ORNL/TM-2023/2971

Development of Microreactor Automated Control System (MACS): Surrogate Plant-level Modeling and Control Algorithms Integration

Wesley C. Williams
Ian Greenquist
Pradeep Ramuhalli

June 2023

Website www.osti.gov

Reports produced before January 1, 1996, may be purchased by members of the public from the following source:

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
Telephone 703-605-6000 (1-800-553-6847)
TDD 703-487-4639
Fax 703-605-6900
E-mail info@ntis.gov
Website http://classic.ntis.gov/

Reports are available to US Department of Energy (DOE) employees, DOE contractors, Energy Technology Data Exchange representatives, and International Nuclear Information System representatives from the following source:

Office of Scientific and Technical Information
PO Box 62
Oak Ridge, TN 37831
Telephone 865-576-8401
Fax 865-576-5728
E-mail reports@osti.gov
Website https://www.osti.gov/

ORNL/TM-2023/2971

Nuclear Energy and Fuel Cycle Division

DEVELOPMENT OF MICROREACTOR AUTOMATED CONTROL SYSTEM (MACS): SURROGATE PLANT-LEVEL MODELING AND CONTROL ALGORITHMS INTEGRATION

Wesley C. Williams
Ian Greenquist
Pradeep Ramuhalli

June 2023

Prepared by
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, TN 37831
managed by
UT-BATTELLE LLC
for the
US DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

This report discusses progress on the modeling and integration task as part of the development of Microreactor Automated Control System (MACS). What follows is a discussion of the software tools used to develop a surrogate microreactor plant-level model, MACS module development, and a summary of the findings from integration with a hardware-in-the-loop (HIL) setup at Idaho National Laboratory (INL). Oak Ridge National Laboratory worked with INL to understand available hardware at INL (such as existing control platforms). With this information and using a preliminary framework for MACS with software interfaces defined, surrogate models and initial automated control algorithms were created for integration into the hardware to provide a HIL demonstration platform for MACS. Available reduced order reactor models that leverage existing microreactor neutronics and thermal hydraulics models have been integrated in the surrogate plant-level model. The surrogate model has been evaluated for sensitivity of all parameters and the MACS software modules (including the surrogate model) have been demonstrated to interact with the hardware. The integration, however, pointed to the need for further improvements to the framework—and specifically improvements to the interfaces to ensure that the HIL simulator is capable of longer-term stable operation with MACS in the loop. These improvements will be the focus of future research.

1. INTRODUCTION

Microreactors are nuclear power reactors with design power outputs less than ~100 MW_{th} and are of interest given their potential use toward meeting energy needs in remote and grid-isolated communities. Current concepts are designed for mobility and are expected to demonstrate longer and more flexible operating cycles from a single fuel load [1]. Most current microreactor concepts are higher-temperature concepts (higher than those typical of light-water reactors) and are being designed for load following and on-demand power [2]. Demonstration concepts include the MARVEL demonstration reactor [3] and concepts being developed under Project Pele [4].

Automated control technologies are among the advanced technologies being considered for enabling economical microreactor deployment and operation. This report describes an ongoing research effort to develop a Microreactor Automated Control System (MACS) framework that utilizes expected reactor inputs and outputs such as reactor temperature, control element (drum or rod) position, coolant temperature and energy transfer to heat sink, and factors such as reactivity feedback. The focus of the research is on development and demonstration activities for MACS that leverage existing designs for microreactors as well as development a set of software modules that can be integrated with a hardware-in-the-loop (HIL) simulator. Such integration is expected to result in a testbed capability for developing and evaluating advanced control algorithms by various stakeholders (industry and the researchers), thereby benefiting the microreactor community and accelerating the deployment of microreactors.

Key to the MACS framework development is the use of software platforms to develop a surrogate microreactor plant-level model that forms the basis for HIL integration and automated control algorithm development. The surrogate models incorporate available microreactor neutronics and thermal hydraulics models. Oak Ridge National Laboratory (ORNL) worked with Idaho National Laboratory (ORNL) to understand available hardware at INL (such as existing control platforms). With this information, software surrogate models were created for integrating into the hardware to provide an HIL demonstration platform for MACS. The surrogate model has been demonstrated to interact with the hardware and has been evaluated for sensitivity of all parameters, with initial tests indicating the potential for a MACS-driven control of the hardware system. This document describes the MACS framework, interfaces, and surrogate models; furthermore, it briefly describes the integration results as well as key remaining issues that must be resolved to achieve a robust MACS testbed.

2. MACS FRAMEWORK AND INTERFACES

Prior work on MACS has identified several requirements and has proposed a separation of the basic reactor control and protection functions from a higher-level coordination layer that can incorporate the automation functions. This allows the development of the control automation function as a non–safety-related system, meeting all existing requirements for isolation of safety and non-safety systems. The separation imposes requirements for MACS not to perform any safety-related functions, not to interfere with the function of any safety related system, and not to override any operator commands, resulting in a classification of MACS as a Level 2 or Level 3 (depending on task) automation system [5, 6]. Therefore, the design of MACS will need to include the ability to switch over from fully automated to fully manual operation, when necessary, without challenging the safety of the reactor.

It is expected that MACS will rely on the use of models of the whole plant (including the reactor itself) and leverage functions provided by the reactor power control and power conversion unit control systems. The use of such models enables simulating the effect of one or more actions on plant behavior before actuating one or more control elements.

Given the need to include plant-level models and interaction with lower-level control systems, the higher-level MACS software framework will need to include the necessary software interfaces to allow communication of necessary information between the various software modules. For initial demonstration purposes, and to ensure compatibility with the HIL setup being developed at INL, the relevant interfaces must enable communication among the following software platforms.

- LabVIEW: This is a design and development environment from National Instruments, and is the platform being used by INL for data acquisition and control of the laboratory hardware testbed. The testbed includes a control drum mockup with the ability to measure and control the drum position. The lower-level drum controller uses a proportional–integral–derivative (PID) control algorithm.

- Modelica: The Modelica language provides the ability to model cyber–physical systems, with connected components governed by mathematical equations. The resulting models are based on first principles and cover a wide range of complex systems that contain electrical, mechanical, thermal, hydraulic, and other relevant components. ORNL has developed an open-source library of nuclear energy relevant components called the Transient Simulation Framework of Reconfigurable Modules (TRANSFORM) using the Modelica language. This library, in conjunction with other Modelica libraries, was selected for the plant-level models. Note that the reactor physics are modeled using reactor point kinetics equations and can be readily embedded within components modeled using the TRANSFORM library. Integrating Modelica models with LabVIEW requires converting the models into a Functional Mock-up Unit (FMU), which uses the Functional Mock-up Interface (FMI) standard for interfacing between multiple dynamic simulation models and is supported by LabVIEW.

- Python or MATLAB: These software platforms provide the necessary support for implementing the MACS framework and algorithms and allow the rapid development and deployment of various algorithms for testing purposes. The selection of these platforms was informed primarily by the availability of commercial-grade support for integrating software developed in these languages with both Modelica and LabVIEW. While the modules being developed in these platforms are intended to be research-focused and not field deployment-ready, the modules may be converted into field-deployable solutions within these platforms, if necessary, in the future.

It is worth noting that both Python and MATLAB provide support for interfacing with FMU modules through various free and paid libraries.

Figure 1 provides an overview of the various modules and the associated communication interfaces that are being developed within MACS. As discussed above, the reactor point kinetics model is integrated within the TRANSFORM-based plant-level reduced order model. This plant-level model provides the necessary information on expected plant behavior. MACS algorithms are expected to use the information from the plant-level surrogate model and, in combination with externally provided setpoints, compute the control element actuation requirements. These actuation requirements (for instance, control drum position, flow rates) can be used as setpoints by the LabVIEW-implemented lower-level PID controllers to physically actuate the hardware. The plant-level surrogate model output may also be used by LabVIEW to drive other hardware components to, for instance, present a visual representation of the reactor power output or temperature.

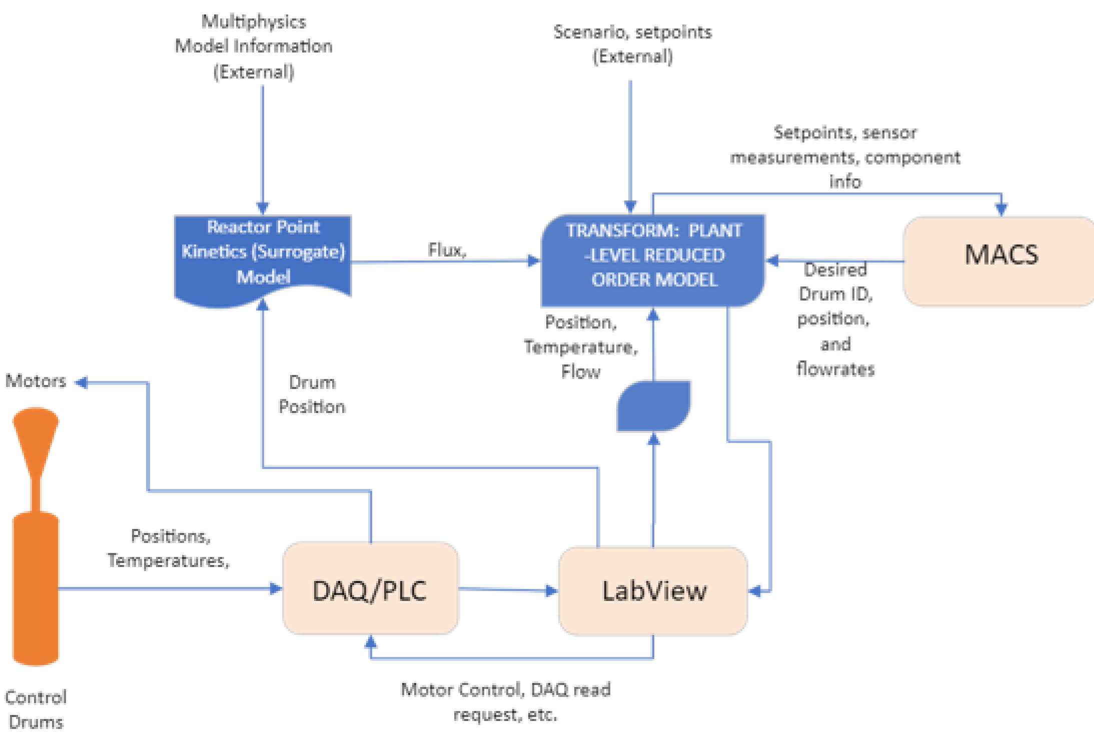


Figure 1. Conceptual interfaces and information flow for MACS.

The rest of this document describes the surrogate model development and its utilization in an initial MACS algorithm.

3. SURROGATE MODEL

3.1 SOFTWARE

The software selected for developing the surrogate models is the Modelica language inside the Dymola developer environment. The ORNL TRANSFORM Modelica library of nuclear energy components [7, 8] was utilized, and the final product is exported as an FMU per the FMI standard.

3.1.1 Modelica

Modelica is a nonproprietary, object-oriented, equation-based programming language used to conveniently model complex physical and cyber–physical systems to account for systems containing mechanical, electrical, electronic, hydraulic, thermal, and control components. A key advantage of Modelica is its separation of physical models and the solvers of the models. This separation enables rapid generation of complex physical system models and control design in a single language without deep knowledge of numeric solvers, code generation, and other factors.

3.1.2 TRANSFORM

A major ORNL initiative has been the development of TRANSFORM inside the Modelica ecosystem [7, 8]. TRANSFORM is an ORNL-developed component library created using the Modelica programming language for the investigation of dynamic thermal hydraulic systems and other multiphysics systems. The TRANSFORM library allows for rapid development of energy systems models, enabling the modeler to customize the components for any application, including instrumentation and control design. The complexity of a large-scale model of an advanced reactor design (in this case a microreactor) can be easily captured inside TRANSFORM.

TRANSFORM allows models to be developed in a modular, open-code manner that can be progressively expanded with increasing complexity as needed. Flexibility and speed are essential to the cost-effective development of next-generation energy systems, especially advanced nuclear power systems. Traditional licensing system codes are not suitable for parameter sweeping, scoping design, or for development of real-time digital twin applications (hardware/software/model-in-the-loop). For this reason, there is a great need for agile system analysis tools for the development of next-generation energy systems that can be integrated into hardware. Therefore, due to its utility for this situation, TRANSFORM was chosen as the base for modeling a microreactor surrogate.

3.1.3 Functional Mock-up Unit (FMU)

The FMU is a model exchange format built to the FMI standard to allow for sharing of system-based dynamic models across platforms. The FMU comprises a zip-format file that contains the system model equations and an XML file with a listing of the input and outputs of the model. In the model exchange format, only the model and listing are provided; the end user must provide a numerical solver (available in National Instruments LabView for this case). Co-simulation is the other format, in which the numerical solver is packaged together with the model, providing a black-box executable. The Dymola software package allows for export of our Modelica/TRANSFORM models in both formats. For compatibility, we are providing model exchange FMUs to be executed inside the LabView environment of the project partners.

3.2 TRANSFORM MODEL FOR MACS

Input parameters were gathered for the TRANSFORM model and consolidated with initial drum reactivity characteristics (drum angle to reactivity conversion) in a TRANSFORM model, which was then converted to an FMU format.

3.2.1 Input for the TRANSFORM Model

Project partners at North Carolina State University provided a nearly complete full model of a reference microreactor built in TRANSFORM. The model is being built with reactor core kinetics values obtained from prior reactor physics calculations; it also includes the key heat transfer geometry, the coolant loops, and will ultimately include a Stirling engine and electrical generation components. Because the model is still a work in progress, this task has extracted only the point kinetics model and a simplified flow loop to pursue an initial surrogate for demonstration of integration with hardware, as shown in Figure 2. The point kinetics parameters are listed in Table 1, the core heat transfer in Table 2, primary heat exchanger in

Table 3, and pump in Table 4 parameters of the Appendix A. The coolant of the loop is sodium, from the TRANSFORM.Media.Fluids library. The pressure for the coolant system was set to 1 bar in the tank for testing purposes.

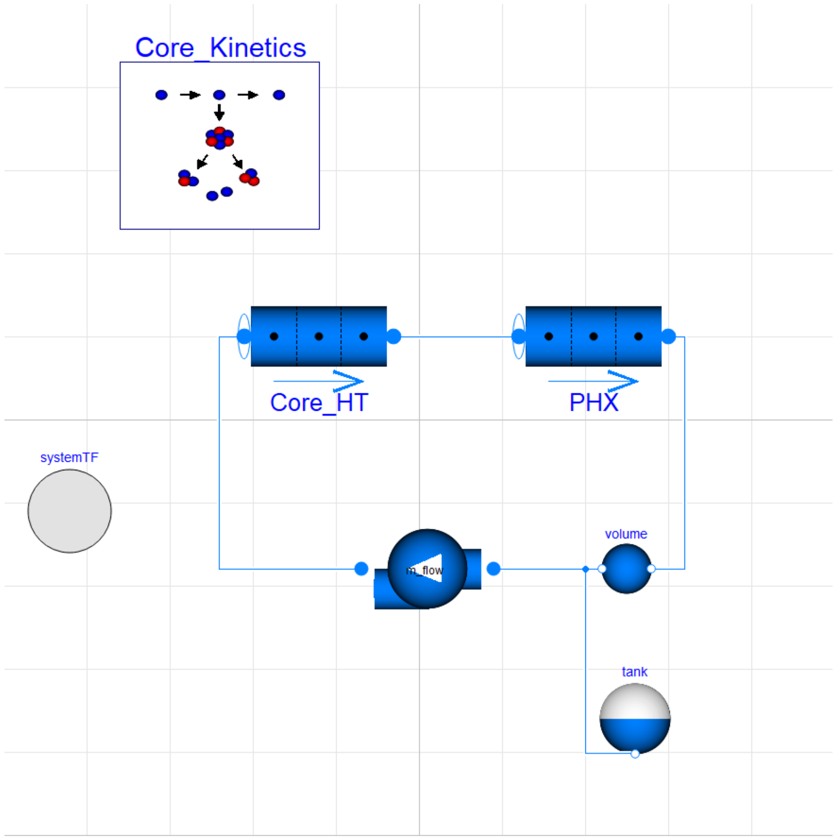


Figure 2. Simplified microreactor model in Modelica. (The Core_Kinetics model contains the reactor core point kinetics equations for calculating the thermal energy production and capturing feedback mechanisms, the resultant heat is captured in the Core_HT model which has the core thermal-hydraulic geometry and calculates the temperature and flow in the core, the flow stream passes through the PHX model which is a generic primary heat exchanger which captures the amount of heat to be dumped to the ultimate heat sink, the volume and tank components add some fluid volume and the ability to expand the coolant without causing numerical issues, finally the m_flow model is the primary coolant pump model for calculating the amount of fluid to circulate in the primary loop.)

3.2.2 Simplified Surrogate TRANSFORM Model

Building on the simplified model shown in Figure 2, inputs were generated for accepting the external reactivity (to be provided by the hardware drum angles in the final demonstration) and a variable thermal load to simulate the Sterling engine, which is driven by the primary heat exchanger. Three output values are created: the measured thermal load power from the primary heat exchanger, the core power, and the core temperature. A diagram of the internals of this model is shown in Figure 3. A proposed application of the FMU as a surrogate is shown in Figure 4. The drum position from the actual hardware can be collected by LabVIEW and routed to the FMU inside the LabVIEW solver environment and produce the changes in reactivity, which can then feed a PID controller (or manual controller), and fed back for the new control drum setpoint. LabVIEW provides the numerical solver to the model exchange format of the FMU. The FMU thus appears in LabVIEW like a block; the inputs and outputs are shown in Figure 3. LabVIEW uses the values from its readings of the drum positioner to provide inputs to the FMU. The

FMU then provides outputs that can be routed to the PID controller either internal to the LabVIEW environment, or to an external controller. So far, the model runs faster than real time. However, as the model becomes more complex, there is a potential need for creating faster running surrogate models.

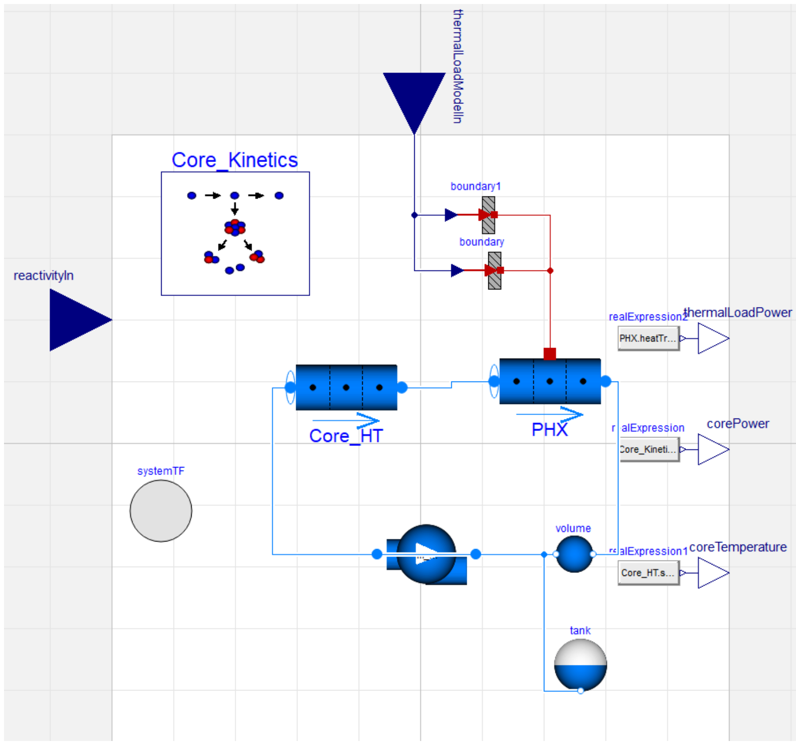


Figure 3. Simplified external controlled model (additional components for capturing an external thermal load is added with the boundary and boundary_1 models, this is connected to an externally provided thermal load input, in the future it could be the Sterling engine. External reactivity, from the control drums, is added as an external input and can be provided by the values calculated from the drum angular position. Three outputs are added to the model to provide the core power, the actual thermal load, and the core temperature for feedback to the external control loop.

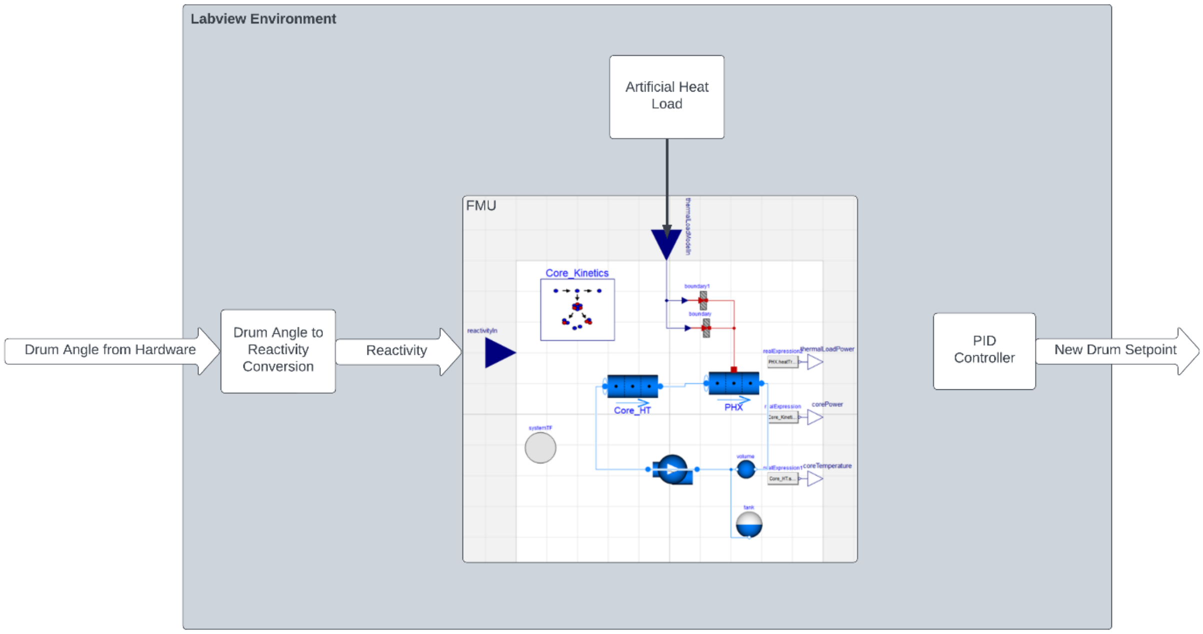


Figure 4. Schematic of the FMU inside the LabVIEW Environment. Drum angle will be provided as an input, it will be converted from angle to external reactivity, this value will be used as an input to the FMU model of the reactor, a response of the core power, temperature and thermal load will be provided to the external PID control, which will calculate the new drum setpoint.

4. TESTING OF THE SURROGATE MODEL

Initial testing of the surrogate model was performed inside the Dymola environment; in Python with FMPy, a Python based library for running FMUs; and also as an FMU inside of MATLAB/Simulink and the LabVIEW environments.

4.1 TESTING IN DYMOLA

Initial testing of the surrogate model was performed inside the Dymola environment using the native DASSL, a differential/algebraic system solver package [9]. Initial values of −0.85e5 W and −0.025 were chosen for the thermal load (to balance the core initial power of 0.85e5 W) and external reactivity to show an open loop holding near steady state power for a 1,000 s duration test. For a nuclear reactor to operate, its overall reactivity must be greater than one, so the core is designed to have a reactivity greater than one but is controlled by, in this case, external reactivity of less than one. Effectively, neutrons are allowed to leak to maintain a steady criticality. The resultant power as a function of time is shown in Figure 5 along with the core reactivity. The delay of the power behind the reactivity is shown in the inset during the initial transient period. This is a demonstration of the feedback dynamics of the kinetics equations. The fluctuation due to the temperature feedback and Xenon buildup, shown in Figure 6, can be seen in the initial transient of the test case. The initial cooldown of the system causes a reactivity spike, shown in the inset of Figure 6, which increases the overall power. The power then comes down due to Xenon buildup and also the negative external reactivity being inserted. This power settles to a steady state after the reactor approaches the approximately steady operating temperature. This demonstration shows that the Dymola implementation is functional and stable and gives appropriate results.

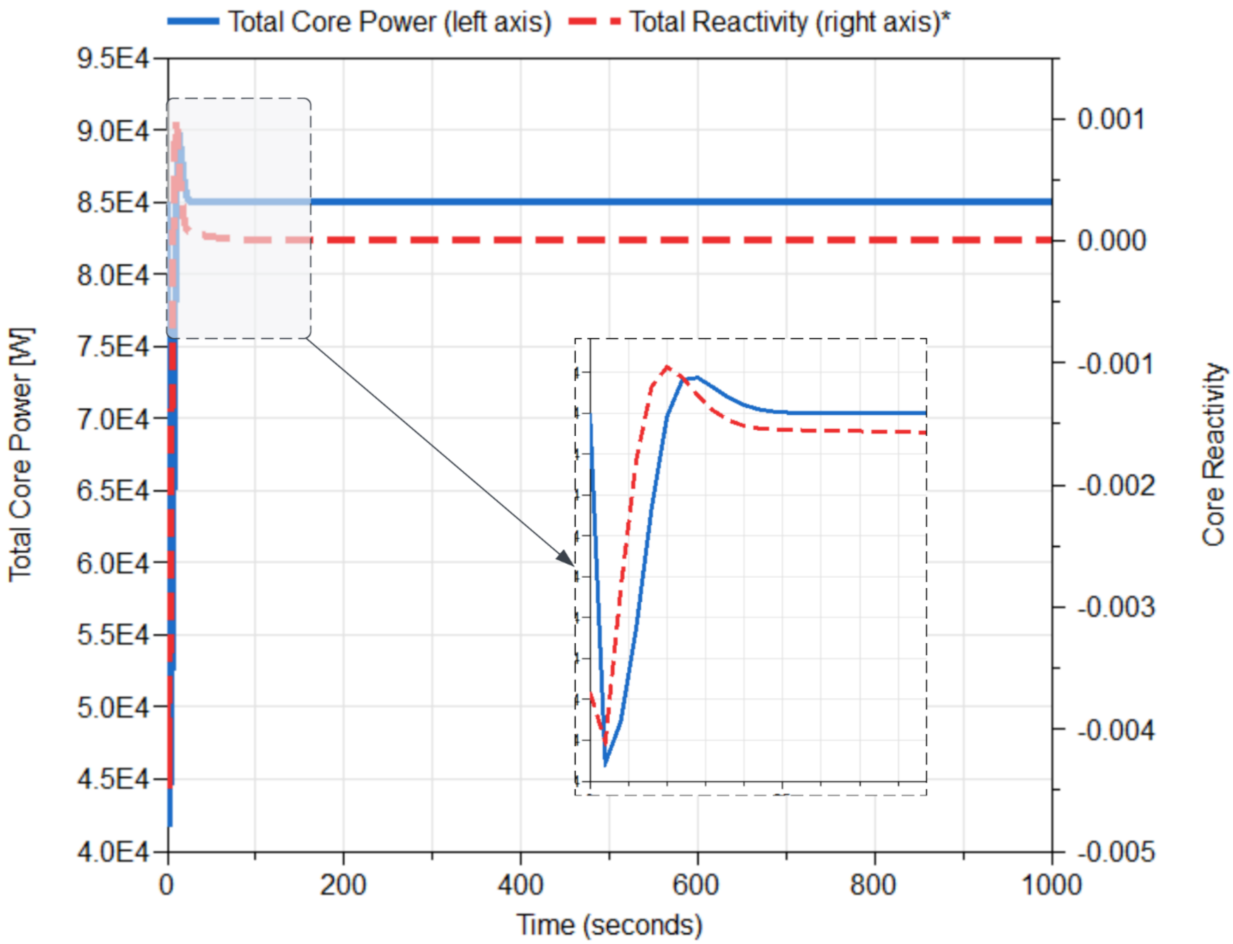


Figure 5. Steady state power test of surrogate model (inset shows the time lag between total core power and the total reactivity due to the kinetic feedbacks).

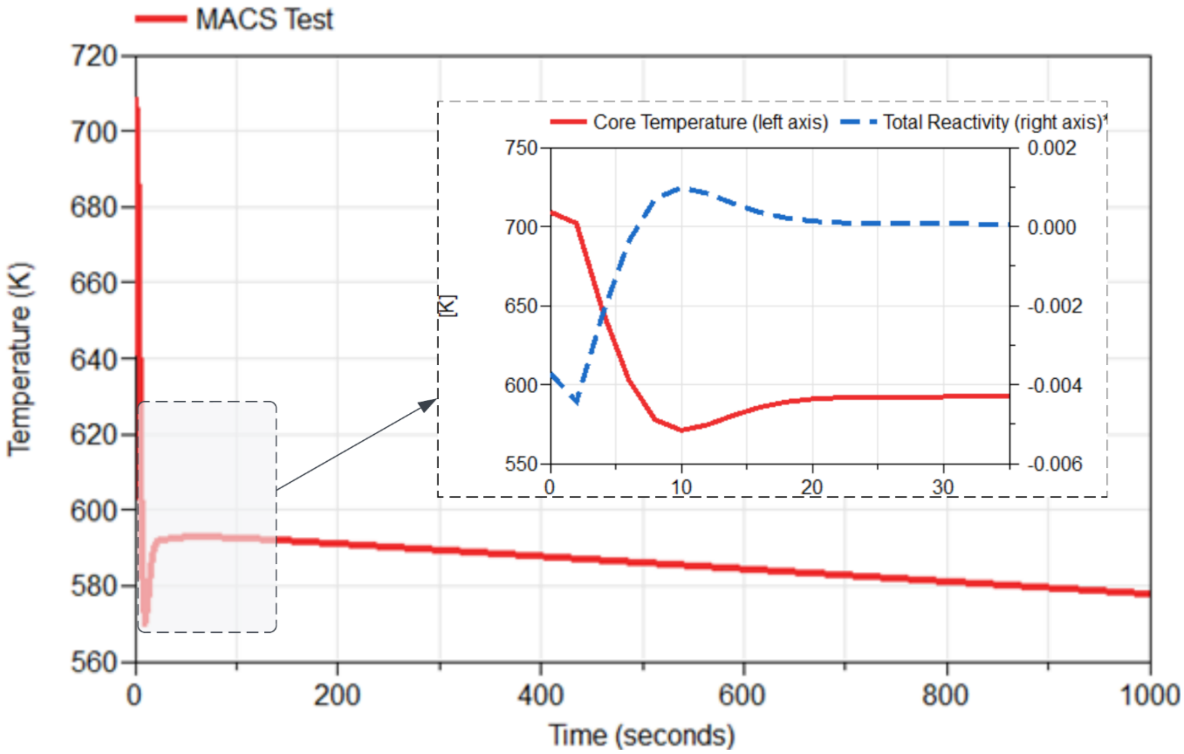


Figure 6. Core average temperature from steady state power test of surrogate model (inset shows temperature feedback effects on the total reactivity of the core).

4.2 TESTING WITH PYTHON FMPY

The Python FMPy environment is a library for running FMUs and is a good way to test the basic functionality of models. FMPy includes a GUI interface for quick viewing of simple simulations. A simulation was performed using the above parameters as a test. The results of the reactivity (Core_Kinetics.rho), core power, and core temperature are shown in Figure 7. The results match the Dymola results well: point checks of errors showed errors of less than 1% on the core power and temperature. The FMPy GUI allows for the simulation model to be exported to a more convenient Jupyter notebook for further testing, like the sensitivity analysis shown in the following section. The result is a successful demonstration of the FMU and using FMPy for the simulation.

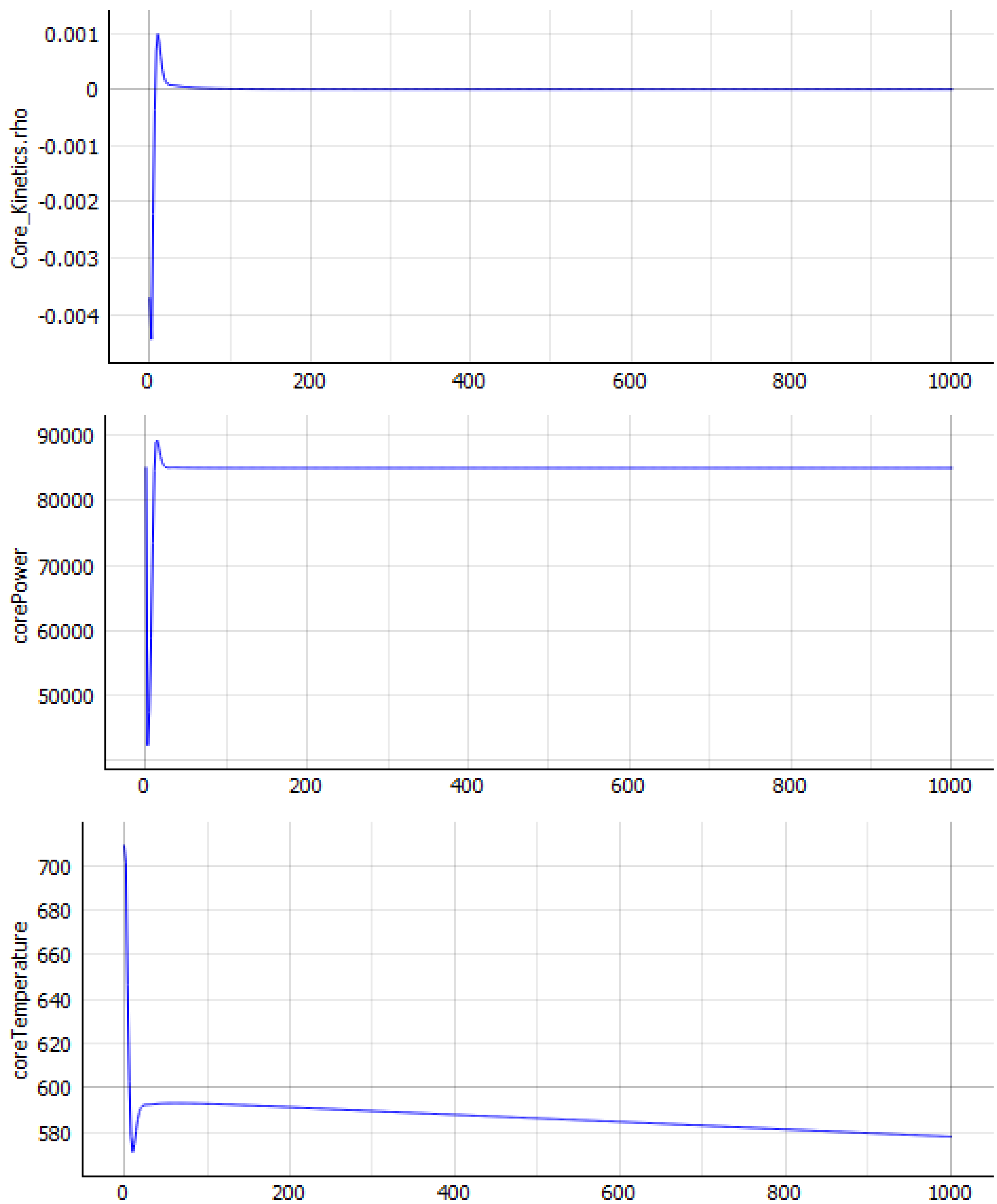


Figure 7. Python FMPy Results for steady power test.

4.3 TESTING IN MATLAB/SIMULINK

Further testing was performed in the MATLAB and Simulink environments, using the identical parameters from the Dymola tests using the FMU. First, the model exchange format was used in MATLAB: the same results were obtained as those for the CVODE or Runge–Kutta solvers, as shown in Figure 8. A complementary run was created using the co-simulate modality in Simulink; in this run, the Dymola solvers were included in the FMU. The Simulink model diagram is shown in Figure 9. The resultant plot of the Simulink results is shown in Figure 10. Point checks of the results show that the core power is less than 1% error from the original Dymola model. This is considered a successful demonstration of the FMU inside the MATLAB and Simulink environments.

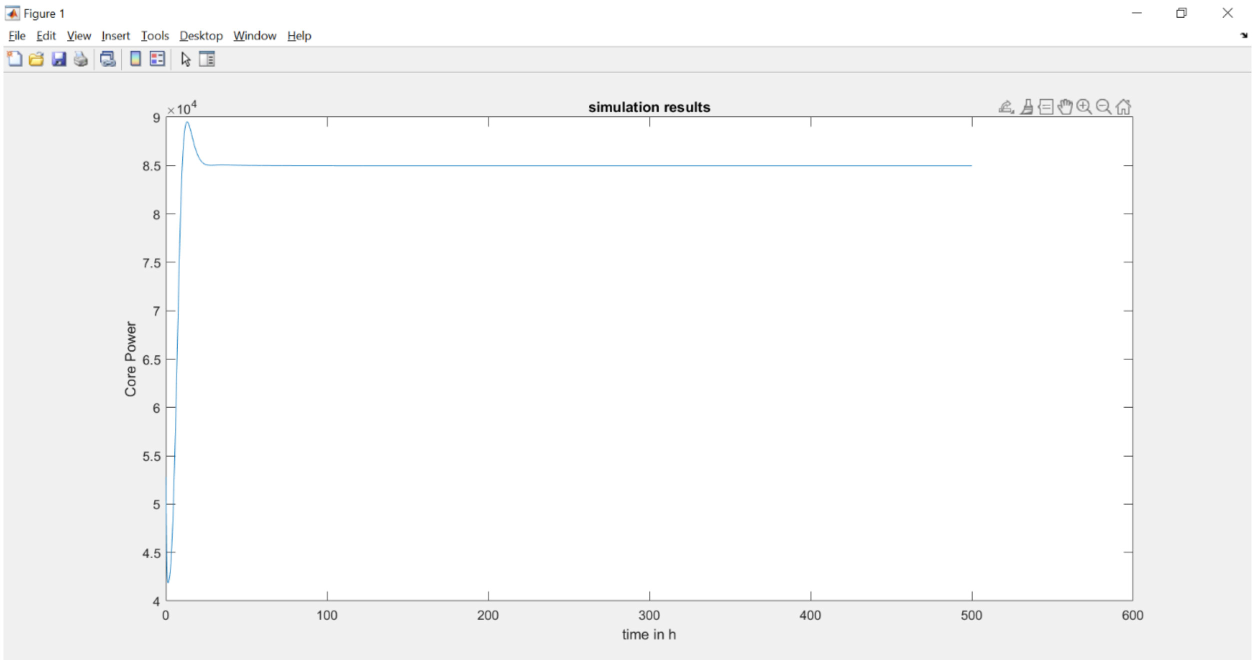


Figure 8. MATLAB model exchange power result for a 500 second simulation.

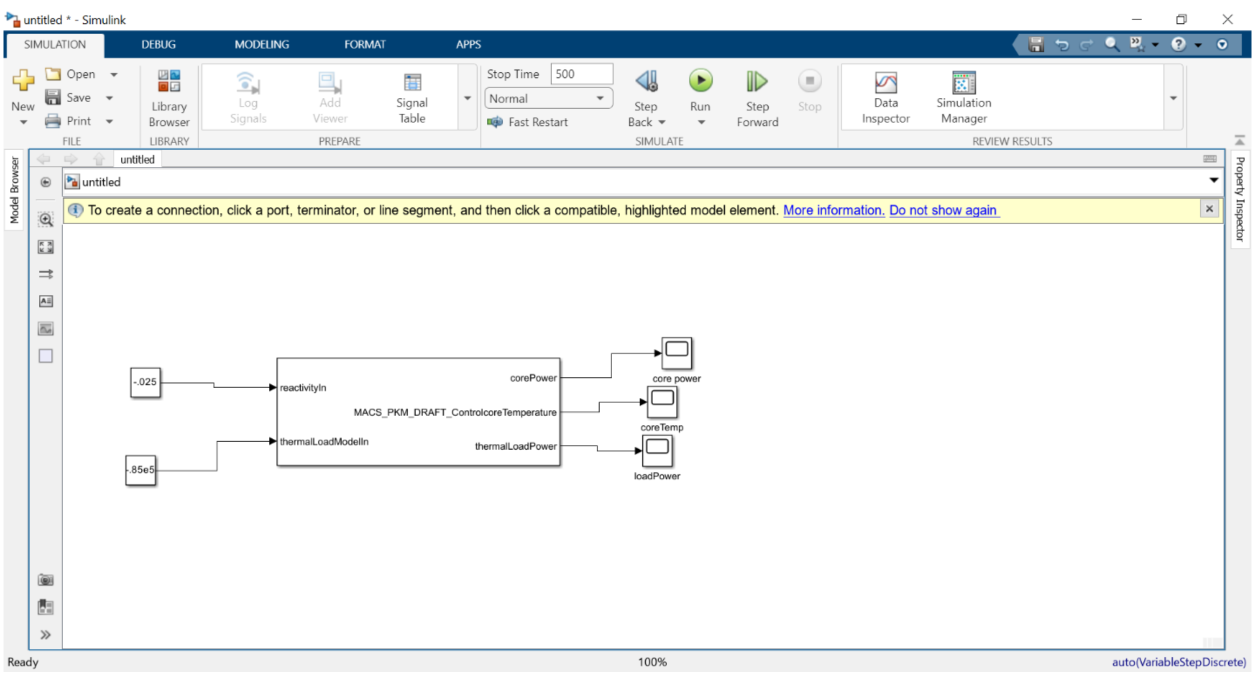


Figure 9. Simulink test model using the co-simulate FMU.

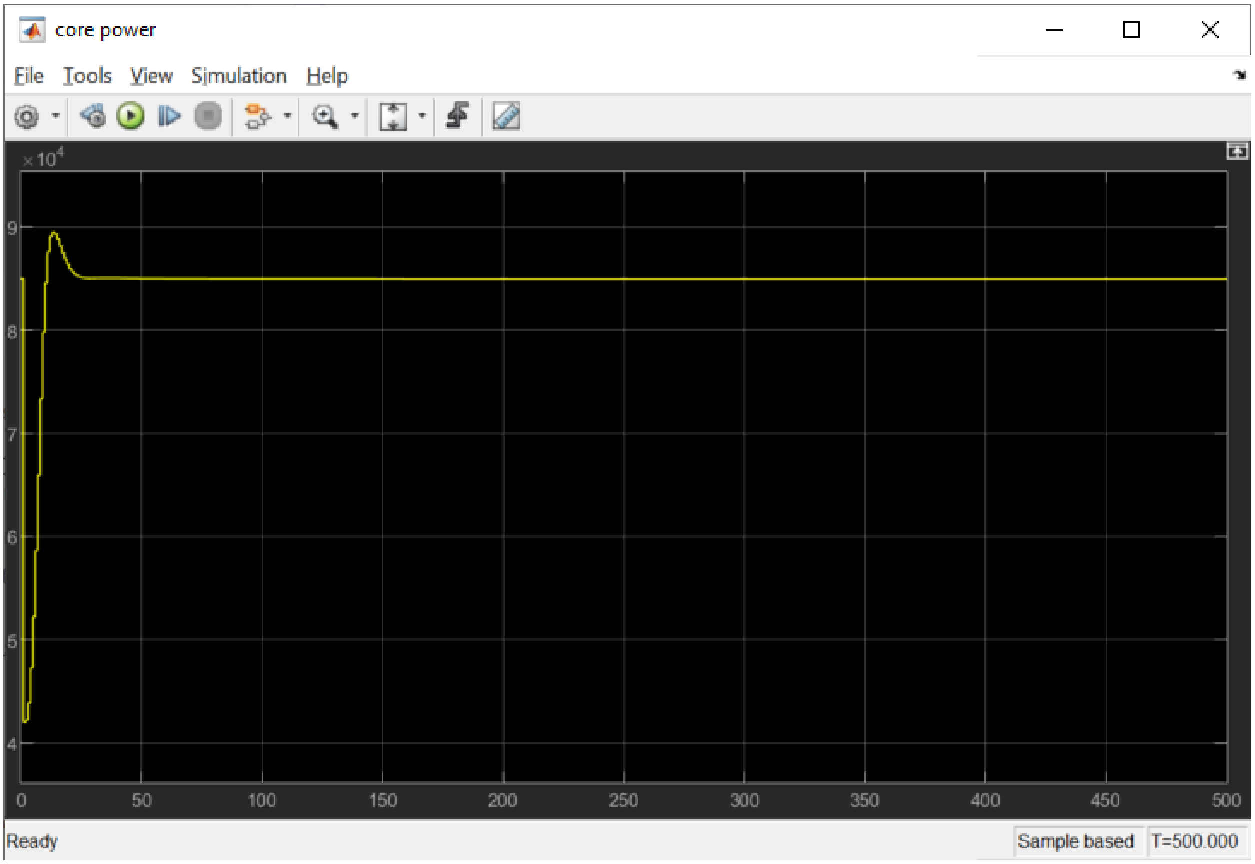


Figure 10. Results of co-simulate FMU test in Simulink.

4.4 IN THE NI LABVIEW ENVIRONMENT

Further testing was done in the NI LabVIEW environment at INL. LabVIEW only accepts the model exchange FMUs and does not accept the combination model exchange and co-simulation FMUs containing both methods. Care must also be taken in selection of 32-bit vs 64-bit FMUs when exporting to match the hardware's operating system. The model exchange FMU was run with a Runge-Kutta 23 solver and was reported to match the above outcomes from the other systems to within 1% error. This was considered to be a successful demonstration of the FMU in the NI LabVIEW environment.

5. CONTROL AUTOMATION AND INTEGRATION

The surrogate model was employed in preliminary control automation studies and as input to a hardware in the loop demonstration, with a focus on a load-following scenario. Two basic types of control algorithms were implemented—a PID controller and a model-predictive controller (MPC)—though the initial focus was on development of the overall framework and integration with the hardware testbed at INL and not necessarily on evaluating the accuracy and robustness of the control algorithms. This focus was primarily due to the need to ensure accuracy and robustness of the surrogate models prior to evaluating the accuracy of the control algorithms themselves.

Therefore, before to testing the controllability, it was desired to see the sensitivity of the surrogate model output to variations of the input to the surrogate model. Sensitivity analysis was performed for the system in both the Dymola environment as parametric studies and using the FMPy FMU in Python.

5.1 PARAMETRIC AND SENSITIVITY STUDIES

Dymola's parametric capabilities were used to test the open loop to ascertain the effects of different static external reactivity values. Several negative values were used up to zero external reactivity. The resulting effect on core power is shown in Figure 11.

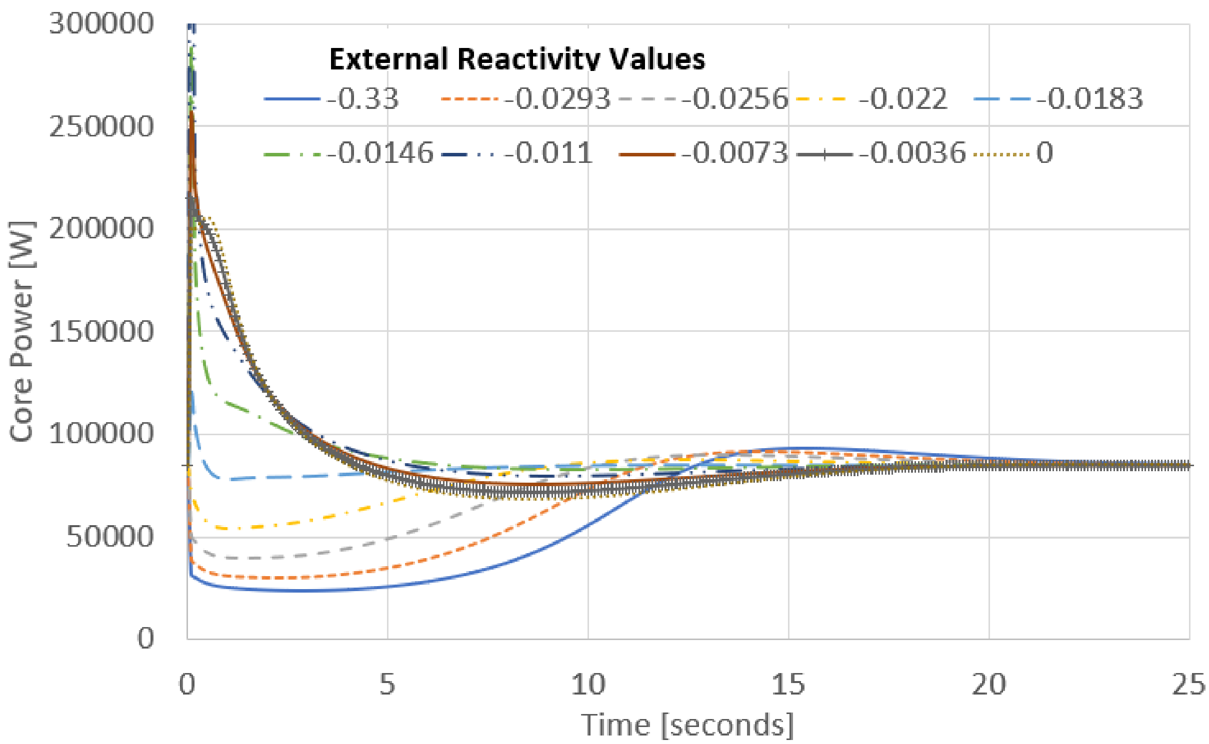


Figure 11. Parametric study of external reactivity on core power.

The time-averaged sensitivity (variation in output from nominal output) of core power due to variation in input to all the variables was performed. The inputs were varied by +/- 5%. The results are presented in Figure 12 and Figure 13, time plots of a couple of the more sensitive values (precursor fraction – alpha and fission yield) are presented. The Core_Kinetics, the TRANSFORM point kinetics model used the

6-group precursor. The core power is most sensitive to the normalized precursor fractions, delayed neutron fraction, neutrons per fission, and slightly less to fission product yields.

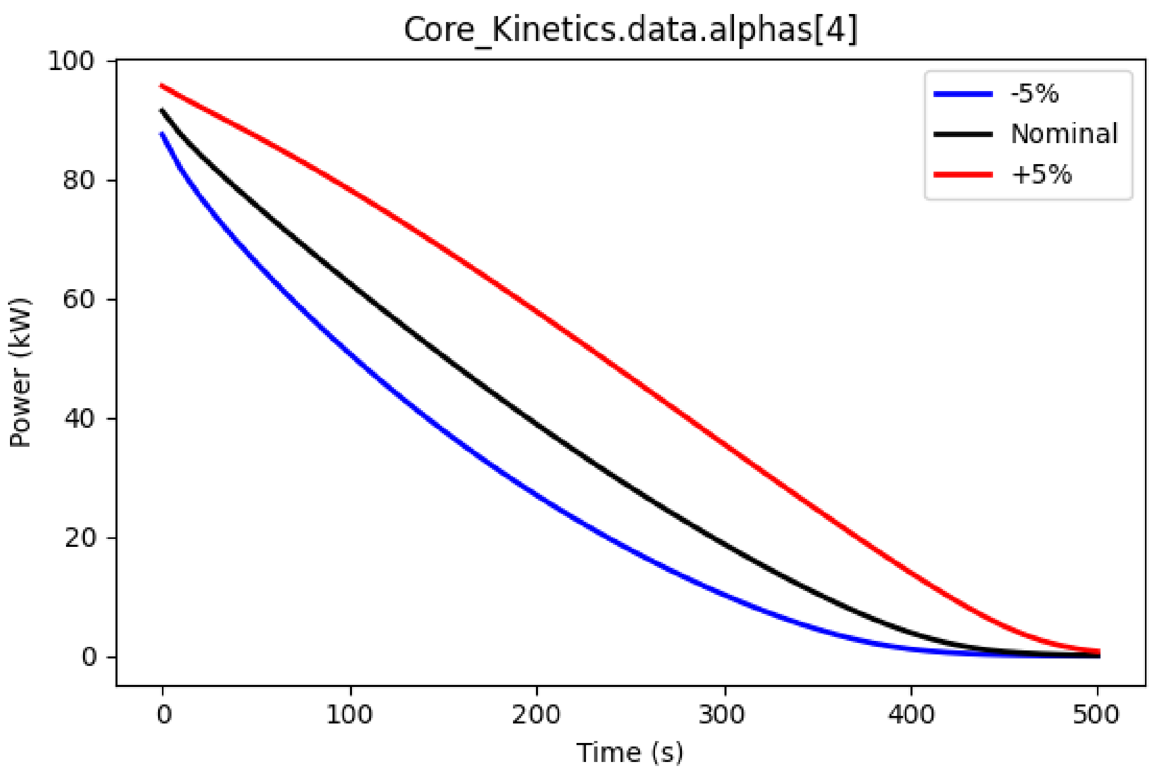


Figure 12. Core power sensitivity to precursor fraction (alpha).

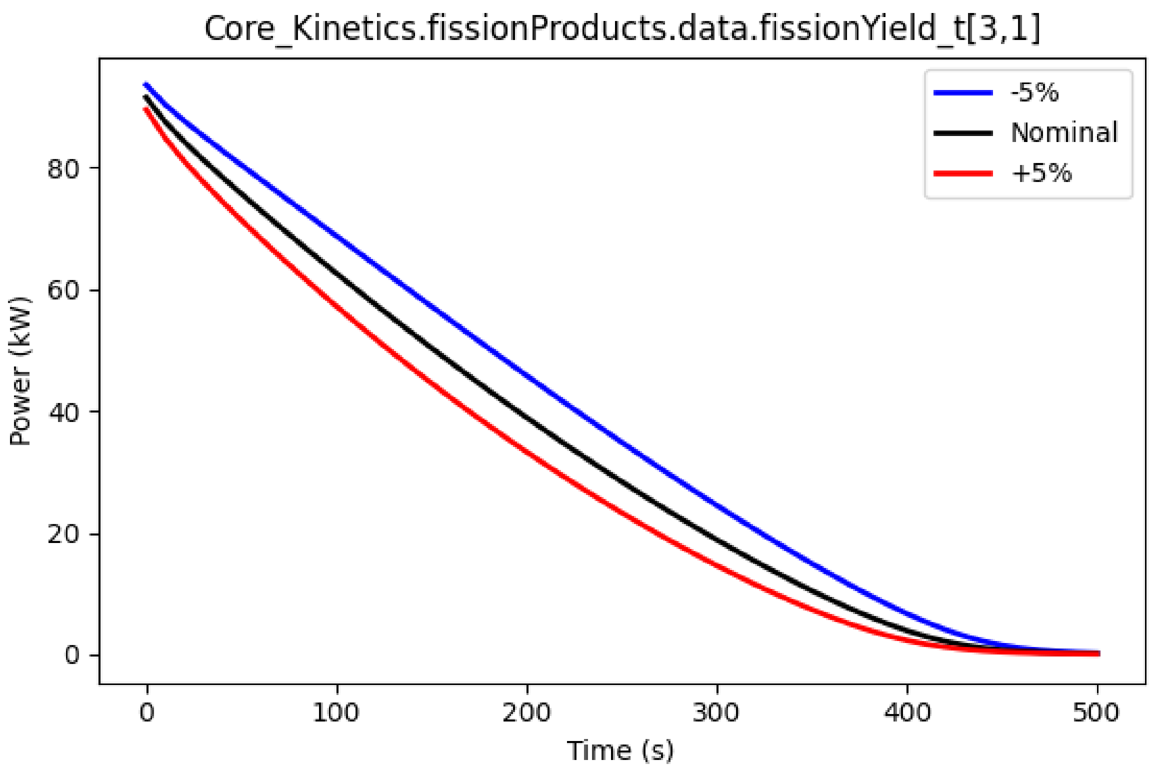


Figure 13. Core power sensitivity to fission yield.

The sensitivity analyses in the previous sections indicate a surrogate model that appears to be reasonably stable over a limited range of reactivities. This is likely due to an inherent limitation of the point kinetics model, where the available model parameters were established for steady-state full power operation. As a result, the model behavior may not be well-defined when deviating significantly from this parameter set, and indeed the sensitivity analysis indicates higher sensitivity to certain parameters that may change significantly based on the startup sequence and operational history. As a result, the available surrogate model was deemed to be sufficient for demonstration of the overall MACS framework, and a comprehensive evaluation of automated control algorithms themselves will require a more complete surrogate model or multiple surrogate models covering the range from startup to full power. Future work must focus on refining the parameters used for the desired microreactor's specific point kinetics parameters; moreover, sufficient information about the core geometry and flow loop geometry must be integrated.

5.2 AUTOMATED CONTROL

5.2.1 External Reactivity Insertion Model

To "demonstrate" the control of the hardware elements, the desired external reactivity insertion must be converted into a hardware drum setpoint. Assuming a control drum is used for reactivity control, the external reactivity insertion by the control drum must be computed as a function of the shaft angle using a function such as that described in Upadhyaya et al. [10] and summarized in the equation below:

$$\rho_d = 6.89 \times 10^{-13}\theta^5 - 2.33 \times 10^{-10}\theta^4 + 3.28 \times 10^{-9}\theta^3 + 4.57 \times 10^{-6}\theta^2 - 5.88 \times 10^{-5}\theta$$

where θ is the shaft angle of the stepper motor driving the control drum and ρ_d is the reactivity worth due to the control drum corresponding to the drum position angle. In the absence of additional information about the specifics of the external reactivity insertion from the control drums in the reference microreactor selected for the initial development and integration of MACS, it was decided to use this relationship for the initial effort and use this function (or its inverse) in the HIL demonstrations.

The sensitivity analysis points to a small range of external reactivity insertions and a corresponding small range of control drum positions for which the surrogate model provides stable outputs. This small range of external reactivities may indicate a potential shortcoming of the surrogate model (either the point kinetics model or the plant model or both) to represent the full range of reactor power levels that will need to be addressed in future work. Note that this issue with the surrogate model is specific to this reference microreactor concept; other microreactor concepts are expected to utilize design-specific models that may overcome this limitation of this surrogate model.

5.2.2 Control Automation

For the purposes of initial demonstration, the focus of the control automation was on demonstrating the ability to adjust the control elements (control drum) to meet a variable load (i.e., load following) scenario. Two basic control algorithms were investigated: a PID controller and an MPC. MPCs offer options for selecting control strategies based on look-ahead simulations and the inclusion of constraints such as reactivity insertion rate. In principle, other constraints—such as those imposed by limited availability of certain components—can also be incorporated though testing this functionality will require additional modifications to both the surrogate models and any hardware components in an HIL simulator. As a result, the initial focus of the effort was on developing the methods and integrating with the control drum hardware testbed at INL.

Figure 14 shows a basic load-following setup using the surrogate models, with setpoints for reactor power varying sinusoidally in a small region about the steady state operating point for the microreactor. The reactor power level is controlled through a simple PID controller that adjusts the control element (i.e., control drum) position, thereby inserting or removing excess reactivity. The example in Figure 15 demonstrates the ability to follow a sinusoidal demand.

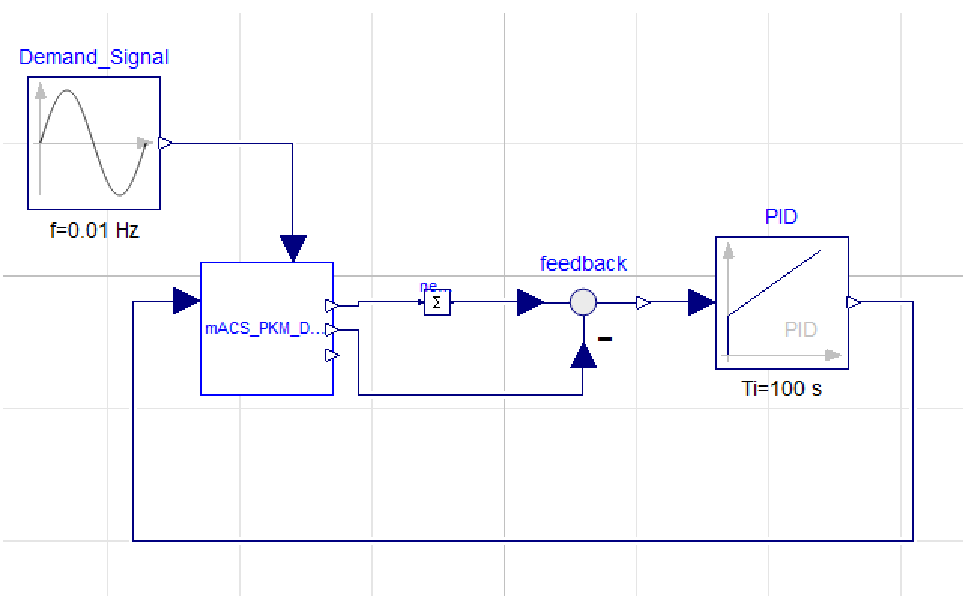


Figure 14. Schematic of MACS FMU model inside a simple PID control loop with a sinusoidal load demand.

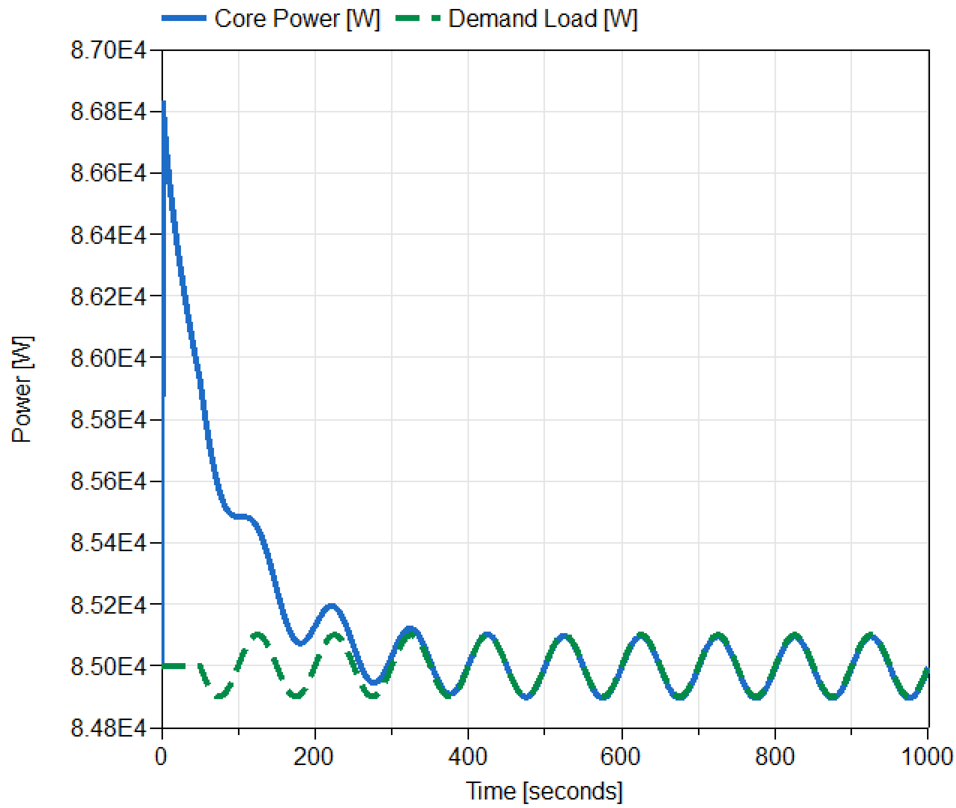
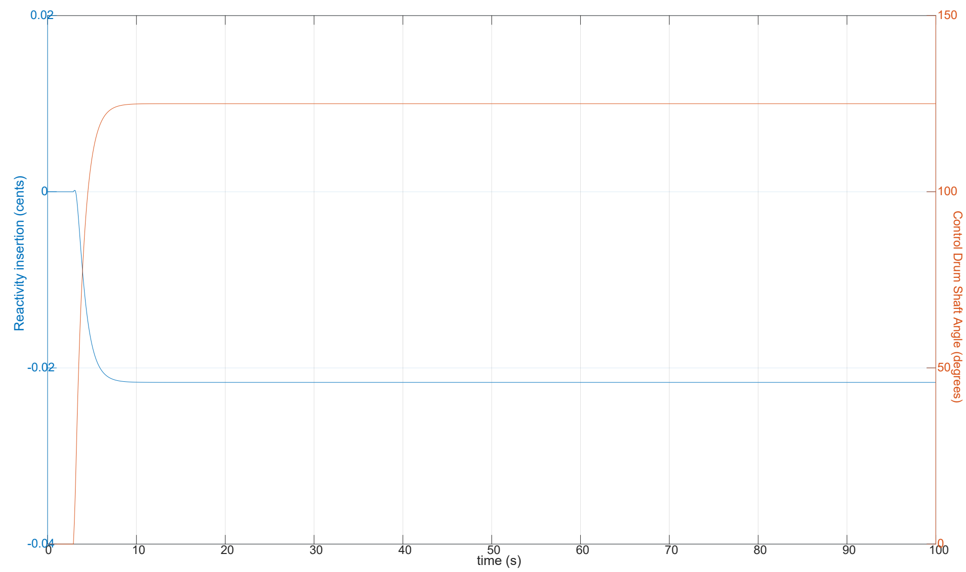


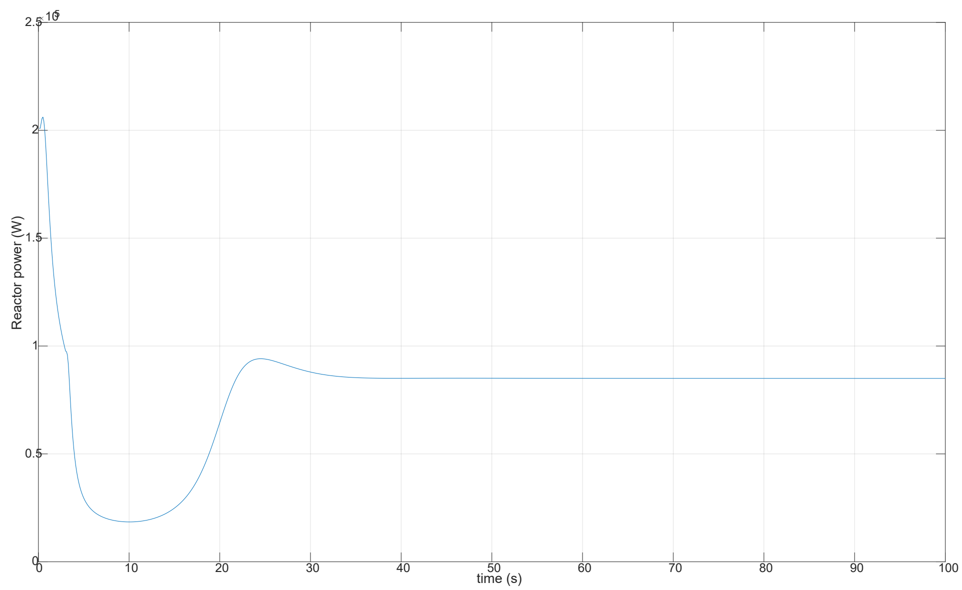
Figure 15. Example of reactivity control to achieve load following.

In addition to the PID controller, an MPC was also implemented. MPC controllers are an advanced control technique that compute an optimal control action through predicting future behavior of a dynamical system using a model [11] and solving a constrained optimization problem. The prediction uses measurements or estimates of the state of the system and hypothetical future inputs or control policies. Only the first control input of the computed optimal control action is used, and new inputs at the next time step are used to repeat the process. MPC controllers provide a level of robustness to uncertainties in the model parameters and are increasingly preferred for applications in which a certain level of automation and robustness is needed.

As indicated above, MPC controllers require a dynamical system model for prediction of future states. In the present application, a basic system model was derived using simulation data generated using the Modelica models that were exercised for the range of external reactivity insertion values described in the sensitivity analyses. The resulting approximate system dynamics model was used, with constraints on the amount and rate of external reactivity insertion, to drive the plant to desired operating states defined by reference values (setpoints) of the reactor power and temperature. Figure 16 shows an example of the external reactivity insertion and drum setpoints as computed by the MPC for a reference reactor power sequence.



(a)



(b)

Figure 16. Example of MPC controller: (a) computed drum position and associated external reactivity insertion, (b) associated reactor power computed by Modelica.

5.2.3 HIL Integration

The software modules were designed and developed for the specific purpose of integration with the control drum testbed hardware at INL. The surrogate model software interface had, as inputs, the thermal load demand and the external reactivity insertion variables in this initial stage of the research. These inputs act as setpoints and measured data for MACS. Outputs from the model included the reactor power, core temperature, and the thermal load output. The surrogate model contains other parameters such as the coolant flow and temperature that can be made available for external access as needed in future iterations of MACS.

Similarly, the control algorithm modules were developed to initially take as input the externally supplied demand signal, the control drum position, drum limit switch locations, and "measurements" of reactor core temperature, power, and thermal output from the plant. Additional parameters, such as limits on the drum rotation speed, were hard-coded in this initial stage of development, though these are available to be readily exposed via the interface with LabVIEW in the future.

Note that the control automation modules at present assume a fully functional reactor, control drum, or other components. Degradation of one or more of these components is assumed to be measurable (in the case of hardware components) or can be represented by a change in the functionality of the component (i.e., components in the plant surrogate model). In the case of components represented in software (i.e., in the surrogate model), MACS assumes that the model is sufficiently accurate as to provide the necessary inputs (reactor and plant outputs) for control decision-making. MACS also includes a basic set of parameters that account for hardware actuator functionality, though these parameters are not currently estimated from measurements.

The current version of the modules also assumes that the drum position setpoints computed by MACS will be applied to all control drums, and the switching process between multiple drums is handled by LabVIEW. Although the MACS control module contains the code base to compute and recommend different positions for each control drum, the surrogate models must be modified to explicitly account for such differences; such work will be a part of the next phase of research.

The integration of the surrogate plant model and automated control algorithms was performed by INL as part of a separate milestone effort within this research program (M2AT-23IN0804054). The integration pointed to the potential for an HIL simulator that integrates hardware elements with a surrogate model and control algorithms to enable a MACS framework.[1] The effort, though successful insofar as hardware control in an HIL setting was demonstrated, also pointed to a few issues that must be resolved before the MACS HIL framework can be opened up for stakeholder access.

- Software platform interoperability. Although the various software platforms can theoretically interface, such interfacing in practice entails nuances that must be addressed. For instance, the existing LabVIEW tools supporting FMU integration require that the FMU be in Model Exchange format. The implementations of these tools also appear to require the user to exercise care to ensure that FMU instances that are started up are properly shutdown in the event of any errors that occur in LabVIEW during execution. Likewise, care appears to be needed in the implementations in LabVIEW and MATLAB to ensure that variable redefinitions in one or both of these tools are properly accounted for.

- Timing and delays. The incorporation of hardware and safety interlocks introduces delays in the feedback loops. These delays and timing shifts must be accounted for in the surrogate models. Tests indicated that the surrogate model dynamics have a short time constant (i.e., shorter than the estimated hardware loop timing delays), resulting in instabilities in the feedback control loops. Although some level of delays can be hard-coded into the surrogate models and control logic (such as limiting the rate at which the drum position setpoint can change; Figure 16), a robust demonstration

[1] Personal Communication, Anthony Crawford (Idaho National Laboratory), June 2023.

requires that the surrogate models incorporate the correct level of inertia to accurately represent the reactor and plant dynamics, and the HIL components reflect these requirements as well.

These issues are the focus of ongoing development and testing efforts at the collaborating laboratories and demonstrate the importance of these research efforts in finding the difficulties in integration of software and hardware for automated systems.

6. CONCLUSION

A simplified surrogate model for a microreactor has been created in the Modelica TRANSFORM library and converted to an FMU for integration as HIL. The FMU accepts as input variables the external reactivity and the ultimate thermal load. The external reactivity can be connected to an external drum angle value signal and calculate the reactivity to be input into the point kinetics model. The point kinetics model, in turn, will calculate the core thermal power, which is connected to a thermal hydraulic loop model connected to an ultimate heat sink to simulate a demand load. The FMU was tested under several modalities and was demonstrated to work in the NI LabVIEW environment, to which it will be deployed in its final state. Tests were performed in the software environment with simple PID control for demand load following and model predictive control for meeting a fixed demand load. Finally, documented discussion points of the integration with INL hardware are presented, showing the key issues with continuing efforts for final hardware integration of the model.

This work was successful in meeting the goals for the milestone to produce a functioning model and export it to an FMU format capable of being integrated into the INL test facility. The stability of the FMU under varied inputs demonstrated that it is a robust methodology. Although the surrogate was simplified to capture the basics of a microreactor, the point kinetics feedbacks are demonstrated to function well. As more refined microreactor models are created in the Modelica environment, it will be easy to follow the same workflow to produce new FMUs that can be integrated into the hardware. Future work will look for improved microreactor models from partners at NC State University. The models can also be interlinked with future hardware. Modelica can model PID controls from inside the FMU or outside of the FMU; these capabilities will be demonstrated as software in the loop when the hardware comes together in the next phases of the project.

7. REFERENCES

1. *Roadmap for the Deployment of Micro-Reactors for U.S. Department of Defense Domestic Installations*. 2018, Nuclear Energy Institute.
2. Peakman, A., Z. Hodgson, and B. Merk, *Advanced micro-reactor concepts.* Progress in Nuclear Energy, 2018. **107**: p. 61-70.
3. *Final Environmental Assessment for the Microreactor Applications Research, Validation, and Evaluation (MARVEL) Project at Idaho National Laboratory*. 2021.
4. Waksman, J., *Project Pele Overview: Mobile Nuclear Power For Future DoD Needs*. 2022.
5. J. M. O'Hara and S. Flegler, *Human-System Interface Design Review Guidelines (Revision 3)*, U. NRC, Editor. 2020.
6. J. M. O'Hara, et al., *Human Factors Engineering Program Review Model (Revision 3)*, U. NRC, Editor. 2012.
7. Greenwood, M.S., *TRANSFORM - TRANsient Simulation Framework of Reconfigurable Models* 2017.
8. Fugate, D.L., et al., *Update on ORNL TRANSFORM Tool: Simulating Multi-Module Advanced Reactor with End-to-End I&C*. 2015, Oak Ridge National Laboratory.
9. Petzold, L.R., *Description of DASSL: a differential/algebraic system solver*. 1982: United States.
10. Upadhyaya, B.R., et al., *Autonomous Control of Space Reactor Systems*. 2007, University of Tennessee, Knoxville: Knoxville, TN.
11. Kouvaritakis, B. and M. Cannon, *Model Predictive Control: Classical, Robust and Stochastic*. Advanced Textbooks in Control and Signal Processing. 2016: Springer Cham.

APPENDIX A. INPUT DATA FOR TRANSFORM MODEL

APPENDIX A. INPUT DATA FOR TRANSFORM MODEL

Table 1. Point Kinetics Parameters in the TRANSFORM Model.

Parameter	Value	Unit
Total Nominal Power	0.85e5	W
Volume for fission product concentration basis	0.0025588	m^3
Number of reactivity feedbacks	1	
Reactivity feedback coefficient	-4.28e-5	1/K
Reference temperature for reactivity feedback	600	K
Prompt neutron generation time	3.6e-5	sec
Energy released per fission	2.46538	
Macroscopic fission cross-section of fissile material	0.00014875	
Initial reactor fission power	0.85e5	W
Normalized precursor fractions (Alphas)	0.0346, 0.1798, 0.1704, 0.3848, 0.1625, 0.0679	
Effective delayed neutron fraction (Beta)	7.48721e-03	
Decay constants for each precursor group (Lambdas)	0.0133, 0.0327, 0.1209, 0.3036, 0.8531, 2.865	1/sec
Fission products	Te-135, I-135, Xe-135	

Table 2. Core Heat Transfer Parameters in the TRANSFORM Model.

Parameter	Value	Unit
Number of parallel volumes	4	
Number of volume nodes	2	
Hydraulic diameter	0.0254	m
Length	1.0	m
Initial Temperature	436	C
Initial Flow Rate	1.25	kg/s
Internal Heat Generation per volume	Q_total from Point Kinetics model	W

Table 3. Primary Heat Exchanger Parameters in the TRANSFORM Model.

Parameter	Value	Unit
Number of parallel volumes	1	
Number of volume nodes	2	
Hydraulic diameter	0.0254	m
Length	1.0	m
Initial Flow Rate	1.25	kg/s
Internal Heat Generation per volume	-1e5	W

Table 4. Pump Parameters in the TRANSFORM Model.

Parameter	Value	Unit
Nominal Flow Rate	1.25	kg/s