# REAL-TIME SUPERVISORY CONTROL IMPLEMENTATION OF SmAHTR POWER PLANT


**Jacob A. Farber and Daniel G. Cole**
Department of Mechanical Engineering and Materials Science
University of Pittsburgh
3700 O'Hara Street, Pittsburgh, PA
jaf166@pitt.edu; dgcole@pitt.edu



## ABSTRACT

This paper discusses an implementation of a supervisory control system for a nuclear power plant model using Small modular Advanced High Temperature Reactors (SmAHTR). The simulation operates in real-time using a network of computer agents, implemented on Raspberry Pi's (RPi's), where each major task is performed by one or more RPi's running calculations in parallel. The system architecture contains local, module, master, and operator actions to automate all aspects of the plant. Local level controllers run individual control loops to ensure the desired reference signals are met. The module level controllers set the local level reference signals within each SMR unit and detect sensor and actuator failures to adjust those reference signals as needed. Master level controllers balance plant power by calculating required heat power from each reactor and electric power from each generator and sending that data to the module controllers. Finally, the operator level contains an operator interface to summarize data and faults gathered through the controller levels, with options to review plant and reactor details as desired. This control implementation allows the plant to run autonomously, load follow a desired electrical power curve, and detect a growing number of fault scenarios. Ultimately, this type of supervisory control system will reduce labor-intensive surveillance and testing and allow fewer operators to more safely monitor an entire plant.



*Key Words*: Supervisory control, hierarchical control, hardware-in-the-loop, SmAHTR


## 1 INTRODUCTION

Small modular reactors (SMR's) will reduce capital costs in nuclear power plants by using modular designs to reduce the custom nature of reactors. However, based on current staffing and operator models, operations and maintenance (O&M) costs would remove any economic advantages of SMR's. Therefore, O&M costs must also be reduced for SMR's to be economically viable. In order to do this, new control schemes are necessary [1]. By using a supervisory control architecture that increases the automation of the plant, a control scheme can be designed for multiple operating conditions that also reduces O&M costs [2]. However, this new supervisory control architecture must be proven safe and effective before it can be implemented in an actual SMR. In order to carry out this validation, an inexpensive test-bed is desirable that can run a power plant simulation in real-time, while implementing the supervisory hierarchy.

This paper is focused on Oak Ridge National Laboratory's concept power plant called the Small modular Advanced High Temperature Reactor (SmAHTR) [3]. The University of Pittsburgh's Instrumentation and Controls Laboratory developed a Simulink Real-Time (R/T) model of the SmAHTR system that was used to implement the supervisory control structure. As a brief overview, the SmAHTR reactor is a small liquid-fluoride-salt-cooled modular reactor design with three in-vessel primary heat exchangers (PHXs). These PHXs transfer heat from the main circulating loop (MCL) to the intermediate cooling loop (ICL), which is connected to a thermal energy storage unit called the salt vault. This enables multiple reactors to be used in parallel to provide electricity, process heat, or both. In the simulation, four reactors are coupled to the salt vault on the power input side and three Brayton cycle generators and an ideal heat sink are coupled on the output side. The specific system models are not necessary for the scope of this paper.

In this paper, a powerful and inexpensive implementation of a supervisory control system is proposed for the SmAHTR power plant described above. First, the hardware required for the supervisory control implementation is discussed. Next, the supervisory levels are detailed, which are local, module, master, and operator levels. This setup allows us to model and run the power plant simulation in real-time and test each of the supervisory control levels to ensure proper plant performance.

## 2 SYSTEM ARCHITECTURE

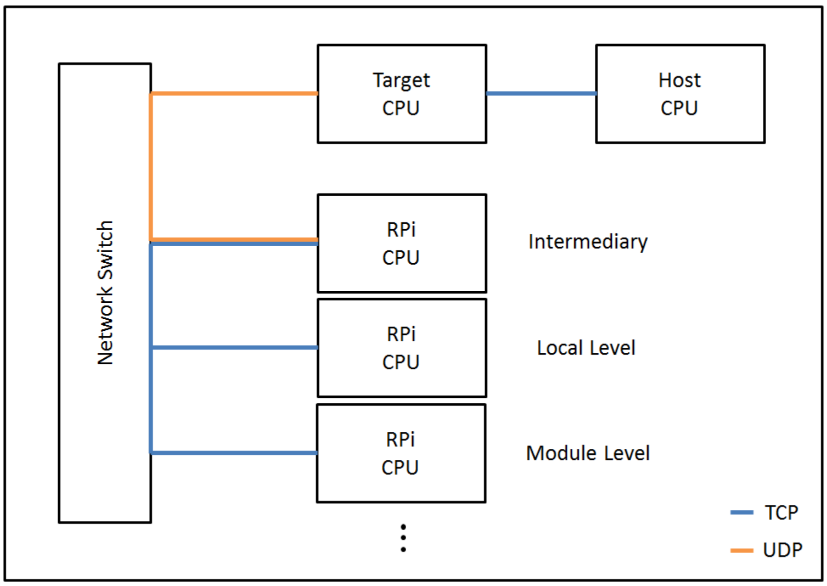


**Figure 1. System architecture showing host computer, target real time machine, Raspberry Pi computer agents, and Ethernet communications schematic.**

This supervisory control implementation uses a hardware-in-the-loop system architecture to monitor and control the SmAHTR power plant simulation in real-time. In order to do this, we integrate a host machine controlled by the user, a target machine running the mathematical model, and a network of computer agents implemented on Raspberry Pi's (RPi's). These components and their connections are shown in Figure 1. In this section, each of these components is detailed, as well as how they communicate.

The host machine is a Windows PC that starts the Simulink R/T simulation and enables the user to interact with the model in real-time. In order to start the simulation, the host machine compiles the Simulink model into C code to be run on the target machine. Once started, the host machine is used to view updates to displays and activate fault blocks manually.

The target machine runs the numerical calculations and has input/output (I/O) ports for communicating with other devices. In the setup, the plant simulation is an open-loop system; this means that without the rest of the hardware connected, there are no controllers in the plant. The target machine is a Speedgoat performance real-time target machine with an Intel core i5-680 processor. It has both Ethernet ports and an I/O 101 module, suitable for digital and analog communications respectively.

Finally, a series of computer agents in the form of RPi's, running the Raspbian operating system, are used to implement the supervisory control system in the loop. The hardware used was the RPi 3, which has a 1.2 GHz, 64-bit, quad-core ARMv8 CPU with 1 GB RAM. In addition, it has USB, HDMI, and Ethernet ports, simplifying connectivity. The primary advantages to using these types of agents are their low cost, high performance, and Ethernet connectivity. In addition, Raspbian is installed with Python, which is a powerful programming language and the one used for this implementation. Python is able to handle all socket commands for Ethernet communication as well as mathematics and logic required for the supervisory control system. Two specific RPi purposes will be mentioned here: (i) an intermediary between the target machine and the remaining RPi's and (ii) a user interface to connect with the remaining RPi's. The intermediary RPi was used due to communications limitations discussed below and the user interface RPi was used both in the supervisory control architecture discussed later and as a way to secure shell (ssh) to the remaining RPi's. These remaining RPi's are used to implement the supervisory control system.

In order for all the above components to interact, Ethernet communication was used. Initially, the I/O 101 module was used with each individual signal sent in analog form. However, the size of the system made this impractical, so digital communication was used instead. The actual routing was handled by a Netgear WNR2000 Ethernet router with a Netgear JGS524 Ethernet switch for additional communications ports. This hardware gives enough ports to connect 25 RPi or other Ethernet connected devices into the loop. The host and target machines communicate using Transmission Control Protocol (TCP), which guarantees the recipient will receive the packets in order and error-checks them. The target machine and the intermediary RPi communicate using User Datagram Protocol (UDP), which simply sends the packets with no guarantee of receipt or error-checking. This is a limitation of the Simulink R/T software. Some error-checking was added back in manually to improve communication. Finally, the RPi's use TCP amongst themselves.

## 3 SUPERVISORY CONTROL SYSTEM

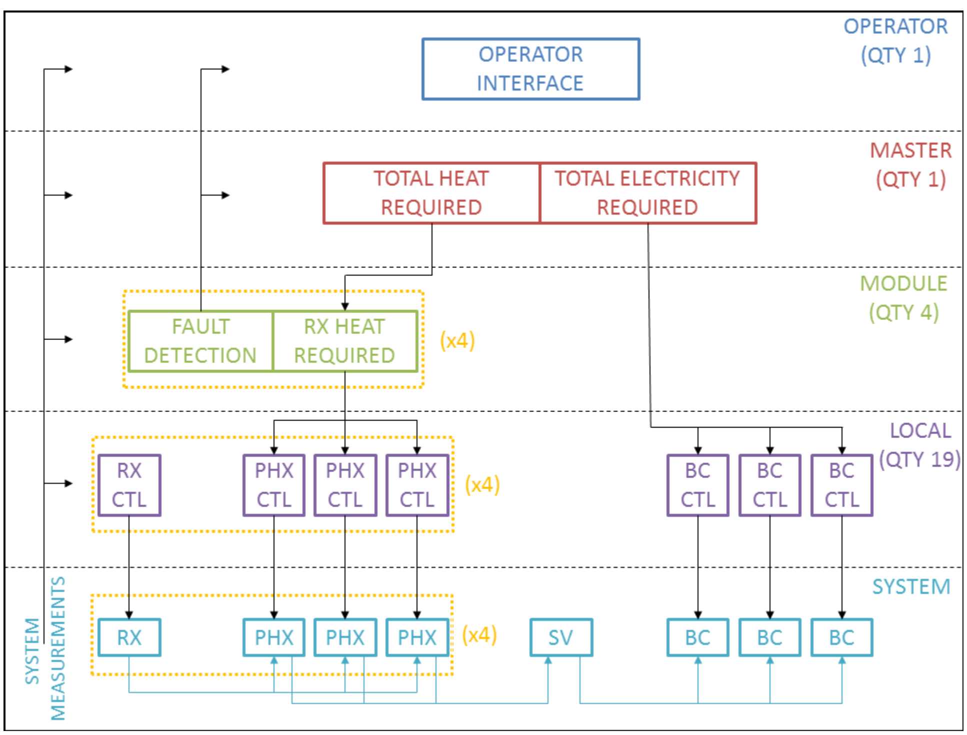


**Figure 2. Supervisory control architecture showing local, module, master, and operator level control actions.**

The previous section discussed the hardware setup required for the proposed supervisory control system in real-time. Here, the actual implementation of the supervisory control system is discussed. It consists of local, module, master, and operator levels, and each level handles a different aspect of the plant. When combined, they are able to monitor the plant as a whole. The supervisory control architecture is shown in Figure 2.

### 3.1 Local Level

The local level consists of (19) individual controllers, and this section starts by discussing their similar functions. To reduce complexity, these are all run on a single RPi; however, in an actual plant, these would likely be individual controllers. Their main purpose is as regulators that use a controllable input to achieve a desired system output. The different controllers are summarized in Table 1; while their details vary, their purposes are all the same. These regulators use proportional plus integral (PI) control techniques. Briefly, this means that the error and integrated error are used to set the controllable input value. The error is defined as the difference between the measurement and the desired measurement setpoint. This controller type works well because it is easy to implement and results in zero steady-state error. Further control theoretic details of this controller design are beyond the scope of this paper. For a more detailed explanation, refer to Nise [4].

**Table I. Summary of local level control measurements, purpose, controllable input, and quantity.**

| Measurement | Purpose | Controllable Input | Qty |
|---|---|---|---|
| Reactor temperature | Maintain a desired reactor outlet temperature | Control rod insertion percentage | 4 |
| PHX temperature and mass flow rate | Maintain a desired heat flow | Secondary mass flow rate | 12 |
| Electrical power output | Maintain a desired electrical power output | Heat exchanger mass flow rate | 3 |

The local level interacts with both the system below it and the module and master levels above it. With respect to the system, the local level uses system measurements to calculate the controllable input values as described above. These values are then sent back to the system to try and achieve the desired setpoint. Regarding upper levels, the local level receives setpoints for the PHX heat flow from the module level and setpoints for the Brayton cycle generators from the master level for use in the PI control algorithm. The local level receives these setpoints from different places because PHX's are considered sub-systems of an individual reactor module, whereas the generators are considered sub-systems of the plant as a whole.

### 3.2 Module Level

The module level consists of (4) individual controllers that supervise the reactor modules. These controllers have two main functions: (i) fault detection, currently detecting intermediate-loop pump failures and PHX fouling; and (ii) PHX heat flow setpoint allocation. The results from these actions are then sent down to the local level and up to the master and operator levels.

In order to detect faults, we are using parallel processing methods as described by Magill [5]. The techniques used are beyond the scope of this paper so only an outline will be given. Briefly, this entails having multiple models of a system that take the form of Extended Kalman filters. Each filter in the bank of filters has slightly different system parameters corresponding to different fault cases and makes estimates for the true state of the plant. Finally, the measurement and estimates are used along with Bayesian probability theory to identify the most likely model.

Using the above parallel processing techniques, we first look for PHX intermediate-loop pump failures. Here, failure means going from fully operational to not working at all. This failure is important because as soon as a pump fails, it stops removing heat from the reactor. Without the knowledge that the pump failed, only two thirds of the desired heat will be removed and will not be compensated for elsewhere. In addition to not transferring enough heat to the salt vault, the reactor will heat up until the local level controller is able to return it to equilibrium using the control rods. This could create a plant transient, which is a deviation from normal parameter operating values beyond an acceptable limit [6]. Based on a Nuclear Regulatory Commission study, plant transients are a major cause of core damage [7]. Quick detection and preventative actions could reduce the severity or prevent a plant transient entirely in the event of a pump failure.

We next use the above techniques to detect PHX fouling, which is the accumulation of undesirable material in the heat exchanger. This is important because increased buildup reduces heat exchanger efficiency, requiring more input energy to transfer heat from the reactor to the salt vault. While this is a much slower process than the pump failure, it still provides information for the operator to schedule maintenance. Although not currently done, this can also provide a tool for how to best allocate the heat load among the three PHX's in a given reactor.

In addition to fault detection, the other goal of the module level is to distribute the required heat load to the individual PHX local level controllers. The module level controller does not consider whether the available pumps are sufficient for safe operation; this is done at the master level. Instead, the controller simply uses the pump fault statuses to evenly distribute the required heat load among the functioning PHX's.

In order to accomplish the fault detection and load distribution, the module level connects with all other levels in the supervisory control scheme. First, it must receive system measurements in order to run the fault detection algorithms. Next, it sends that fault information up to the operator and master levels for higher level decision making. It also receives the total heat load required, which comes from the master level. Finally, it sends the individual PHX setpoints down to the local level controllers.

### 3.3 Master Level

The master level looks at the system as a whole unit and is primarily concerned with balance of plant. This is broken into balancing heat power, electrical power, and a heat sink to remove extra heat from the salt vault if needed. In order to incorporate the system as a whole, the master level interacts with all other levels in the supervisory control system.

First, the balance of plant with respect to the heat is discussed. In the power plant, the salt vault is a thermal battery, where the temperature represents its total stored energy. Using this intermediary between the reactors and power grid is advantageous because the reactors can run at a constant power level and the salt vault can fluctuate in temperature to take care of the cyclical, daily, energy demand. In order to accomplish this, the reactors run at an average load to essentially charge the salt vault during below average energy usage and discharge it during above average energy usage. We are assuming knowledge of the energy demand curve to calculate this average.

Next, the total heat must be allocated among the four reactors. The SmAHTR reactor concept was designed to be able to operate normally with only two working PHX's. In the current implementation, we have assumed that at least two pumps are working so that the reactors can always run at full capacity if needed. Therefore, the master level controller divides that heat load evenly among the four reactors without worrying about overloading a reactor.

The balance of plant for the electrical power refers to ensuring that the Brayton cycle generators are producing the desired rate of electrical power. In reality, this value is sent from the utility company based on economic dispatching. As mentioned above, we have assumed an energy demand curve to estimate the average reactor power. This total electrical demand is divided evenly among the three generators representing fully functional and economically equal conditions.

The final agent in the balance of plant is the heat sink, which removes heat from the salt vault as needed. This can be used for plant startup, shutdown, or as a safety precaution when the salt vault reaches an upper limit. In our simulation, the heat sink is modeled as a simple heat exchanger that cools its working fluid down to ambient temperatures. The master level controls this pump in its efforts to ensure proper balance of plant.

As the master level performs the balance of plant functions, it interacts with all levels of the supervisory control system. As mentioned previously, it sends individual reactor heat loads down to the module level to be distributed among the PHX's. In addition, it sends the individual generator power setpoints down to the local level. Finally, it sends all this information up to the operator level.

## 3.4 Operator Level

The operator level is a human-machine-interface (HMI) that gathers data from the system and other levels and displays it in an organized fashion. It is organized into a summary page, a balance of plant page, and more detailed system pages. The main goal of the operator level is to provide a real-time summary of the plant operation cleanly so that the operator can make decisions beyond the scope of the supervisory control system.

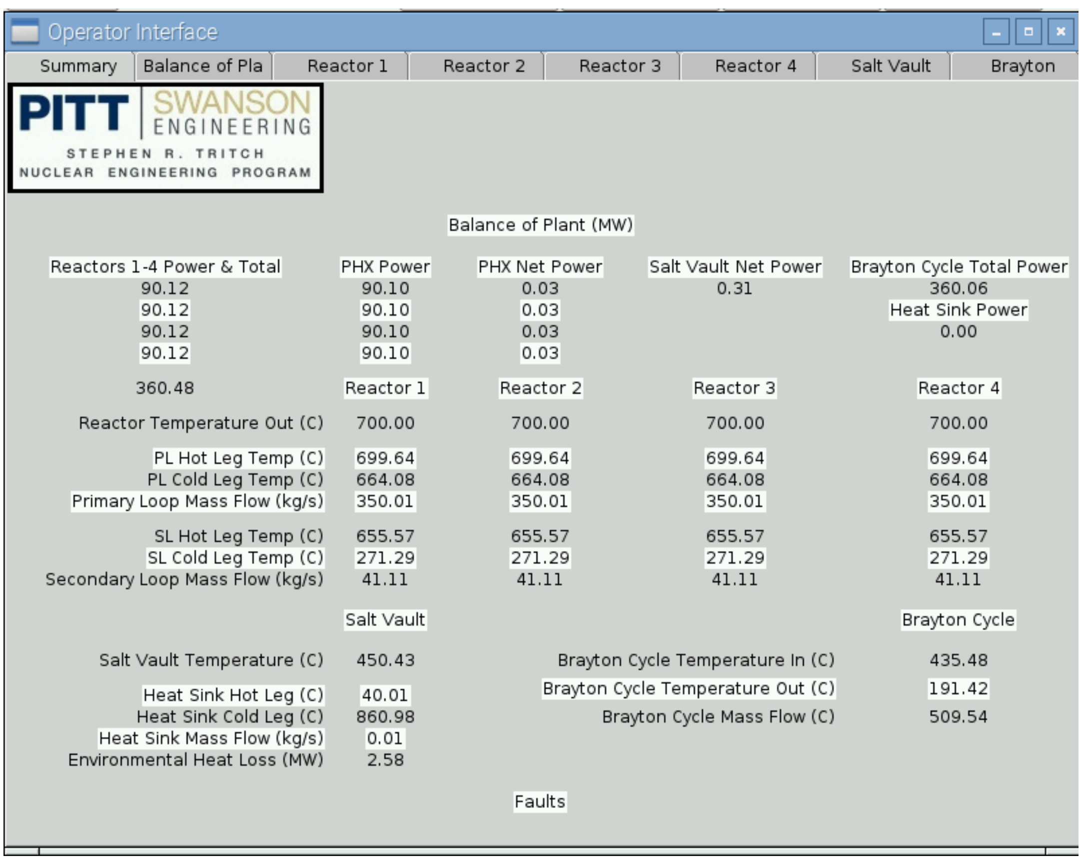


**Figure 3. Screenshot of the operator interface showing the summary page.**

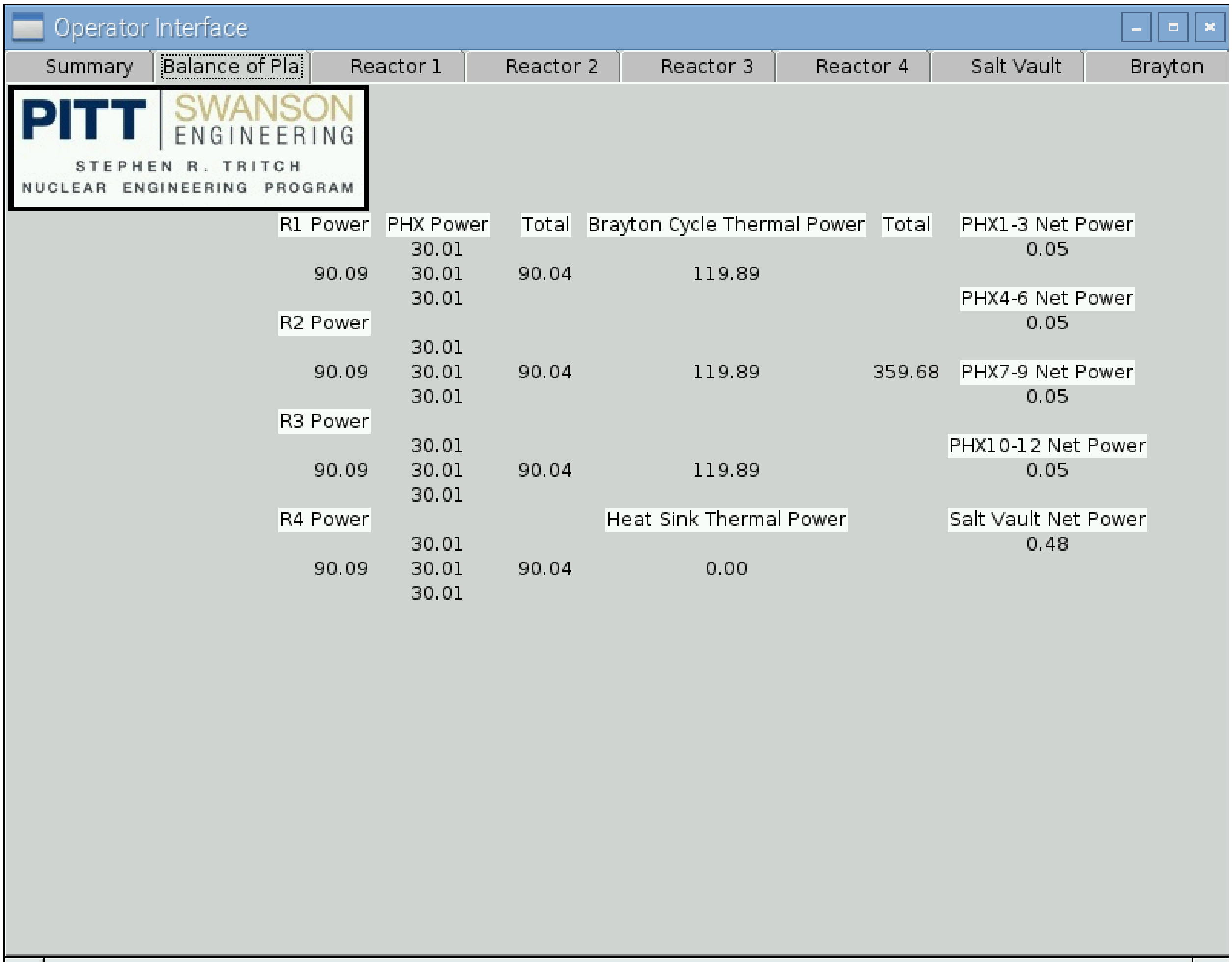


**Figure 4. Screenshot of the operator interface showing the balance of plant page.**

The different tabs of the supervisory control system enable the operator to view either large scale overviews, such as the summary and balance of plant, or more detailed system-specific data. The summary page, found in Figure 3, shows an overview of the instantaneous sensor measurements, balance of plant calculations, and a fault summary. This is useful for general purpose overview for plant health. To get a slightly more detailed overview, the operator can switch to the balance of plant page, found in Figure 4. This allows an operator to ensure that heat is properly flowing through the system, which is critical for plant functionality and safety. Finally, the operator can view system-specific details. A detailed page for a reactor is shown in Figure 5, including the real-time sensor measurements and a plot of past measurements. Using radio buttons, the operator can toggle between sensor plots or view multiple plots next to each other.

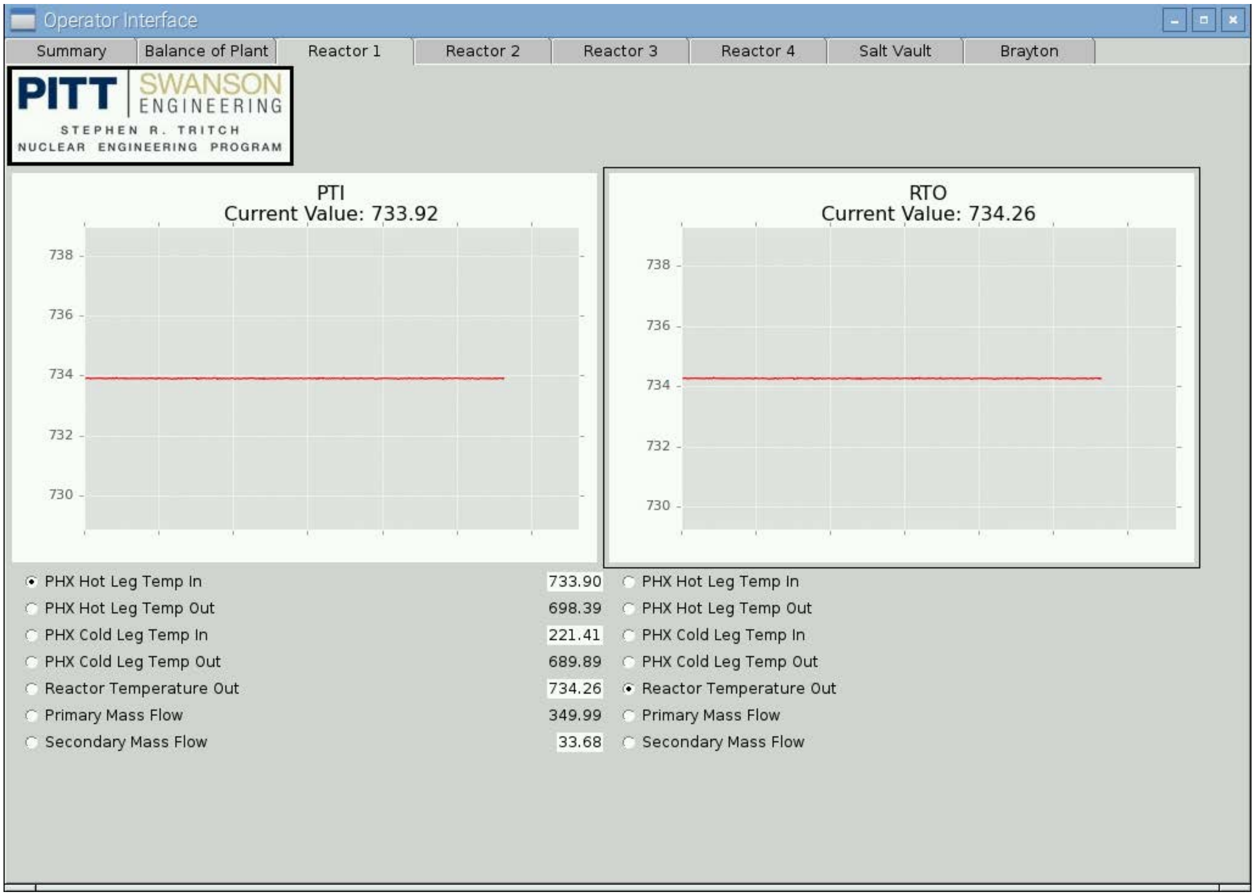


**Figure 5. Screenshot of the operator interface showing a detailed reactor page.**

## 4 SUMMARY

This paper details the hardware requirements for a real-time hardware-in-the-loop simulation as well as the implementation of a supervisory control system on a SmAHTR power plant. The hardware setup uses Simulink R/T and RPi's running Python, allowing for a powerful and inexpensive method for implementation. The supervisory control structure has local, module, master, and operator levels to automate, fault check, and balance the plant.

While the results of this implementation are beyond the scope of this paper, the supervisory control architecture was able to effectively perform its desired functions. The current implementation can run the simulation, initiate faults, detect a subset of those faults, and respond accordingly. For additional details of the techniques used and the results, please refer to the interim reports for DoE NEUP grant number DE-NE0000739 documenting these.

Supervisory control structures will be necessary as nuclear power plants move towards automation. Nuclear power plants are large and complex systems with many layers of operation and high safety demands. Using the supervisory hierarchy, multiple safety checks can be implemented to ensure that they can be safely run by fewer operators. This will ultimately lower O&M plant costs, helping plants become more cost effective for future use.

Future work involves adding more features to the current supervisory control system. For example, sensor failure detection capabilities could be included to increase the number of problems the supervisory control system can detect. In addition, more work could be done regarding the economic dispatching problem. For example, the system could determine how many generators to have on and how to best allocate them based on cost curves. Similarly, the PHX allocation could be optimized as they foul to minimize pump wear and input energy. Finally, benefits could be examined to having the parallel reactors, including compensating for shutting a single reactor down due to maintenance or xenon poisoning.

## 5 ACKNOWLEDGEMENTS

This material is based upon work supported under a Department of Energy, Office of Nuclear Energy, Integrated University Program Graduate Fellowship. Any opinions, findings, conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the Department of Energy Office of Nuclear Energy.

## 6 REFERENCES

1. Dwight Clayton and Richard Wood, "The role of instrumentation and control technology in enabling deployment of small modular reactors," *Seventh American Nuclear Society International Topical Meeting on Nuclear Plant Instrumentation, Control and Human-Machine Interface Technologies*, Las Vegas, Nevada, November 7-11 (2010).
2. Bojan Petrovic and Gary Storrick, "Supervisory control strategy development," *Technical Report STD-AR-07-01*, Westinghouse Electric Company LLC, February (2007).
3. S. R. Greene, J. C. Gehin, D. E. Holcomb, J. J. Carbajo, D. Ilas, A. T. Cisneros, V. K. Varma, W. R. Corwin, D. F. Wilson, G. L. Yoder Jr., A. L. Qualls, F. J. Peretz, G. F. Flanagan, D. A. Clayton, E. C. Bradley, G. L. Bell, J. D. Hunn, P. J. Pappano, and M. S. Cetiner, "Pre-Conceptual Design of a Fluoride- Salt-Cooled Small Modular Advanced High-Temperature Reactor (SmAHTR)," *Technical Report ORNL/TM-2010/199*, Oak Ridge National Laboratory, December (2010).
4. Norman S. Nise, *Control Systems Engineering*, Wiley, fourth edition (2004).
5. D. Magill, "Optimal adaptive estimation of sampled stochastic processes," *IEEE Transactions on Automatic Control*, **10(4)**, 434-439 (1965).
6. Sacit Cetiner, Michael Muhlheim, George Flanagan, and Richard Wood, "Level-0 PRA: Risk-informing the nuclear power plant I&C system", *Nuclear News*, February (2015).
7. "Reactor Safety Study: An Assessment of Accident Risks in U.S. Commercial Nuclear Power Plants," *Technical Report WASH-1400*, U.S. Nuclear Regulatory Commission, October (1975).