application or to run the program, a STOP

button to stop the program, and a CLOSE button to close the application.

- Real-time graph panel. This panel shows the trends of variable inputs in all RPS channels, as well as the setpoints.

Sensor inputs data

The HMI of the RPS simulator is intended to be used for all types of reactors including SMR. However, in this study, the sensor inputs and the trip setpoints are taken from inputs and trip setpoints data from the HTR-10 reactor. HTR-10 is an HTGR with a power of 10 MWth developed by Tsinghua University, China. HTR-10 is used for experimental and research reactor, specifically designed for technology development and testing. Table 2 provides the sensor inputs and trip setpoint data for the HMI of the RPS simulator. For simulation purposes and simplification, and due to the sensor input values being generated randomly within a specific range in the Python code, the sensor inputs and setpoint values were set in Table 2. However, the sensor input values and the setpoints are still in the range of plant values.

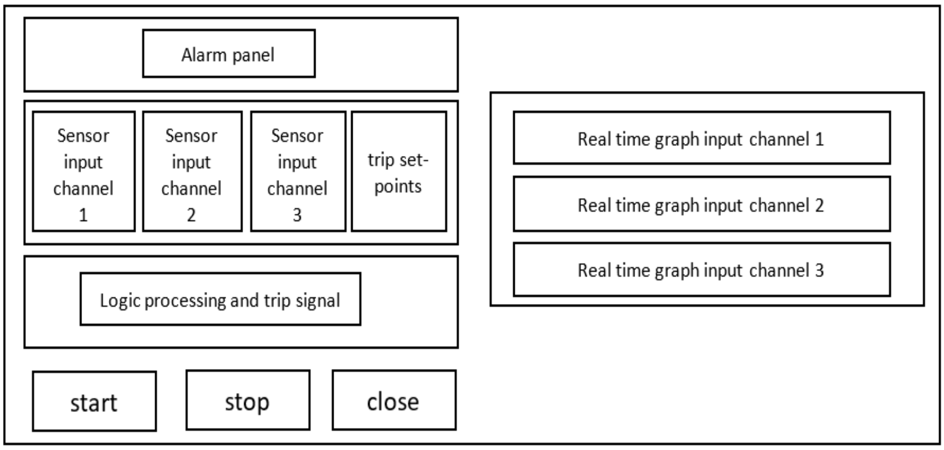


Fig. 3. Layout of HMI of RPS simulator

Table 2. Sensor inputs and trip setpoints data for RPS simulator [12, 13, 15]

No	Sensor inputs	Plant		Simulator	
		Range	Setpoint	Range	Setpoint
1	Flux neutron ($cm^{-2}\,s^{-1}$)	$5\times10^6 – 5\times10^8$	≥ 200%	$5\times10^6 – 12\times10^6$	11×10^6
2	Helium temperature (°C)	0 – 850	≥ 740	0 – 850	740
3	Primary coolant pressure (MPa)	3 – 4	4.4	2 – 6	4.4

4.3. Python-based HMI Development

The HMI was designed using Python programming language and various libraries such as the normal library to generate the sensor inputs, the *tkinter* library to create the graphical user interface, and the matplotlib library to produce the real-time graphs. Key components of the GUI are as follows:

- Frames: Different sections of the GUI for sensor inputs, logic processing, alarms, control, and graphs.
- Labels, Entries, and Buttons: GUI elements for displaying and interacting with sensor values, setpoints, initiation signals, and alarms.
- Real-Time Graphs: Plots showing neutron flux, helium temperature, and coolant pressure over time.

Table 3 shows the Python codes for developing the components of the HMI. It can be seen, for example, for generating the sensor inputs, the *"generate_sensor_inputs(self)"* class of Python is used. The code is used to generate the random values for neutron flux (range: 5×10^6 – 12×10^6 $cm^{-2}\,s^{-1}$), helium temperature (range: 0 – 850°C), and coolant pressure (range: 2 – 6 MPa). Below is a part of the code:

```
def generate_sensor_inputs(self):
 """Generate random sensor inputs for neutron flux, helium temperature, and primary_coolant_pressure."""
              return {
                 "neutron_flux": random.uniform(5e+6, 12e+6),
                 "helium_temperature": random.uniform(0, 850),
                 "coolant_pressure": random.uniform(2, 6)
         }
```

Table 3. HMI components and Python code

No	HMI components	Python code	Description
1	Sensor Input Generation	generate_sensor_inputs(self):	Generates random values for neutron flux, helium temperature, and coolant pressure
2	ADC Conversion	adc_conversion(self, analog_value):	Converts an analog sensor value to a digital value (integer).
3	Setpoint Comparison	compare_with_setpoints(self, digital_value, setpoint):	Compares a digital sensor value with the setpoint value to determine if the initiation signal should be triggered.
4	Actuation Trip Signal	determine_actuation_trip_signal(self, init_signals):	Determines if the actuation trip signal should be triggered based on the initiation signals
5	Updating Sensor Inputs	update_sensor_inputs(self):	If the simulation is running, it generates new sensor inputs, updates the display, and checks if the initiation signals and actuation trip signals should be triggered.
6	Blinking Alarm Buttons	blink_button(self, button)	Blinks the button to indicate an alarm
7	Program Control	start_program(self) stop_program(self) close_program(self):	Starts stops, and closes the protection system simulation
8	Updating Graphs	update_graphs(self, frame)	Updates the graphs with the latest data
9	User Interface Initialization	init_ui(self)	Initializes the user interface components, including frames, labels, entries, buttons, and real-time graphs.
10	Main Block	if __name__ == "__main__":	Creates an instance of the Tkinter root window. Initializes an instance of the HTGR Protection System class with the root window. Starts the Tkinter main loop to run the application.

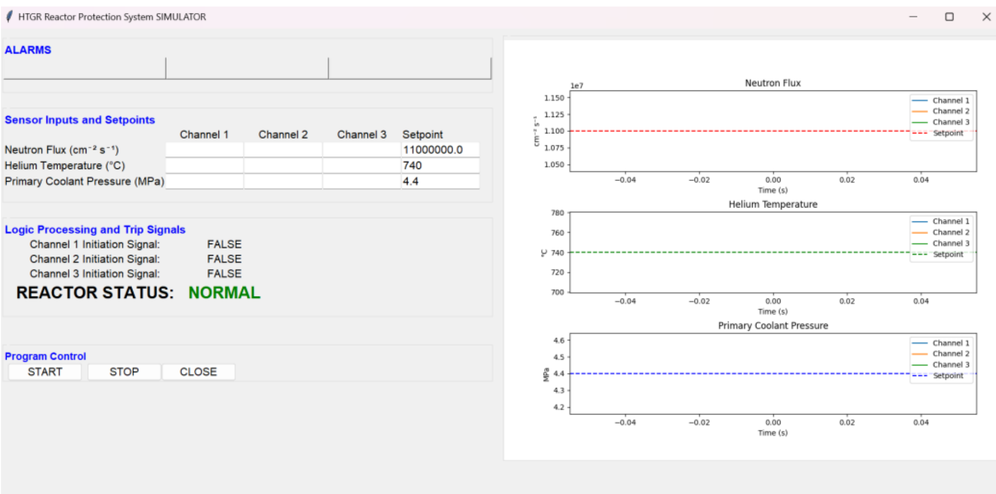


Fig. 4. Main view of the HMI of RPS

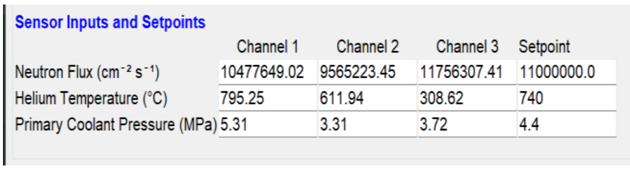
4.4. Case Study

In order to ensure that the HMI design is working properly, some functional and performance testing have been conducted. Functional testing was performed to verify that all features and controls operate as intended, while the performance testing was conducted to ensure smooth operation of the HMI.

Starting the application

The application is initiated when the START button on the control frame is clicked. This action triggers the *generate_sensor_inputs(self):* script to randomly generate sensor inputs (neutron flux, helium temperature, primary coolant pressure) for each channel within the predefined ranges. The results are displayed in the sensor inputs and setpoint frame as illustrated in Fig. 5. As an instance, at a specific time, the values of primary coolant pressure were 5.31 MPa (channel 1), 3.31 MPa (channel 2), and 3.71 MPa. The primary coolant pressure value in channel 1 exceeds the setpoint, indicating that the primary coolant pressure signal in channel 1 is considered an initiation signal.

.



Sensor Inputs and Setpoints

	Channel 1	Channel 2	Channel 3	Setpoint
Neutron Flux (cm^{-2} s^{-1})	10477649.02	9565223.45	11756307.41	11000000.0
Helium Temperature (°C)	795.25	611.94	308.62	740
Primary Coolant Pressure (MPa)	5.31	3.31	3.72	4.4

Fig. 5. Sensor inputs and setpoint values

Additionally, the generated sensor inputs are presented in real-time graphs, as shown in Fig. 6. The three graphs display the trends of sensor input values for neutron flux, helium temperature, and primary coolant pressure across three channels: channel 1 (blue line), channel 2 (orange line), and channel 3 (green line). The graphs illustrate that the signals fluctuate within a defined range. At certain times, the signal values exceed the setpoints (indicated by dashed lines) established by the bistable processors in each channel. These data visualizations demonstrate how sensor input signals are generated, processed, and compared to the setpoints to produce initiation trip signals in the bistable processors of the RPS.

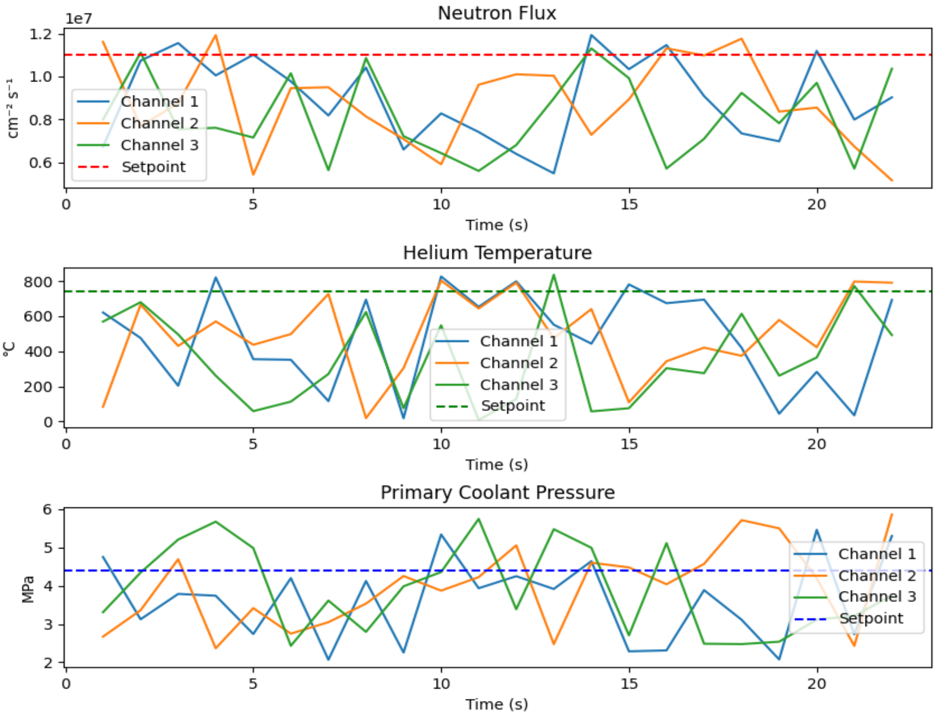


Fig. 6. Trend of sensor input values for all channels

Reactor Status

To improve safety and dependability, reactor protection systems frequently employ the "2 out of 3" voting logic design. The same parameter (such as neutron flux, temperature, or pressure) is usually monitored by three redundant sensors or detectors in this system. Before sending out a trip signal to shut down the reactor or perform other protective measures, the protection system requires confirmation from at least two of the three sensors regarding an abnormal state (such as a high neutron flux). This logic ensures that the system is less vulnerable to false alarms caused by a single sensor malfunction. If only one sensor identifies an abnormality, no action is taken. If two or three sensors detect the condition, the system interprets it as a true event and activates the protective response.

The HMI of the RPS simulates the reactor status into two (2) conditions: normal operation and trip operation. Normal operation involves conditions where there are no initiation trip signals in all RPS redundancy channels or if there are two channels without an initiation trip signal. On the other hand, trip operation occurs when there are initiation trip signals in all channels or 2 out of 3 channels have initiation trip signals. On the HMI of RPS, the absence of an initiation trip signal is indicated by the term "FALSE" while the initiation trip signal is indicated by the term "TRUE".

Figure 7 simulates the RPS during normal operation. It illustrates that all three channels have FALSE trip signals, indicating that none of the sensor input values exceed the trip setpoints in any RPS channel. The normal operation condition is further depicted in Fig. 8. In this scenario, despite one initiation signal being TRUE, meaning one sensor input value exceeds the trip setpoint in one channel, the reactor remains in normal operation.

Fig. 9 displays the HMI of the RPS during a trip operation. In this figure, there are two TRUE initiation trip signals, causing the alarm panel to turn red and blink. The alarm also identifies which sensor input values have exceeded the trip setpoints, leading to the reactor trip. Table 4 summarizes the potential of initial trip signals in all channels and the reactor trip status.

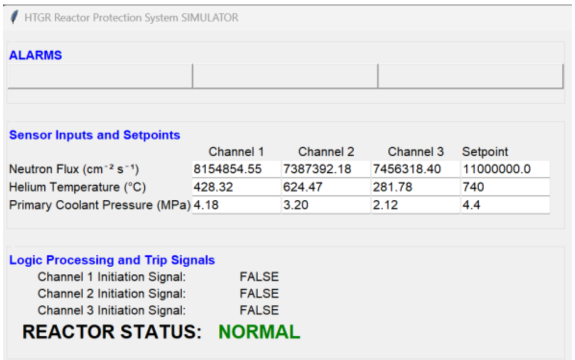


Fig. 7. Normal operation (3 FALSE initiation signals)

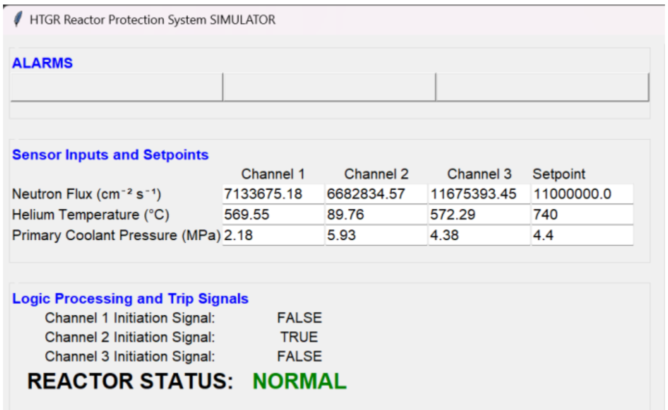


Fig. 8. Normal operation (2 FALSE initiation signals)

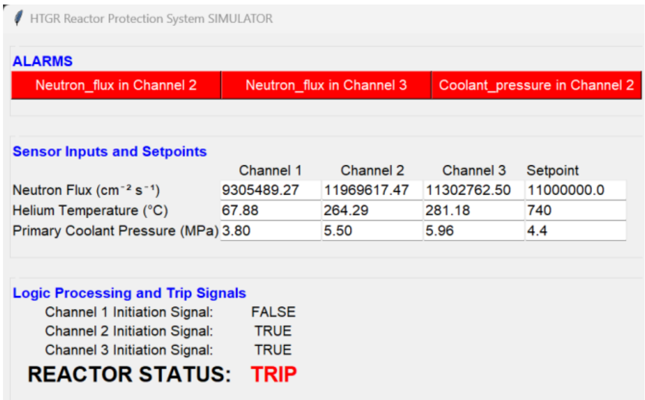


Fig. 9. Trip signal operation (2 TRUE 1 FALSE)

Table 4. Potential conditions in RPS

	Channel 1	Channel 2	Channel 3	Reactor status
	FALSE	FALSE	FALSE	Normal
	FALSE	FALSE	TRUE	Normal
	FALSE	TRUE	FALSE	Normal
Initiation signals	TRUE	FALSE	FALSE	Normal
	TRUE	TRUE	TRUE	Trip
	TRUE	TRUE	FALSE	Trip
	TRUE	FALSE	TRUE	Trip
	FALSE	TRUE	TRUE	Trip

Based on the case study, several points can be summarized from the HMI of the RPS simulator design, as follows:

a. The design can visualize the flow of data or signals in the RPS, including generating and processing sensor inputs, creating initiation and actuation trip signals.
b. The design facilitates users or operators to understand the functionality of RPS.
c. The design enhances the situational awareness of operators regarding the status of the plant, aiding them in making decisions and taking actions after a reactor trip based on the sensor inputs that cause the trip, thus preventing reactor core melting and the release of radioactive materials into the environment.

5. CONCLUSION

The modeling and simulation of signal flow in Nuclear Reactor Protection Systems provide valuable insights into the system's functionality and reliability. The results demonstrate that the HMI of RPS can generate the sensor inputs and display them on panels and trend graphs. Additionally, information on reactor status (normal or trip) is presented, along with a blinking alarm panel that indicates the sensor inputs initiating the reactor to trip. The results contribute to a deeper understanding of RPS functionality and guide future enhancements in reactor safety systems.

In the future, the HMI of RPS design should be enhanced, particularly in the method of generating the sensor input data, and by adding scenarios of anomalies.

ACKNOWLEDGMENT

This study is funded by the Rumah Program HITN of the Research Organization of Nuclear Energy of the National Research and Innovation Agency (BRIN) fiscal year 2024.

AUTHOR CONTRIBUTION

All authors equally contributed as the main contributors to this paper. All authors read and approved the final version of the paper.

REFERENCES

1. Jung J., Ahmed I. Development of Field Programmable Gate Array-based Reactor Trip Functions Using Systems Engineering Approach. Nucl. Eng. Technol. 2016. **48**(4):1047–57.
2. Ahmed I., Zio E., Heo G. Risk-informed Approach to the Safety Improvement of the Reactor Protection System of the AGN-201K Research Reactor. Nucl. Eng. Technol. 2020. **52**(4):764–75.
3. Qingzhu Liang, Mingxing Liu, PengXiao, YunGuo, Jun Xiao C.P. Reliability Assessment for a Safety-Related Digital Reactor Protection System Using Event-Tree/Fault-Tree (ET/FT) Method. Sci. Technol. Nucl. Install. 2020.:9.
4. Shouman M.A., Saber A.S., Shaat M.K., El-Sayed A., Torkey H. Dynamic Modeling of Reactor Protection System in Nuclear Power Plant for Reliability Evaluation Based on State Transition Diagram. Menoufia J. Electron. Eng. Res. 2021. **30**(1):13–21.
5. IAEA Human Factors Engineering in the Design of Nuclear Power Plants. IAEA SSG-51. 2019.
6. Wang C., Huang T., Gong A., Lu C., Yang R., Li X. Human-Machine Interaction in Future Nuclear Power Plant Control Rooms-A Review. IFAC-PapersOnLine. 2020. **53**(5):851–6.
7. Jia M., Zhang X., Huang H. Research on Nuclear Power Plant Safety Functional Requirements Analysis and Function Allocation. Int. Conf. Nucl. Eng. Proceedings, ICONE. 2018. **1**:1–4.
8. Miao R., Jia Q., Li D., Dong Z. A Comprehensive Situation Awareness Measurement Method for Analyzing the Operators' Situation Awareness of Multi-Module High-Temperature Gas-Cooled Reactor Plants. Energies. 2023. **16**(15)
9. Singh H.V.P., Mahmoud Q. LSTM-based Approach to Monitoring Operator Situation Awareness via HMI State Prediction. Proc. - IEEE Int. Conf. Ind. Internet Cloud, ICII 2019. 2019.(Icii):328–37.
10. Hinss M.F., Brock A.M., Roy R.N. Cognitive Effects of Prolonged Continuous Human-Machine Interaction: The Case for Mental State-based Adaptive Interfaces. Front. Neuroergonomics. 2022. **3**
11. Sudarno, Santoso S., Santosa K., Maerani R., Deswandri Assessment of Input Parameters and Architecture of RDE Reactor Protection System. J. Phys. Conf. Ser. 2019. **1198**(5)
12. Suryono T.J., Sudarno, Santoso S., Maerani R. Modelling of FPGA-based Reactor Protection Systems of an Experimental Power Reactor Modelling of FPGA-based Reactor Protection Systems of an Experimental Power Reactor. J. Phys. Conf. Ser. 2021. **2048**
13. Suryono T.J., Sudarno, Santoso S. Information Processing in the Reactor Protection Systems of High-Temperature Gas-Cooled Reactors. J. Teknol. Reakt. Nukl. Tri Dasa Mega. 2020. **22**(3):81–8.
14. O'hara J.M., Brown W.S., Persensky J.J., Lewis P.M. Human-System Interface Design Review Guidelines, NUREG-0700 Rev 3. 2020.
15. Lang M., Dong Y. Analysis on Anticipated Transient Without Scram (ATWS) Accidents of the HTR-10GT. Int. Top. Meet. Nucl. React. Therm. Hydraul. 2015, NURETH 2015. 2015. **9**:7844–57.