COLLEGE OF ENGINEERING
NUCLEAR ENGINEERING & RADIOLOGICAL SCIENCES
UNIVERSITY OF MICHIGAN

Nuclear Reactor
Analysis & Methods
Group
NuRAM

NURAM-2023-006-00

High Fidelity Transient Simulations of the Multi-module HTGR Special Purpose Reactor

Sooyoung Choi[1], Qicang Shen[1], Chang-ho Lee[2], and Brendan Kochunas[1]

[1]*University of Michigan*
[2]*Argonne National Laboratory*

09/14/2023

This page is intentionally blank.

REVISION LOG

Revision	Date	Affected Pages	Revision Description
0	09/14/2023	All	Initial Release

Document pages that are:

Export Controlled:	None
IP/Proprietary/NDA Controlled:	None
Sensitive Controlled:	None
Unlimited:	All

EXECUTIVE SUMMARY

In this report we present the methodology and sensitivities of a novel simulation capability that couples state-of-the-art control algorithms with high-fidelity reactor physics simulation tools.

The summary of the contributions of this report are:

- Development of a novel simulation capability that combines state-of-the-art control algorithms with high-fidelity reactor physics simulations
- 3D multiphysics transient load follow simulations of an High-Temperature Gas-Cooled Reactor (HTGR) microreactor (Fig. EC.1)
- Detailed sensitivity analysis of parameters affecting controller performance

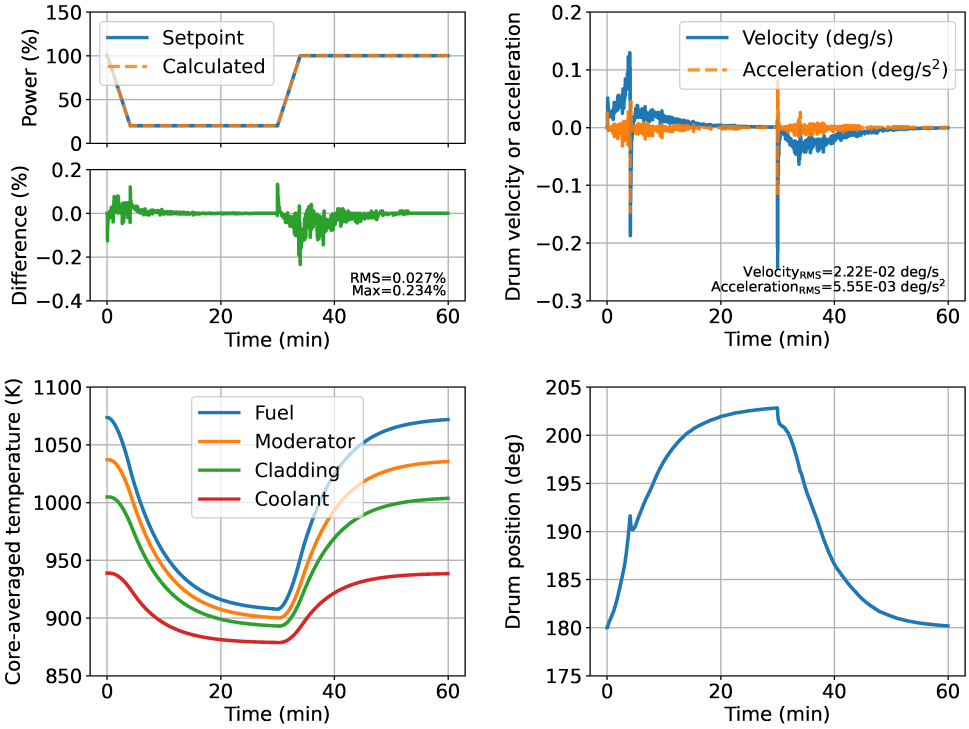


Figure EC.1. 3D microreactor load-follow simulation results

This report provides a review our representative HTGR-like microreactor, and previously developed Model Predictive Control (MPC) based autonomous control algorithm for the reactivity control system. We then give an overview of the high-fidelity simulation techniques and the developments necessary to perform load-follow transient simulations. A detailed description of the reduced order reactor model and corresponding state-space model for the controller then follows. The MPC formulation and closed loop controller structure is described where *adaptive* MPC is used to capture the nonlinear effects inherent in the point reactor model, and observer is developed to enable the coupling between the controller and high-fidelity simulation tool. An "off the shelf" optimization

library for Operator Splitting solver for Quadratic Program (OSQP) is used to solve the control optimization problem.

Through several numerical experiments, we observe that the MPC and point reactor state-space model provide excellent control of the reactivity control system where tracking errors between the core power and set point are within 0.234%, with control inputs remaining within constraints. Transient load-follow results from a 3D high-fidelity multiphysics reactor simulation coupled with the MPC controller to calculate rod position are shown in Fig. EC.1.

Extensive sensitivity tests are also performed to gain insights into the performance of the controller in terms of some of its tuning parameters, and accuracy of key reactor physics quantities needed in the state-space model. Overall, the sensitivities of the parameters evaluated revealed that the controller is quite robust. Most parameters would need to differ by more than an order of magnitude to significantly degrade the controller performance. The two largest sensitivities identified are the accuracy of the constraints applied to the controller, and accuracy of the control drum worth curves. A reduction of the constraints to 10% of their nominal value increased the maximum tracking error to 8%, while having control drum reactivity worths off by ±60% demonstrated the controller can become saturated and provide highly oscillatory control inputs (although it was still able to provide an accurate control solution).

Typical technical specifications for zero power physics testing and measurement of reactivity worth require significantly tighter agreement than 60% between the calculated design values and measured values. Further, a detailed understanding of reactivity control system drive constraints is necessary for the final safety analysis report to satisfy licensing requirements. Therefore, we conclude that current regulatory procedures for the design and operation of reactors should ensure sufficient performance of an MPC controller (excluding risks due to failure modes of the controller which is beyond the scope of this report).

We also include as an appendix a draft of a journal article we plan to submit based on the work documented in this report.

This page is intentionally blank.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

ANL Argonne National Laboratory

ARPA-E Advanced Research Project Agency-Energy

SPM Subcritical Power Modules

TH Thermal Hydraulics/Fluids

STH Simplified Thermal Hydraulics/Fluids

PWR Pressurized Water Reactor

OSQP Operator Splitting solver for Quadratic Program

CMFD Coarse Mesh Finite Difference

HTGR High-Temperature Gas-Cooled Reactor

MPC Model Predictive Control

1. INTRODUCTION

This project aims to characterize the dynamic behavior of High-Temperature Gas-Cooled Reactor (HTGR) like specialized purpose reactors (e.g., microreactors) and examine the feasibility of component design for passive system control. Additionally, we consider traditional control systems as a point of comparison. The focus of this report within the overall project is to develop and demonstrate the integration of state-of-the-art control algorithms with high-fidelity simulation tools for the analysis of load follow operational transients. We pursue this objective as it represents a novel capability for high-fidelity simulation tools, and provides some value to analysts and designers in assessing relationships between instrumentation and control systems and the reactor design.

In prior milestones of this project and related work [3, 4], we explored the use of Model Predictive Control (MPC), to evaluate the feasibility of the reactor design in terms of its controllability through the usual reactivity control systems. However, in this previous work we employed simplified point dynamic models due to an absence of physical plant data and high-fidelity codes capable of representing the dynamic behavior of microreactors effectively. While point reactor dynamics models can deliver acceptable results once they are properly established, they rely on a significant amount of approximations in capturing reactor behavior.

Therefore, in this work, we utilize the high-fidelity reactor analysis code PROTEUS, developed by Argonne National Laboratory (ANL), as a stand-in for a physical reference plant model to couple to the previously developed control algorithms. With the application of this high-fidelity reactor analysis code, we anticipate depicting more realistic reactor behavior, and, thus, a better assessment of the controller's performance.

1.1 Background of Holos Quad Reactor

As a specific use case for an HTGR, we use the reactor design under development by HolosGen. The Holos-Quad design is a scaled-down HTGR with the core being composed of four Subcritical Power Modules (SPM). Each SPM is effectively an independent closed-loop Brayton cycle power

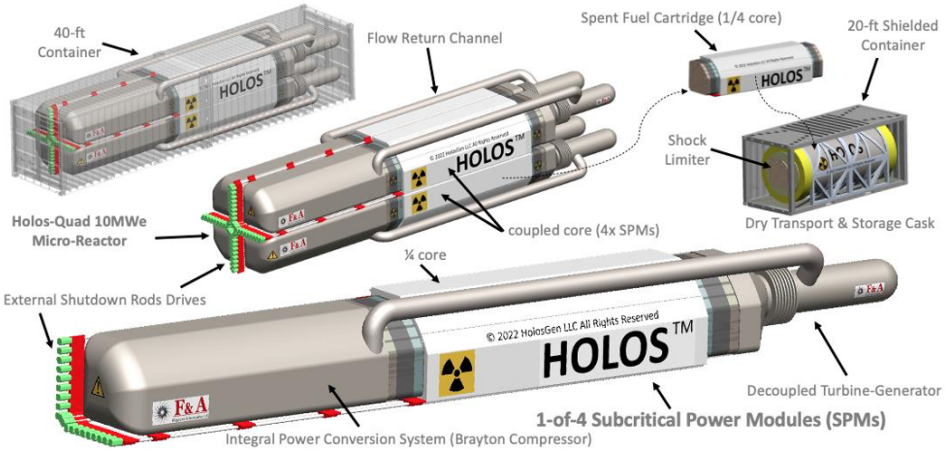


Figure 1. Full-scale Holos-Quad (Gen 2+) subcritical power modules [1, 2]

conversion unit with a nuclear heat source in a tube-shell heat exchanger configuration. This design effectively eliminates the balance of the plant. In earlier designs of this reactor, the four SPMs were configurable so that they will create a critical reactor. An illustration of the Holos-Quad SPMs is shown in Fig. 1.

The design has continued to evolve under the Department of Energy Advanced Research Project Agency-Energy (ARPA-E) MEITNER program. A new, proprietary design was developed by the ARPA-E Resource team at ANL. The new design was reported in [1]. The updated core design is the focus of the calculations and analysis of this report.

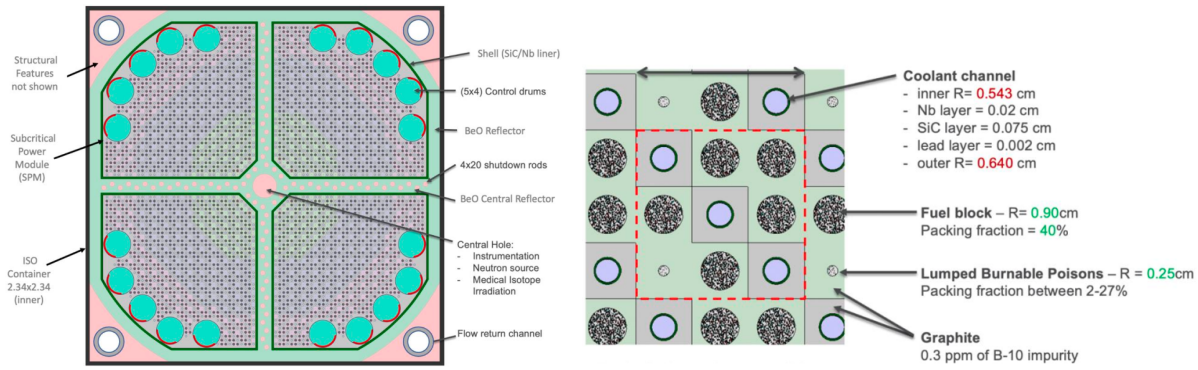
1.2 Organization of this Report

The remainder of this report is structured as follows: Firstly, we first present the approximated core model used in our simulations, that is based on the Holos Quad reactor. Secondly, we outline the methodology used to represent the MPC plant model using PROTEUS. Subsequently, we introduce the methodology and the reduced-order model used in the MPC controller. In the numerical results section, we assess the performance of the MPC controller and present an extensive range of sensitivity tests to verify this code system. Finally, we discuss our numerical results and provide conclusions to this report.

2. HTGR MODEL

This milestone aims to develop and apply the MPC algorithm for HTGR-like microreactors. The Holos-Quad microreactor is one of the proper candidates for this purpose. The microreactor is a Gen-IV microreactor proposed by HolosGen LLC [2]. The radial layout of the Holos-Quad (Gen 2+) is presented in Fig. 2a [1]. The design is an HTGR type, but the reactor has distinctive features such as the SPM and control drums. The rotating control drums located in the radial reflector region are used for load-follow operation. We define unit-cell of the reactor as a 3x3 groups pins as shown in Fig. 2b. In each unit-cell, there are 5 fuel compacts and 3 helium coolant channels, and 1 burnable absorber.

Our initial intent with this milestone was to simulate the Holos-Quad microreactor with the high-

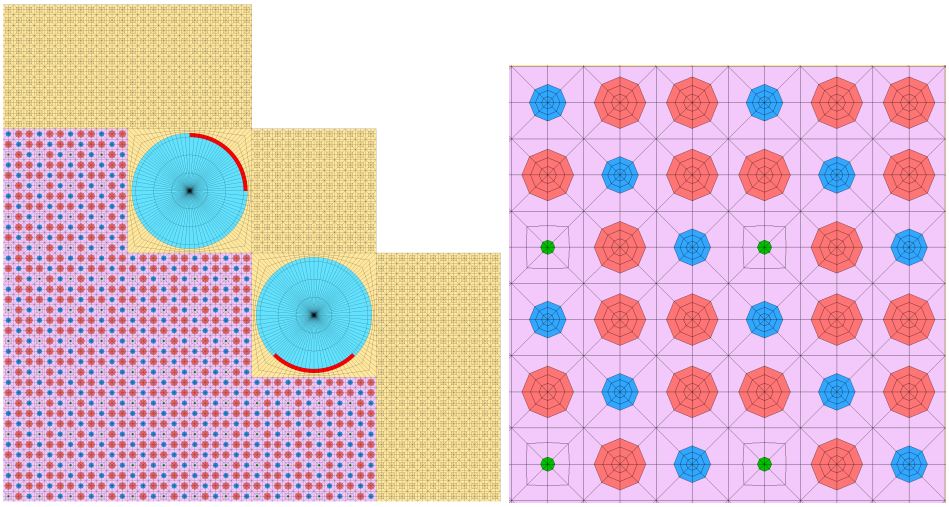


(a) Radial layout **(b) Unit-cell**

Figure 2. Holos-Quad (Gen 2+) model [1]

fidelity reactor analysis code PROTEUS [5]. However, a separate project to develop PROTEUS for Holos-Quad analysis is ongoing and some features were not yet ready for use in the context of this work. Additionally, the full core Holos-Quad model in PROTEUS still presents challenges for computational resources. Therefore, we decided to modify the Holos-Quad model to be smaller and simpler as the primary purpose of this work is to establish, demonstrate, and assess a high-fidelity transient simulation capability integrated with MPC for microreactor load-follow analysis.

Figs. 3a and 3b present the simplified microreactor model and its unit-cell model, respectively. Table 1 compares some design parameters of the Holos-Quad and the simplified microreactor. The simplified microreactor makes use of quarter symmetry in the geometry, and assumes quarter symmetry in the drum operation. The simplified model reduces the core size compared with the Holos-Quad while preserving the linear power density. The control drum has 90 deg of absorber material. In the quarter core, there are two control drums instead of 5, but only one drum is manipulated in the load follow scenarios analyzed in this work to further simplify the problem. In other words, the absorber of one control drum is toward the core peripheral region while the other control drum may rotate to reach a desired power level.



(a) Radial layout **(b) Unit-cell**

Figure 3. Simplified microreactor model

Table 1. Comparison between Holos-Quad and simplified model

Parameter	Holos-Quad (Gen 2+)	Simplified Microreactor
Power (MW)	22.00	2.42
# of fuel compacts	2300	480
Active core height (cm)	380	200
Power density (W/m)	2517	2517
# of coolant channels	1528	288
Core coolant mass flow rate (g/s)	21896	3085
Inlet temperature (K)	863	863
Estimated outlet temperature (K)	1123	1014

3. METHODOLOGY AND TOOL

This section describes the methodologies and computational tools to simulate the load-follow of the simplified microreactor.

3.1 High-Fidelity Model with PROTEUS

3.1.1 Neutronics solver

The PROTEUS code [6] developed by ANL is a high-fidelity neutron transport code based on unstructured finite element meshes that allow users to model complex and unconventional geometry reactor problems like microreactors and Gen-IV reactors. PROTEUS has several methods for performing teh transport calculation. In this work, the MOCEX solver is used which uses 2D MOC radially and the discontinuous Galerkin finite element method axially with extruded geometry [6]. The transient fixed source problem solver makes use of the isotropic approximation of the angular flux time derivative–which is quite common in thermal reactor transient analysis. The transport solution is accelerated by the consistent Coarse Mesh Finite Difference (CMFD) [7]. A 6 energy group cross section library in the ISOTXS format was generated by solving the whole-core Holos-Quad reactor with the Serpent Monte Carlo code for individual materials in the reactor. The cross sections are parameterized as function of temperature between 600 K and 1800 K. While this may not provide realistic predictions, of the actual core cross sections, it is more than sufficient for the capability demonstrated in this report, and subsequent sensitivity analysis.

3.1.2 Thermal Hydraulics/Fluids feedback solver for HTGR

It is necessary to consider the Thermal Hydraulics/Fluids (TH) feedback properly for load-follow simulations. There has been other research coupling PROTEUS and SAM using the MOOSE-based wrapper for the molten salt reactor [8]. Since the coupling of PROTEUS and SAM requires significant efforts and computational time, a Simplified Thermal Hydraulics/Fluids (STH) solver for the HTGR was implemented in this work. The STH solver assumes four regions including fuel compact/graphite moderator/cladding/helium coolant. The coolant channels in each unit cell Figs. 2b and 3b are merged into one coolant channel. The STH solver also assumes the single-phase flow in an axial flow channel under constant pressure. The governing equation for one coolant channel is

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho v}{\partial z} = 0 \,, \tag{1}$$

$$\frac{\partial \rho h}{\partial t} + \frac{\partial \rho h v}{\partial z} = q_f \frac{n_f}{n_c} + q_c = q \,, \tag{2}$$

where ρ is the density; v is the velocity; t is the time; z is the axial coordinate; h is the enthalpy. q_f is the power density per fuel compact; n_f is the number of fuel compact per unit cell; n_c is the number of coolant channel per unit cell; q_c is the heat deposited in the coolant; and q is the total heat source.

The fuel compact, moderator, and cladding in each unit cell are individually homogenized and transformed to an equivalent cylindrical geometry such as Fig. 4.

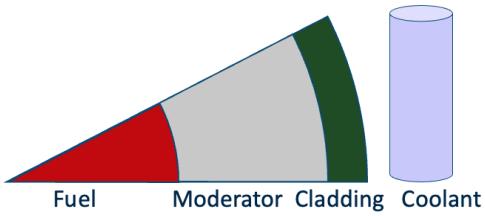


Figure 4. STH solver cylindrical geometry

The STH solver solves the radial heat transfer equation in the 1D cylindrical coordinate:

$$\rho c_p \frac{\partial T}{\partial t} = \frac{\partial}{\partial x} k(T) \frac{\partial T}{\partial x} + q \ , \tag{3}$$

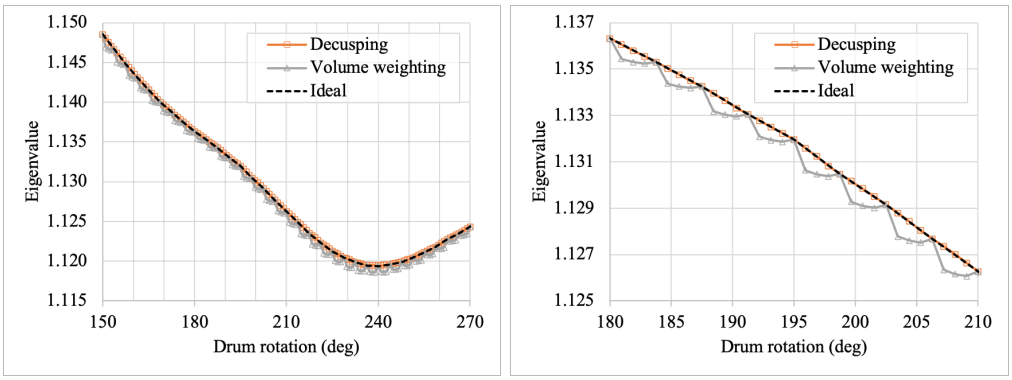
where c_p is the heat capacity; T is the temperature; and k is the thermal conductivity. The boundary condition is applied at the cladding wall as follows:

$$q'' = -k \frac{\partial T}{\partial x}\bigg|_w = h_w (T_w - T_b) \ , \tag{4}$$

where q'' is the heat flux; h_w is the heat transfer coefficient at the surface; T_w is the wall temperature; and T_b is the coolant temperature. Eq. (3) is defined in each region of the geometry so that a system of equations is made for the geometry. The system of equations is solved with the finite difference method. Since the unit cell geometry is approximated to the cylindrical coordinate, the accuracy may be degraded. However, the thermal conductivity and heat transfer coefficients can be chosen such that they preserve the desired temperature of each region at the steady-state condition for some more rigorous method. The desired temperature may come from a solver using a higher-order method as the finite element.

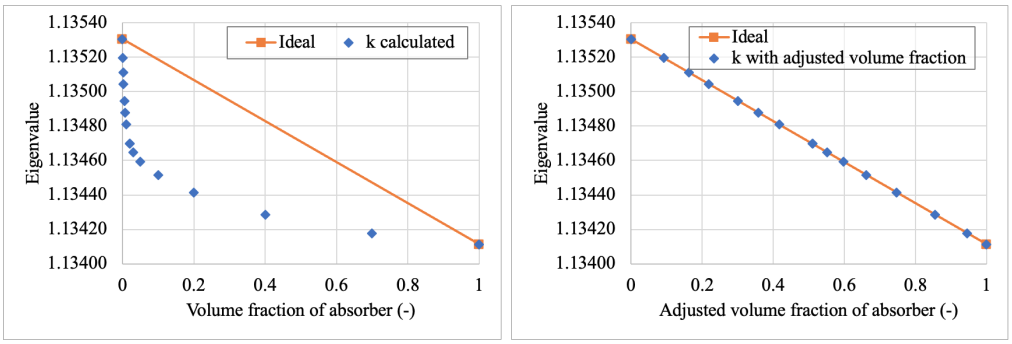
3.1.3 Control drum modeling

The microreactor has multiple control drums and the power level is controlled by the control drums that may rotate continuously in space while the mesh is discrete. This situation can result in significant numerical error effects as the material discontinuity of the absorber moves through the mesh and becomes aligned or homogenized. A possible solution to this is to remesh the problem as the drum moves, however this is non-trivial to accomplish, and often not necessary. It is much simpler to enforce a unified mesh between the previous and current time steps. Doing so greatly simplifies the treatment of the flux or source from the previous time step in solving the transient fixed source equation. As a result, to correctly model the drums, an effective material or cross section should be calculated for the mesh which is partially spanned by the absorber material. Simply homogenizing it based on volume fraction may lead to significant error and cause the so-called control rod or drum cusping issue. To minimize the drum cusping issue, the control drum is divided into 96 angular sectors. However, this mesh discretization was still not sufficient. As shown in the Fig. 5, the volume weighting method causes a cusping effect in the eigenvalue as a function of the control drum rotation. The drum cusping behavior does not represent a realistic situation and it makes the reactivity control difficult since the drum differential worth does not have a monotonic slope.

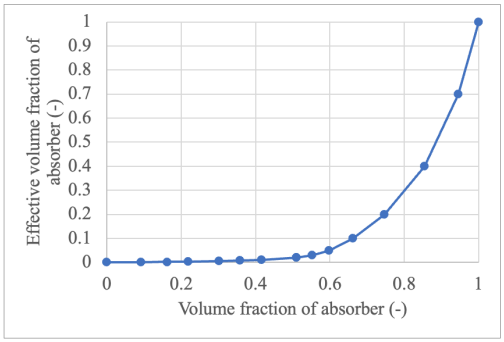


(a) Eigenvalue from 150 to 270 deg **(b) Zoomed in**

Figure 5. Eigenvalue as a function of control drum rotation with different methods



(a) Eigenvalue vs. volume fraction of absorber **(b) Eigenvalue vs. adjusted volume fraction of absorber**



(c) Decusping function

Figure 6. Control drum decusping function

To reduce the cusping effect, a homogenization correction factor has been generated and used in this work. The way to generate the correction factor is similar to Pressurized Water Reactor (PWR) applications [9]. First, 15 eigenvalue calculations were performed with different lengths of control drum absorbers from 90 deg to 93.75 deg. In this calculation, the starting point of the absorber was fixed and only the length was changed. The volume weighting homogenization was used when calculating cross section of the mesh which is partially spanned by the absorber. Recall that the control drum geometry has 96 azimuthally divided pieces so each segment has 3.75 deg. Fig. 6a

shows the eigenvalue as a function of volume fraction. In the figure, 0.0 of volume fraction means the length of absorber is 90 deg while 1.0 means the length of absorber is 93.75 deg. Ideally, the eigenvalue should follow the linear shape. Based on the assumption that the eigenvalue should vary linearly, the adjusted volume fractions were calculated as Fig. 6b. Additional eigenvalue calculations were not necessary to calculate the adjust volume fraction. Instead, the x-axis was shifted so that eigenvalues can be located on the ideal line. Fig. 6c shows the adjust volume fraction as a function of the volume fraction of absorber. This function is used to calculate the effective volume fraction for homogenization of the drum absorber "tip" when the absorber material partially spans a mesh. The function is used for both sides of the absorber arc. With this drum decusping function, the eigenvalue changes smoothly as shown in Fig. 5. This approach has limitations in that the function may not cover other cases than this problem or changes in the control drum design, however it is sufficient for our simulation of a single design here. A more general method may need additional research but the decusping function made in this work behaves well (e.g. agrees with non-homogenized case and provides monotonic reactivity worth curve) for the simplified microreactor.

3.2 Controller

This section describes the methodology of the controller. In this work, the MPC is used for the controller. Since the MPC relies on the mathematical model, it is necessary to set up reasonable models. Section 3.2.1 describes reduced order models for neutronics and TH feedback. Section 3.2.2 describe how to define the state-space equation from the reduced order models. Section 3.2.3 describes the theroy of MPC.

3.2.1 Reduced order models

The neutron kinetics is represented by the point kinetics equations with normalized forms:

$$\frac{d\bar{n}(t)}{dt} = \frac{\rho(t) - \beta_t}{\Lambda}\bar{n}(t) + \frac{\sum_{i=1}^{6} \lambda_i \bar{c}_i(t)}{\Lambda} , \tag{5}$$

$$\frac{d\bar{c}_i(t)}{dt} = \lambda_i \bar{n}(t) - \lambda_i \bar{c}_i(t) , \quad i = 1 \ldots 6 , \tag{6}$$

where $\bar{n}(t)$ is the normalized neutron density at time t; $\rho(t)$ is the reactivity; β_t is the total delayed neutron fraction; Λ is the neutron generation time; λ_i is the delayed neutron time constant for i-th group; and $\bar{c}_i(t)$ is the normalized precursor density for i-th group.

Since the plant model represented by PROTEUS has the TH feedback, it is better to have the lumped TH feedback model for a reduced order model for a more accurate MPC prediction. The lumped TH model has three temperatures for the fuel, graphite moderator, and helium coolant:

$$m_f c_{p,f} \frac{dT_f(t)}{dt} = q_f P_r \bar{n}(t) - K_{f\to m}\left(T_f(t) - T_m(t)\right) , \tag{7}$$

$$m_m c_{p,m} \frac{dT_m(t)}{dt} = (1 - q_f) P_r \bar{n}(t) + K_{f\to m}\left(T_f(t) - T_m(t)\right) - K_{m\to c}\left(T_m(t) - T_c(t)\right) , \tag{8}$$

$$m_c c_{p,c} \frac{dT_c(t)}{dt} = K_{m\to c}\left(T_m(t) - T_c(t)\right) - K_c\left(T_c(t) - T_{in}\right) , \tag{9}$$

where subscripts f, m, and c denote the fuel, moderator, and coolant, respectively; m_x is the mass of material x; $c_{p,x}$ is the heat capacity; $T_x(t)$ is the temperature; q_f is the fraction of heat generated from fuel; P_r is the rated power; $K_{x\to y}$ is the heat transfer coefficient from material x to y; and T_{in} is the inlet coolant temperature.

The rotation of the control drum is also one of the state variables. The following is the equation for the control drum:

$$\frac{dD_1(t)}{dt} = V_1(t) \ , \tag{10}$$

where $D_1(t)$ is the position of control drum; and $V_1(t)$ is the velocity of the drum. The controller determines the velocity.

The reactivity model has the reactivity feedback from temperature, and the reactivity change from control drum:

$$\rho(t) = \rho_b + \alpha_f(t)T_f(t) + \alpha_m(t)T_m(t) + \alpha_c(t)T_c(t) + W_1(t)D_1(t) \ , \tag{11}$$

where ρ_b is the base reactivity; $\alpha_x(t)$ is the temperature coefficient of the material x; and $W_1(t)$ is the differential reactivity worth of the control drum.

The values of the coefficients of these equations are given later in Section 4 for the specific Holos-like mini-core model described in Section 2.

These equations for the reduced order models can be written in a system of equations as follows:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \ , \tag{12}$$

where $\mathbf{x}(t)$ and $\mathbf{u}(t)$ are the state vector and input vector, respectively. $\mathbf{x}(t)$ contains the state variables introduced above, and it is defined as follows:

$$\begin{aligned} \mathbf{x}(t) &= \begin{bmatrix} x_1 & \dots & x_{N_x} \end{bmatrix}^T \\ &= \begin{bmatrix} \bar{n}(t) & \bar{c}_1(t) & \dots & \bar{c}_6(t) & T_f(t) & T_m(t) & T_c(t) & D_1(t) \end{bmatrix}^T \ , \end{aligned} \tag{13}$$

where N_x is the number of state variables.

For simplicity, we assume a single input case for the system therefore N_u is 1 here. With this assumption $\mathbf{u}(t)$ is

$$\mathbf{u}(t) = \begin{bmatrix} u_1 & \dots & u_{N_u} \end{bmatrix}^T = \begin{bmatrix} V_1(t) \end{bmatrix}^T \ . \tag{14}$$

The system of equations, $\mathbf{f}$, is defined as follows:

$$\begin{aligned}\mathbf{f}(\mathbf{x}(t),\mathbf{u}(t)) &= \begin{bmatrix} f_1 & \dots & f_{N_x} \end{bmatrix}^T \\ &= \begin{bmatrix} f_{\bar{n}} & f_{\bar{c}_1} & \dots & f_{\bar{c}_6} & f_{T_f} & f_{T_m} & f_{T_c} & f_{D_1} \end{bmatrix}^T \\ &= \begin{bmatrix} \frac{d\bar{n}(t)}{dt} & \frac{d\bar{c}_1(t)}{dt} & \dots & \frac{d\bar{c}_6(t)}{dt} & \frac{dT_f(t)}{dt} & \frac{dT_m(t)}{dt} & \frac{dT_c(t)}{dt} & \frac{dD_1(t)}{dt} \end{bmatrix}^T \end{aligned} . \tag{15}$$

This form may be helpful for the linearization of the equations.

3.2.2 State-space representation

State space representation is a mathematical model of a physical system expressed as a function of input, output, and state variables related by first-order differential equations or difference equations. All system variables may be represented by a linear combination of the state variable and system inputs by the ordinary differential equations.

It is useful to define the state-space model based on the linearization of the nonlinear equations. The state-space model is used in the control theory; typically to make future predictions about the plant state. The state-space model is defined as follows:

$$\begin{aligned}\dot{\mathbf{x}}_s(t) &= \mathbf{A}_s\mathbf{x}_s(t) + \mathbf{B}_s\mathbf{u}_s(t) \\ \mathbf{y}_s(t) &= \mathbf{C}_s\mathbf{x}_s(t) \ ,\end{aligned} \tag{16}$$

where $\mathbf{A}_s$ is the system matrix with dimensions $N_x \times N_x$; $\mathbf{B}_s$ is the input matrix with dimensions $N_x \times N_u$; and $\mathbf{C}_s$ is the output matrix with dimensions $N_y \times N_x$. $\mathbf{x}_s(t)$, $\mathbf{u}_s(t)$, and $\mathbf{y}_s(t)$ are the state vector, input vector, and output vector of state-space model, respectively. The column vectors, $\mathbf{x}_s(t)$, $\mathbf{u}_s(t)$, and $\mathbf{y}_s(t)$ have length N_x, N_u, and N_y, respectively.

In the state-space representation, it is inherently assumed that $\mathbf{x}_s(t)$, $\mathbf{u}_s(t)$, and $\mathbf{y}_s(t)$ are the differences from nominal condition where the linearization is done. In other words, the vectors are defined as follow:

$$\begin{aligned}\mathbf{x}_s(t) &= \mathbf{x}(t) - \mathbf{x}(t_n) \\ \mathbf{u}_s(t) &= \mathbf{u}(t) - \mathbf{u}(t_n) \\ \mathbf{y}_s(t) &= \mathbf{y}(t) - \mathbf{y}(t_n) \ ,\end{aligned} \tag{17}$$

where t_n is the time at the nominal condition where the linearization is done.

An arbitrary function $g(z)$ can be linearly approximated at a nominal state z_n as follows:

$$g(z) - g(z_n) \approx \frac{dg(z_n)}{dz}(z - z_n) \ . \tag{18}$$

The accuracy of this approximation depends primarily on the assumption that z is "close enough" to z_n such that $g(z)$ behaves linearly.

Similarly, Eq. (12) is approximated at nominal time t_n as follows:

$$\dot{\mathbf{x}}(t) - \dot{\mathbf{x}}(t_n) = \mathbf{A}_s(\mathbf{x}(t) - \mathbf{x}(t_n)) + \mathbf{B}_s(\mathbf{u}(t) - \mathbf{u}(t_n)) \ , \tag{19}$$

where

$$\mathbf{A}_s = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{t=t_n} = \left.\left[\frac{\partial \mathbf{f}}{\partial x_1} \quad \cdots \quad \frac{\partial \mathbf{f}}{\partial x_{N_x}}\right]\right|_{t=t_n} = \left.\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_{N_x}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{N_x}}{\partial x_1} & \cdots & \frac{\partial f_{N_x}}{\partial x_{N_x}} \end{bmatrix}\right|_{t=t_n} , \tag{20}$$

$$\mathbf{B}_s = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{t=t_n} = \left.\left[\frac{\partial \mathbf{f}}{\partial u_1} \quad \cdots \quad \frac{\partial \mathbf{f}}{\partial u_{N_u}}\right]\right|_{t=t_n} = \left.\begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \cdots & \frac{\partial f_1}{\partial u_{N_u}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{N_x}}{\partial u_1} & \cdots & \frac{\partial f_{N_x}}{\partial u_{N_u}} \end{bmatrix}\right|_{t=t_n} . \tag{21}$$

Eq. (19) is the same form as Eq. (16). In many cases, t_n is the time at the initial steady-state, namely $t_n = 0$. The linearization process can be done either by the analytical method or the numerical method. In the following section (Section 3.2.3), the state-space model generated at the initial steady-state condition is used for the controller algorithm. Later in section Section 3.2.4 this assumptionis relaxed.

3.2.3 Model Predictive Control

To solve the state-space control problem, we use the MPC algorithm. MPC is an advanced method to control a process while satisfying a set of constraints [10]. It is based on an iterative finite-horizon optimization of the system (i.e. trajectory optimization). In the form presented here we focus on the linear, time-invariant state-space model for simplicity and clarity. However, there are several variations on MPC, and these can typically be distinguished in how the state space is represented or how the cost function is formulated.

To minimize an error between a desired set-point and predicted output, a control input is computed for a relatively short time horizon in the future by evaluating a cost function. This calculation is then repeated at each subsequent instant or time-window.

The basic theory of MPC is described here for completeness. The following derivation of MPC is based largely on [11]. The state-space model is written as the following time-discretized linear system.

$$\mathbf{x}_d(k+1) = \mathbf{A}_d\mathbf{x}_d(k) + \mathbf{B}_d\mathbf{u}_d(k), \tag{22}$$

$$\mathbf{y}_d(k) = \mathbf{C}_d\mathbf{x}_d(k) \ , \tag{23}$$

where k is the time step index, and $\mathbf{A}_d$, $\mathbf{B}_d$, and $\mathbf{C}_d$ are the discrete forms of $\mathbf{A}_s$, $\mathbf{B}_s$, and $\mathbf{C}_s$, respectively. These discretized matrices are defined with Tayler expansions as follow:

$$\mathbf{A}_d = \sum_{l=0}^{\infty} \frac{1}{l!} (\mathbf{A}_s \Delta t)^l \ , \tag{24}$$

$$\mathbf{B}_d = \left[\sum_{l=1}^{\infty} \frac{1}{l!}\mathbf{A}_s^{l-1}\Delta t^l\right]\mathbf{B}_s \ , \tag{25}$$

$$\mathbf{C}_d = \mathbf{C}_s \ , \tag{26}$$

where Δt is the time difference between step k and $k+1$.

Applying a finite difference approximation to Eq. (23) yields

$$\mathbf{x}_d(k+1) - \mathbf{x}_d(k) = \mathbf{A}_d\left(\mathbf{x}_d(k) - \mathbf{x}_d(k-1)\right) + \mathbf{B}_d\left(\mathbf{u}(k) - \mathbf{u}(k-1)\right) . \tag{27}$$

Next, we make use of the following simplifying notation

$$\Delta(\cdot)(k) = (\cdot)(k) - (\cdot)(k-1) \ , \tag{28}$$

to yield

$$\Delta\mathbf{x}_d(k+1) = \mathbf{A}_d\Delta\mathbf{x}_d(k) + \mathbf{B}_d\Delta\mathbf{u}(k) \ . \tag{29}$$

Note that the input to the state-space model is $\Delta\mathbf{u}_d(k)$. The advantage of using Eq. (29) is that the mismatch between the plant model and the reduced-order model can be canceled, assuming the mismatch varies little between step k and $k+1$. The next step is to relate $\Delta\mathbf{x}_d(k)$ to the output, $\mathbf{y}_d(k)$. To do so, a new state variable vector is defined as

$$\mathbf{x}_a(k) = \begin{bmatrix}\Delta\mathbf{x}_d(k)^T & \mathbf{y}(k)\end{bmatrix}^T \ . \tag{30}$$

This results in the following expressions

$$\begin{aligned}\mathbf{y}(k+1) - \mathbf{y}(k) &= \mathbf{C}_d\left(\Delta\mathbf{x}_d(k+1)\right) \\ &= \mathbf{C}_d\mathbf{A}_d\Delta\mathbf{x}_d(k) + \mathbf{C}_d\mathbf{B}_d\Delta\mathbf{u}(k) \ .\end{aligned} \tag{31}$$

Combining Eqs. (29) and (31) leads to the following discretized state-space model:

$$\overbrace{\begin{bmatrix}\Delta\mathbf{x}_d(k+1)\\ \mathbf{y}_d(k+1)\end{bmatrix}}^{\mathbf{x}_a(k+1)} = \overbrace{\begin{bmatrix}\mathbf{A}_d & \mathbf{O}_{N_x\times N_o}\\ \mathbf{C}_d\mathbf{A}_d & \mathbf{I}_{N_o\times N_o}\end{bmatrix}}^{\mathbf{A}_a}\overbrace{\begin{bmatrix}\Delta\mathbf{x}_d(k)\\ \mathbf{y}_d(k)\end{bmatrix}}^{\mathbf{x}_a(k)} + \overbrace{\begin{bmatrix}\mathbf{B}_d\\ \mathbf{C}_d\mathbf{B}_d\end{bmatrix}}^{\mathbf{B}_a}\overbrace{\Delta\mathbf{u}_d(k)}^{\Delta\mathbf{u}_a(k)}, \tag{32}$$

$$\mathbf{y}_a(k) = \overbrace{\begin{bmatrix}\mathbf{O}^T & \mathbf{I}_o\end{bmatrix}}^{\mathbf{C}_a}\overbrace{\begin{bmatrix}\Delta\mathbf{x}_d(k)\\ \mathbf{y}_d(k)\end{bmatrix}}^{\mathbf{x}_a(k)} , \tag{33}$$

where $\mathbf{O}_{N_x\times N_o}$ is the zero matrix with dimension $N_x \times N_o$; and $\mathbf{I}_{N_o\times N_o}$ is the identity matrix with dimension $N_o \times N_o$. $\mathbf{A}_a$, $\mathbf{B}_a$, and $\mathbf{C}_a$ are called the augmented model which are used in the design of the predictive control.

Assuming that at the sampling instant k, where $k > 0$, some values of the the state variable vector $\mathbf{x}(k)$ are available through measurement, then the state $\mathbf{x}(k)$ provides the current plant information. The future control trajectory is denoted by

$$\Delta\mathbf{u}_a(k),\ \Delta\mathbf{u}_a(k+1)\ \ldots\ \Delta\mathbf{u}_a(k+N_c-1), \tag{34}$$

where N_c is called the control horizon—this dictates the number of time steps into the future for which the control action is simulated for the optimization. With the information given in $\mathbf{x}(k)$, the future state variables are predicted for N_p number of time steps into the future, where N_p is called the prediction horizon. N_p is also the length of the optimization window. We denote the future state variables as

$$\mathbf{x}_a(k|k),\ \mathbf{x}_a(k+1|k)\ \ldots\ \mathbf{x}_a(k+N_p|k), \tag{35}$$

where $\mathbf{x}(k+m|k)$ is the predicted state variable at $k+m$ with given current plant information $\mathbf{x}(k)$. The control horizon N_c is chosen to be less than (or equal to) the prediction horizon N_p.

The state-space model for the prediction horizon is denoted as follows:

$$\mathbf{Y} = \mathbf{F}\mathbf{x}_a(k) + \mathbf{\Phi}\Delta\mathbf{U}, \tag{36}$$

where

$$\mathbf{Y} = \left[\mathbf{y}_a(k+1|k)^T \quad \ldots \quad \mathbf{y}_a(k+N_p|k)^T\right]^T\ , \tag{37}$$

$$\Delta\mathbf{U} = \left[\Delta\mathbf{u}_a(k)^T \quad \ldots \quad \Delta\mathbf{u}_a(k+N_c-1)^T\right]^T\ , \tag{38}$$

$$\mathbf{F} = \begin{bmatrix} \mathbf{C}_a\mathbf{A}_a \\ \vdots \\ \mathbf{C}_a\mathbf{A}_a^{N_p} \end{bmatrix}\ , \tag{39}$$

$$\mathbf{\Phi} = \begin{bmatrix} \mathbf{C}_a\mathbf{B}_a & 0 & 0 & \cdots & 0 \\ \mathbf{C}_a\mathbf{A}_a\mathbf{B}_a & \mathbf{C}_a\mathbf{B}_a & 0 & \cdots & 0 \\ \mathbf{C}_a\mathbf{A}_a^2\mathbf{B}_a & \mathbf{C}_a\mathbf{A}_a\mathbf{B}_a & \mathbf{C}_a\mathbf{B}_a & \cdots & 0 \\ \vdots & & & & \\ \mathbf{C}_a\mathbf{A}_a^{N_p-1}\mathbf{B}_a & \mathbf{C}_a\mathbf{A}_a^{N_p-2}\mathbf{B}_a & \mathbf{C}_a\mathbf{A}_a^{N_p-3}\mathbf{B}_a & \cdots & \mathbf{C}_a\mathbf{A}_a^{N_p-N_c}\mathbf{B}_a \end{bmatrix}\ . \tag{40}$$

For a given set-point signal (or reference trajectory), the objective of the predictive control system is to bring the predicted output as close as possible to the set-point signal. One of the advantages of MPC is that it can use a preview capability for the set-point signal. If an application allows anticipating the signal, the MPC controller with signal previewing can improve reference tracking. This objective is then translated into an input to find the "best" control parameter vector $\Delta\mathbf{U}$ such that an error function between the set-point and the predicted output is minimized. The cost function J, which reflects the control objective, is defined as follows:

$$J = (\mathbf{R}-\mathbf{Y})^T\,(\mathbf{R}-\mathbf{Y}) + \Delta\mathbf{U}^T\mathbf{W}_u\Delta\mathbf{U} + \mathbf{U}^T\mathbf{W}_v\mathbf{U}\ . \tag{41}$$

In Eq. (41), the first term is linked to the objective of minimizing the errors between the predicted output $\mathbf{Y}$ and the set-point signal $\mathbf{R}$. Since this is a product with the transpose it is essentially like the normal equations in the least squares method, so this term is minimizing the L_2-norm of the error in trajectory.

The vector $\mathbf{R}$ contains the set-point information as follows:

$$\mathbf{R} = \begin{bmatrix} \mathbf{r}(k)^T & \mathbf{r}(k+1)^T & \dots & \mathbf{r}(k+N_p-1)^T \end{bmatrix}^T , \tag{42}$$

$$\mathbf{r}(k) = \begin{bmatrix} r_1(k) & r_2(k) & \dots & r_{N_o}(k) \end{bmatrix}^T , \tag{43}$$

where $r_1(k)$ to $r_{N_o}(k)$ are the set-point signals corresponding to output vector. If an application does not allow previewing the set-point signals for some reason, $\mathbf{r}(k)$ can then be used for the rest of the prediction horizon.

The second term reflects the consideration given to the size of $\Delta\mathbf{U}$. This term essentially allows one to minimize the control "effort". $\mathbf{W}_u$ is a diagonal matrix in the form that $\mathbf{W}_u = w_u \mathbf{I}_{N_c \times N_c}$ $(w_u \geq 0)$ where w_u is used as a weighting parameter for the desired closed-loop performance.

The third term also reflects the consideration to minimize the control effort but the third term is to reduce $\mathbf{u}_a$ instead of $\mathbf{\Delta u}_a$. This means that we desire a control action that is not very oscillatory or "jerky"–just as we do want to avoid large (in magnitude control actions), like rotation 1 control drum a significant distance, we also want to avoid solutions that unnecessarily move the control drums back and forth.

$\mathbf{U}$ is defined as follows:

$$\begin{aligned} \mathbf{U} &= \begin{bmatrix} \mathbf{u}_a(k)^T & \dots & \mathbf{u}_a(k+N_c-1)^T \end{bmatrix}^T \\ &= \mathbf{C}_1 \mathbf{u}_a(k-1) + \mathbf{C}_2 \Delta\mathbf{U} , \end{aligned} \tag{44}$$

where $\mathbf{C}_1 = \mathbf{1}_{N_c \times 1} \otimes \mathbf{I}_{N_u \times N_u}$; $\mathbf{1}_{N_c \times 1}$ is the 1-vector with dimension $N_c \times 1$; $\mathbf{C_2} = \mathbf{T} \otimes \mathbf{I}_{N_u \times N_u}$; and $\mathbf{T}$ is the lower triangle with all non-zero elements are 1 and dimension $N_c \times N_c$. $\mathbf{W}_v$ in the third term is a diagonal matrix in the form that $\mathbf{W}_v = w_v \mathbf{I}_{N_c \times N_c}$ which w_u is used as a weighting parameter.

The cost function is next rewritten as a function of $\Delta\mathbf{U}$ based on the simplification from Eq. (29). This form also facilitates solution by quadratic programming.

$$J = \Delta\mathbf{U}^T \mathbf{H} \Delta\mathbf{U} + 2\Delta\mathbf{U}^T \mathbf{L} + c , \tag{45}$$

where

$$\mathbf{H} = \mathbf{\Phi}^T \mathbf{\Phi} + \mathbf{W}_u + \mathbf{C}_2^T \mathbf{W}_v \mathbf{C}_2 , \tag{46}$$

$$\mathbf{L} = \mathbf{\Phi}^T \left(\mathbf{F}\mathbf{x}_a(k) - \mathbf{R} \right) + \mathbf{C}_2^T \mathbf{W}_v \mathbf{C}_1 \mathbf{u}_a(k-1) , \tag{47}$$

and c is the remainder which is not multiplied to $\Delta\mathbf{U}$. The remainder, c, is not used when solving the quadratic programming problem.

One of the other advantages of MPC is that design constraints can be considered in the optimization problem. To incorporate the design constraints into the control problem, it is necessary to translate the constraints into linear inequalities. The constraints are taken into consideration for each moving

horizon window. Since the MPC problem is formulated and solved in the framework of the receding horizon control, the constraints on the rate of change are expressed as

$$\Delta \mathbf{u}^{min} \leq \Delta \mathbf{u}_a(k) \leq \Delta \mathbf{u}^{max}, \tag{48}$$

where the superscripts *min* and *max* denote the minimum and the maximum constraints, respectively. The constraints are defined within the control horizon so that it can be expressed in terms of function $\Delta \mathbf{U}$ as follows:

$$\Delta \mathbf{U}^{min} \leq \Delta \mathbf{U} \leq \Delta \mathbf{U}^{max} , \tag{49}$$

where $\Delta \mathbf{U}^{min}$ and $\Delta \mathbf{U}^{max}$ are column vectors with N_c elements of $\Delta \mathbf{u}^{min}$ and $\Delta \mathbf{u}^{max}$, respectively.

The constraints then need to be decomposed into two parts to reflect the lower and the upper limit. This is expressed as follows:

$$\begin{bmatrix} -\mathbf{C}_1 \\ \mathbf{C}_1 \end{bmatrix} \Delta \mathbf{U} \leq \begin{bmatrix} -\Delta \mathbf{U}^{min} \\ \Delta \mathbf{U}^{max} \end{bmatrix} , \tag{50}$$

where $\mathbf{I}$ is the identity matrix, and its size depends on N_c and $\mathbf{u}$.

This procedure also applies to the control input and output constraints. All constraints are expressed in terms of $\Delta \mathbf{U}$. The constraints for the velocity, which $\mathbf{U}$ defined in Eq. (51), is written as follows:

$$\mathbf{U}^{min} \leq \mathbf{C}_1 \mathbf{u}_a(k-1) + \mathbf{C}_2 \Delta \mathbf{U} \leq \mathbf{U}^{max} . \tag{51}$$

Using Eq. (36), the output constraints are expressed in terms of $\Delta \mathbf{U}$ as:

$$\mathbf{Y}^{min} \leq \mathbf{F}\mathbf{x}(k) + \mathbf{\Phi} \Delta \mathbf{U} \leq \mathbf{Y}^{max}. \tag{52}$$

Finally, the MPC in the presence of constraints is proposed as finding $\Delta \mathbf{U}$ that minimizes the quadratic cost function

$$J = \Delta \mathbf{U}^T \mathbf{H} \Delta \mathbf{U} + 2 \Delta \mathbf{U}^T \mathbf{L} + c , \tag{45 revisited}$$

subject to the inequality constraints:

$$\mathbf{M} \Delta \mathbf{U} \leq \mathbf{N} , \tag{53}$$

where

$$\mathbf{M} = \begin{bmatrix} -\mathbf{C}_2 \\ \mathbf{C}_2 \\ -\mathbf{C}_1 \\ \mathbf{C}_1 \\ -\mathbf{\Phi} \\ \mathbf{\Phi} \end{bmatrix} ; \ \mathbf{N} = \begin{bmatrix} -\mathbf{U}^{min} + \mathbf{C}_1\mathbf{u}(k-1) \\ \mathbf{U}^{max} - \mathbf{C}_1\mathbf{u}(k-1) \\ -\Delta\mathbf{U}^{min} \\ \Delta\mathbf{U}^{max} \\ -\mathbf{Y}^{min} + \mathbf{F}\mathbf{x}(k) \\ \mathbf{Y}^{max} - \mathbf{F}\mathbf{x}(k) \end{bmatrix}. \tag{54}$$

To solve this quadratic programming problem, the Operator Splitting solver for Quadratic Program (OSQP) solver is used [12]. The OSQP solver is a numerical optimization package for solving convex quadratic programs. The solver is very efficient and accurate for these kinds of convex quadratic programming problems. The OSQP solver is not applicable to the more complicated nonlinear optimization problem for nonlinear MPC. However, the optimization problem with a nonlinear model can be handled with the adaptive MPC which is not much different from the standard MPC. In adaptive MPC the state-space model is updated during simulation by re-linearizing the nonlinear model every time step. A detailed description of the algorithm is presented next in Section 3.2.4.

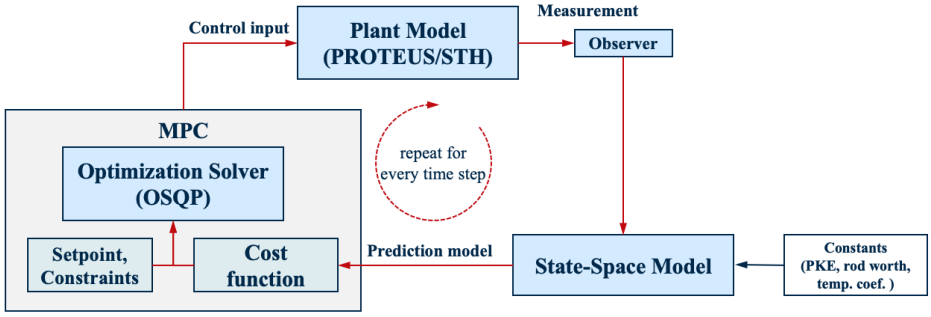


Figure 7. Calculation flow with standard MPC

Although the optimal parameter vector $\Delta\mathbf{U}$ contains the controls $\Delta\mathbf{u}_a(k), \Delta\mathbf{u}_a(k+1), \ldots, \Delta\mathbf{u}_a(k+N_c-1)$, with the receding horizon control principle, the first sample of this sequence, i.e. $\Delta\mathbf{u}(k)$, is implemented only while ignoring the rest of the sequence. When the next sample period arrives, the more recent measurement is taken to form the state vector $\mathbf{x}(k+1)$ for calculation of the new sequence of control signal. This procedure is repeated in real time to give the receding horizon control law. Fig. 7 presents the calculation flow of the standard MPC.

3.2.4 Adaptive MPC

The theory of MPC is presented in Section 3.2.3. The standard MPC described in Section 3.2.3 is based on the state-space model which is linearized or defined at a particular time such as an initial steady-state condition. However, the reduced order model described in Section 3.2.1 is not linear. Ignoring the nonlinearity of the reduced order model then, may degrade the accuracy of reference tracking. Applying nonlinear MPC [13] may resolve the issue. However, the nonlinear MPC is computationally expensive and a nonlinear constraint is rarely necessary for many applications. Therefore, this work uses the adaptive MPC [14]. The adaptive MPC controller adjusts its prediction model at run time to compensate for nonlinear or time-varying characteristics. Figs. 7 and 8 present

the calculation flow with the standard MPC and the adaptive MPC.

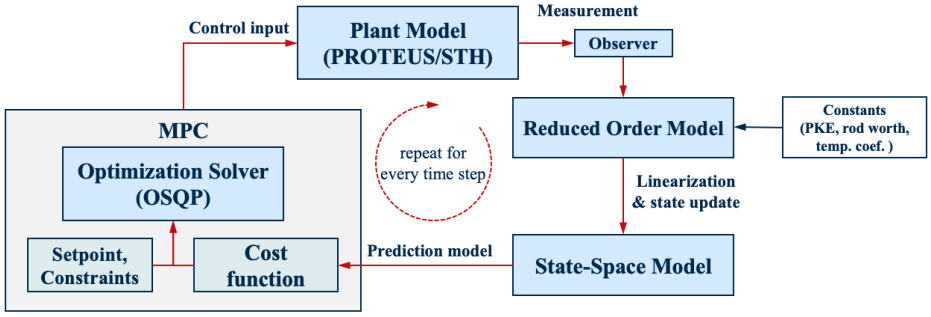


Figure 8. Calculation flow with adaptive MPC

The only difference between the standard MPC and the adaptive MPC is that the state-space model is "re-linearized" during the simulation in the adaptive MPC. To do this, the reduced order model defined in Eq. (12) needs to be solved during the simulation as well as the plant model and the state-space model. The state-space model is updated every time step or as needed by using the exact same process in Section 3.2.2. The MPC problem is updated repeatedly with the updated state-space model. Compared to the standard MPC, the adaptive MPC requires additional calculations to solve the reduced order model and repetitive setup of the state-space model and MPC problem. However, in our application, the overhead is negligible and is readily manageable with modern computing power while the accuracy of the adaptive MPC may be comparable to that of the nonlinear MPC as long as the linearization is repeated with a short time interval. Therefore, the adaptive MPC can be a good trade-off between the standard MPC and the nonlinear MPC.

3.2.5 Model mismatch and observer

The MPC optimization is conducted based on the reduced order state-space model. It is nearly impossible to directly use the plant model or a high-fidelity model in the MPC optimization due to the complexity and computational cost. Instead, the reduced order model is used as described in Section 3.2.1. The issue is that the reduced order model also approximates the plant or high-fidelity model. It means that the solutions from the plant model and the reduced order model are different. Therefore, there should be a process to correct the difference. This is done by an observer.

Before explaining the observer, it is necessary to mention the measurement. It is necessary to consider which state variable can be obtained from the plant or high-fidelity model. In practice, it is not feasible to measure some or all state variables (e.g. the delayed neutron precursors and core-averaged temperature for each material cannot be directly measured). The power level or relative neutron density (i.e. $\bar{n}$) can usually be indirectly measured in the real plant through in-core or ex-core neutron detectors. In this sense, it is assumed that only the power level or relative neutron density is obtained from the plant model in the simulation and the observer is used to correct the neutron density. We note that some other properties, like core average coolant temperature and inlet or outlet temperatures can also be measured–but for the current work on a single-output problem, these are not used.

The measurement correction process done by the observer is quite straightforward. Once the measurement is obtained from the plant or high-fidelity model, the output vector, $\mathbf{y}_d(k)$ in Eqs. (32)

and (33) is corrected as follows:

$$\mathbf{y}'_d(k) = \mathbf{y}_d(k) + \mathbf{K}\left(\tilde{\mathbf{y}}(k) - \mathbf{y}_d(k)\right) \tag{55}$$

where $\mathbf{y}'_d(k)$ is the corrected output vector; $\tilde{\mathbf{y}}(k)$ is the output vector from the plant; and $\mathbf{K}$ is the observer matrix with dimension $N_o \times N_o$. When the optimization problem is a single output case and only power can be measured from the plant, the $\mathbf{K}$ would be $\mathbf{1}_{1\times1}$. What the observer is doing is simply replacing the output from the state-space model with measurable output from the plant. The corrected output vector, $\mathbf{y}'_d(k)$, is then used in the subsequent MPC optimization instead of $\mathbf{y}_d(k)$.

An additional correction is necessary for the adaptive MPC since the adaptive MPC algorithm solves the reduced order model separately and the measurement correction needs to be applied to the reduced order model as well. The observer for the state-space model simply corrects the output vector. However, only updating the neutron density in Eq. (12) may lead to an unphysical neutron time derivative since the corrected neutron density is not calculated from the reduced order model. Consequently, there is no guarantee of neutron balance between time steps. Thus, an additional calculation to preserve the time derivative of neutron density is necessary to resolve the issue. In other words, the following relation must also be satisfied:

$$\frac{d\bar{n}(t)}{dt} = \frac{d\bar{n}'(t)}{dt} \,, \tag{56}$$

where $\bar{n}'(t)$ is the corrected neutron density which is the same as the neutron density of the plant model. To preserve the time derivative with the corrected neutron density, an additional parameter is needed. This is handled by updating the base reactivity. Using Eq. (5), the above equation is written as following:

$$\frac{\rho(t) - \beta_t}{\Lambda}\bar{n}(t) + \frac{\sum_{i=1}^{6}\lambda_i\bar{c}_i(t)}{\Lambda} = \frac{\rho'(t) - \beta_t}{\Lambda}\bar{n}'(t) + \frac{\sum_{i=1}^{6}\lambda_i\bar{c}_i(t)}{\Lambda} \,, \tag{57}$$

where $\rho'(t)$ is the corrected reactivity corresponding to $\bar{n}'(t)$ and it is defined as a function of corrected and uncorrected neutron densities:

$$\rho'(t) = (\rho(t) - \beta_t)\,\frac{\bar{n}(t)}{\bar{n}'(t)} + \beta_t \,. \tag{58}$$

The correction on the reactivity may be assumed so that the base reactivity, ρ_b, is updated to have $\rho'(t)$. The updated neutron density and the reactivity are then used in the subsequent time steps.

4. NUMERICAL RESULTS

4.1 3D microreactor Results

Table 2. Applied constraints to Holos reactor control

Parameters	Constraints
Control drum rotation (deg)	$0 \leq \mathbf{s} \leq 360$
Control drum rotation rate (deg/s)	$-0.36 \leq \mathbf{u} \leq 0.36$
Control drum rotation acceleration (deg/s^2)	$-0.36 \leq \Delta\mathbf{u}/\Delta t \leq 0.36$

In this section, the load-follow simulation results for the HTGR-type microreactor described in Section 2 are presented. As mentioned in Section 3.2.3, one of the advantages of the MPC is an ability to consider constraints in the optimization problem. Three kinds of constraints are applied to the drum rotation rate $\mathbf{u}$, and drum rotation acceleration $\Delta\mathbf{u}/\Delta t$. It should be noted that these constraints are assumed because they have not been determined in the reactor design, yet. The numerical values of the constraints are listed in Table 2. There is no specific constraint for the control drum rotation since the drum can rotate over 0 degree or 360 degree. The constraints for $\mathbf{u}$ and $\Delta\mathbf{u}/\Delta t$ are simply chosen to have ± 0.1%/s and ± 0.1%/s^2, respectively. These constraints are corresponding to ± 0.36 deg/s and ± 0.36 deg/s^2, respectively. In reality, the constraints should be first based on the mechanical performance of the drum rotation system, and secondarily on technical specification limits for the operation that should satisfy safety and operational performance requirements.

Table 3. Parameters used in the reduced order model simulation

Parameter	Value	Unit	Parameter	Value	Unit
β	0.01181	-	Λ	2.627E-5	s
β_1	0.00041	-	λ_1	0.01334	1/s
β_2	0.00213	-	λ_2	0.03274	1/s
β_3	0.00204	-	λ_3	0.12078	1/s
β_4	0.00457	-	λ_4	0.30278	1/s
β_5	0.00187	-	λ_5	0.84949	1/s
β_6	0.00078	-	λ_6	2.85300	1/s
α_f	-4.0	pcm/K	c_f	977.0	J/kg/K
α_m	-0.2	pcm/K	c_m	1697.0	J/kg/K
α_c	0.0	pcm/K	c_c	5190.0	J/kg/K
W_1	-33.3	pcm/deg	-	-	-

Table 4. Power level setpoint

Time period (sec)	Description for power level
0 - 5	Maintain 100%
5 - 245	Decrease to 20% with ramp rate of -20%/min
245 - 1800	Maintain 20%
1800 - 2040	Increase to 20% with ramp rate to 20%/min
2040 - 3600	Maintain 100%

Table 3 presents parameters used in the reduced order model. The delayed neutron fraction and the delayed neutron time constants were extracted from the 6-group cross section library of PROTEUS. A typical generation time for thermal reactors is used. Therefore, the point kinetics parameters are not necessarily accurate. Unfortunately, the feature to generate the point kinetics parameters in PROTEUS was not available for this work. Nevertheless, these assumptions and their validity do not preclude the development and assessment of the coupling of high-fidelity simulations with control algorithms.

In Section 4.2, the sensitivity test results are presented to determine the effect of these parameters. The temperature coefficients and control drum differential worth were determined from a series of 2D core simulations.

Fig. 9 presents the results for the 3D microreactor load follow simulation. The prescribed power scenario is given in Table 4, and starts at 100%, decreases to 20%, and then increases back to 100%. The ramp rate between 100% and 20% power is ±20%/min which is the fastest ramp rate requirement identified in [3]. As depicted in Fig. 8, the plant is represented by PROTEUS coupled with STH. The adaptive MPC is used for the simulation. A one second time step is used for exchanging information between the PROTEUS simulation and MPC controller. The initial position of the control drum is 180 deg. This means the center of the control drum absorber faces south as shown in Fig. 3a. In the simulation, 40 computing cores were used.

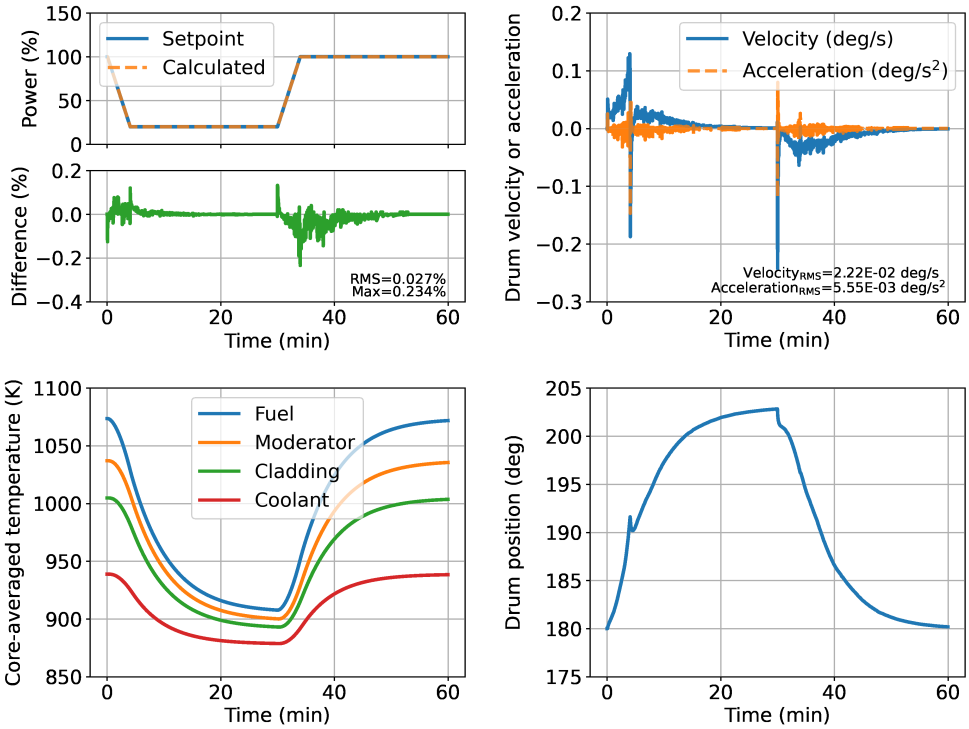


Figure 9. 3D microreactor load-follow simulation results

The objective of the controller is to compute the control inputs so that the core power matches the setpoint while minimizing control costs, and being regulated within the given constraints. The calculated power with the MPC controller is very accurate. The RMS tracking error over the entire

simulation is 0.027%. The maximum tracking error is 0.234%. The core-averaged temperature is also calculated for the fuel, moderator, cladding, and coolant. The temperature changes much more slowly than power. This is expected given the slow thermal time constants of graphite. The power level decreases to 20% and maintains the level between 4 min and 30 min. In this period, the MPC controller needs to keep solving the optimization problem to compensate for the temperature feedback effect by adding negative reactivity with the control drum. Therefore, during this same period, the control drum keeps rotating. Similar behavior is repeated when the power increases back to 100% but with opposite direction of rotation. During the load-follow simulation, the control drum rotates between 180 deg and 203 deg.

To determine the cost for input control, RMS velocity and RMS acceleration are calculated and shown in Fig. 9. The RMS velocity and RMS acceleration are 2.22E-2 deg/s and 5.55E-3 deg/s^2, respectively. It is not straightforward to determine whether these costs are large or small in and of themselves, but this provides some metric for a relative comparison. Consequently, the same parameters are calculated for several other 2D core simulations and then compared to each other in the next section.

4.2 Sensitivity Tests

Some amount of error in the reduced order model is acceptable and handled by the observer which corrects the model error by incorporating power data (e.g., a measurement) from the "plant" (the PROTEUS simulation). However, it is still necessary to have a sufficiently accurate surrogate model so that the controller can produce an accurate adn stable simulation result. In this section, sensitivity tests are performed to determine which parameters in the state space representation are most important to the overall accuracy of the controller. As mentioned in Section 4.1, some parameters were assumed since PROTEUS does not have a feature to calculate them online. However, the load-follow result for the 3D microreactor was very accurate.

4.2.1 2D core vs. 3D core

To minimize the computational resources of the sensitivity study, all sensitivity tests are performed with a 2D core model. The 2D core has the same design and radial size as the 3D core, however the axial boundary condition at the top and bottom surface of the core is now reflective. Therefore, only one axial plan is necessary to simulate the core model. Fig. 10 shows the 2D core simulation results. Table 5 shows the tracking error and control cost for the 2D and 3D reactor simulations. This comparison is made to confirm that the 2D core model has similar behavior as the 3D core so that we have some confidence in the validity of our conclusions from the subsequent sensitivity tests.

Table 5. 2D and 3D simulation results summary

Description	Tracking difference (%)		Normalized control cost	
	RMS	Max	Velocity	Acceleration
3D core simulation	0.027	0.234	1.09	1.09
2D core simulation	0.017	0.170	1.00	1.00

Table 5 presents the comparison between the 2D and 3D core simulation. Both 2D and 3D core simulations have a similar level of accuracy. The 2D core has a smaller tracking error, i.e., the

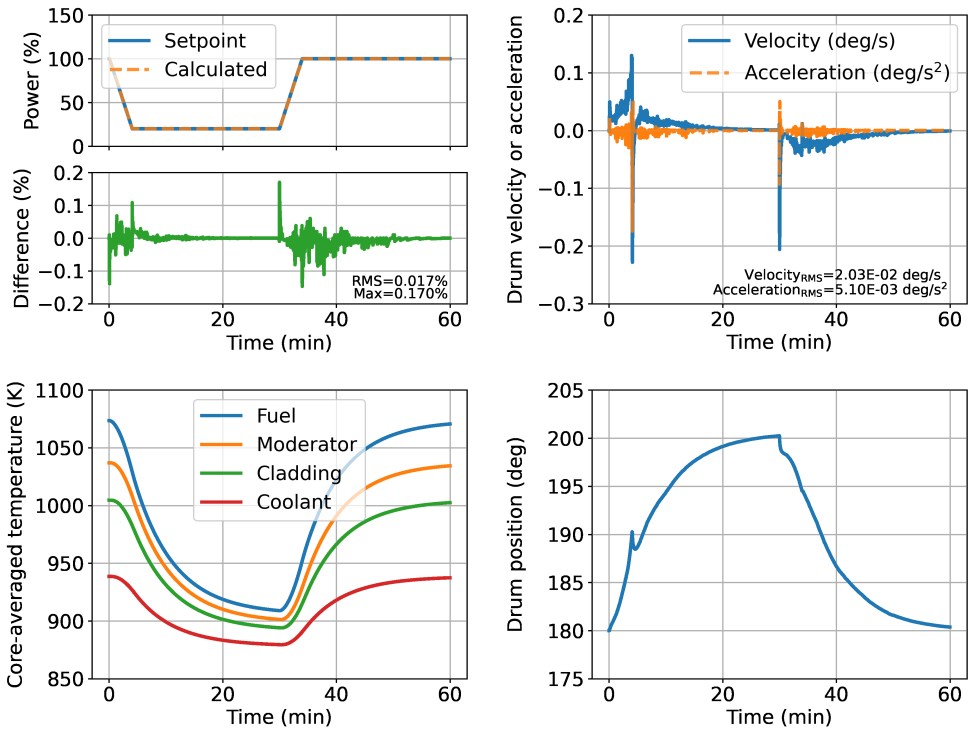
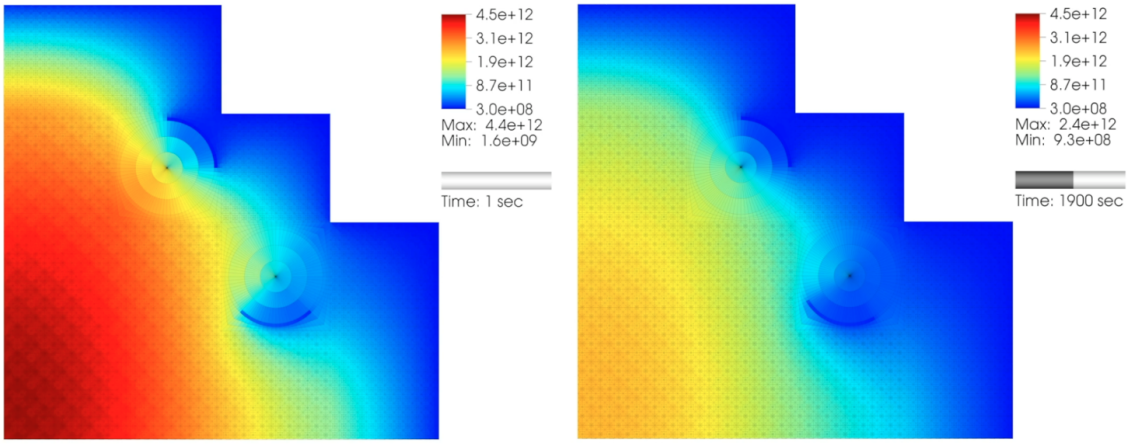


Figure 10. 2D microreactor load-follow simulation results

maximum error is 0.17%. The normalized control cost in Table 5 indicates the RMS velocity and RMS acceleration of the control drum over the entire simulation but these are normalized by the costs of 2D core simulation. The normalization is performed since it is not straightforward to determine whether the control costs are expensive or not based on the absolute values. Therefore, relative or normalized control costs are calculated and presented. The 3D core simulation shows



(a) At 1 sec **(b) At 1900 sec**

Figure 11. 2D core flux distribution

about 9% higher control costs for both control drum velocity and acceleration than those of the 2D core. This is likely explained by the control drums not having a uniform worth axially along their length, as the worth near the axial boundaries should be reduced since neutrons are able to leak and reflect back in these areas. Moreover, the reduced order model may approximate the 2D core model more accurately since the control drum differential worth and temperature reactivity coefficients were generated based on the 2D core simulation. These are the reasons why the 2D core load follow results can be observed to have higher accuracy and lower control cost than those of the 3D core simulation. However, these differences are not significant and it can be assumed that the 2D core load follow simulation has a similar behavior as the 3D core. Therefore, the 2D core model is used for the rest of the sensitivity tests.

Fig. 11 displays the flux distribution from the 2D core simulation at 1 and 1900 seconds. In the figure, it can be observed that the flux is significantly reduced due to the control drum. As the 3D core utilizes an extruded geometry derived from the 2D core geometry, the flux distribution of the 3D core is expected to be similar to that of the 2D core.

4.2.2 Adaptive MPC vs. Standard MPC

In Sections 4.1 and 4.2.1, the results are based on the adaptive MPC controller. As discussed in Section 3.2.4, the adaptive MPC can consider the nonlinear characteristics of the reduced order model without the need to invoke nonlinear optimization methods. The impact of considering the nonlinearity can be determined by comparing it to the standard MPC case. Fig. 12 shows the results with the standard MPC controller. Table 6 summarizes the accuracy and control costs.

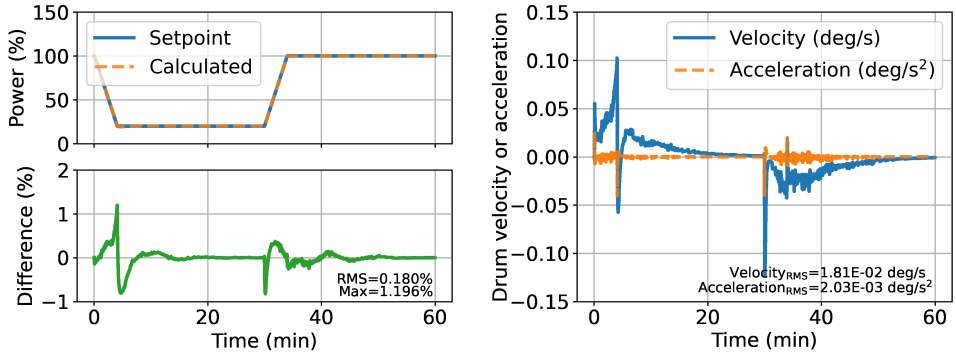


Figure 12. 2D microreactor load-follow simulation results with standard MPC

Table 6. Comparison between adaptive MPC and standard MPC

Description	Tracking difference (%)		Normalized control cost	
	RMS	Max	Velocity	Acceleration
Adaptive MPC (Base case)	0.017	0.170	1.00	1.00
Standard MPC	0.180	1.196	0.89	0.40

The standard MPC has less accurate results compared with the adaptive MPC. The RMS tracking error is 0.18% and the maximum error is 1.196%. The difference may be acceptable but there is clear accuracy degradation due to the methodology of the controller. The main difference comes from the time-varying components of the matrix $\mathbf{A}_s$ in the state-space equations (i.e. Eq. (16)).

Fig. 13 shows the time-varying components of $\mathbf{A}_s$ which is calculated from the adaptive MPC. Four elements from the adaptive MPC highly depend on time and vary significantly. These terms are essentially the nonlinear terms of the point reactor model. Especially, $df_{\bar{n}}/dT_f$, $df_{\bar{n}}/dT_m$, and $df_{\bar{n}}/dD_1$ have upto 80% differences compared to those at the initial condition. On the other hand, the state-space model of the standard MPC is generated at the initial steady state and does not vary during the simulation. It means the standard MPC overestimates the magnitude of $df_{\bar{n}}/dT_m$, and $df_{\bar{n}}/dD_1$ between 4 min and 30 min. According to the power difference in Fig. 12, once a tracking error is calculated for any reason, the tracking error does not shrink rapidly. The controller needs to calculate more aggressive control inputs to eliminate the tracking error. However, the standard MPC controller is unable to do so since it overestimated $df_{\bar{n}}/dD_1$. The normalized control costs of the standard MPC are smaller than those of the adaptive MPC due to the same reasons—overestimated $df_{\bar{n}}/dD_1$. If the state-space model is linearized at low power, then the controller may overshoot the control input due to underestimated $df_{\bar{n}}/dD_1$, and the power may oscillate—-this result is not presented only inferred.

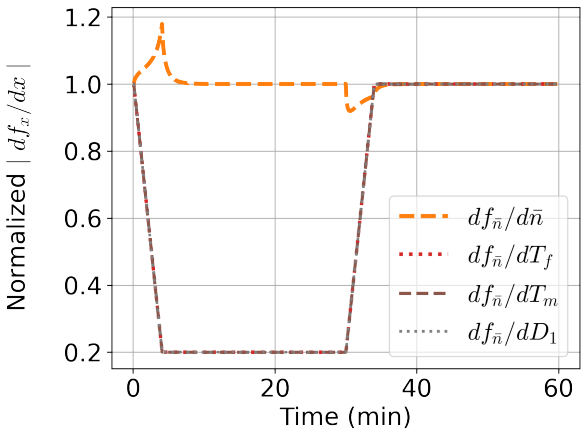


Figure 13. Time-varying components of matrix $\mathbf{A}_s$ from adaptive MPC

4.2.3 Prediction and control horizons

The MPC computes the optimal control inputs within the prediction and control horizons, but only the first control input is actually implemented in the plant model. The calculations made by the MPC controller are aimed at minimizing the tracking error throughout the entire prediction horizon. However, merely enhancing the number of horizons does not always ensure improved accuracy. On the contrary, utilizing a small number of horizons does not guarantee superior accuracy either, as it may potentially compute an aggressive input focusing solely on the immediate time frame.

Figs. 14 and 15 and Table 7 depict the simulation results corresponding with varying numbers of horizons. While all instances demonstrate reasonable accuracy and stability, minute discrepancies merit further examination. With smaller numbers of horizons as depicted in Fig. 14, the overall tracking accuracy sees a slight enhancement compared to the base case. As anticipated, a larger maximum error of 0.213% is evident—this is likely due to the more aggressive inputs calculated by taking into account immediate timing rather than a wider time window, resulting in higher normalized control costs.

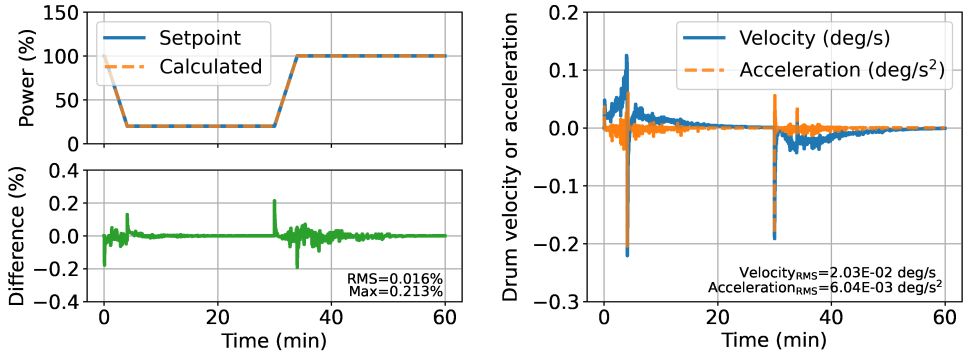


Figure 14. 2D core simulation results with $N_p = 2$ and $N_c = 1$ time step(s)

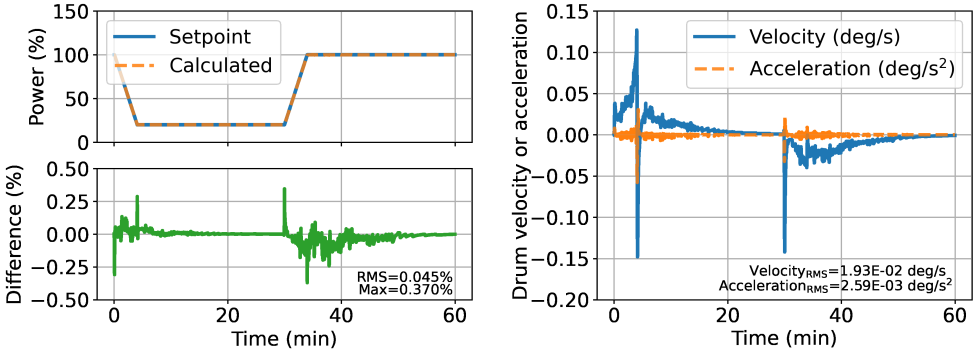


Figure 15. 2D core simulation results with $N_p = 8$ and $N_c = 4$ time steps

Table 7. Comparison with different prediction and control horizons

Description	Tracking difference (%)		Normalized control cost	
	RMS	Max	Velocity	Acceleration
$N_p = 2, N_c = 1$	0.016	0.213	1.00	1.18
$N_p = 4, N_c = 2$ (Base case)	0.017	0.170	1.00	1.00
$N_p = 8, N_c = 4$	0.045	0.370	0.95	0.51

Conversely, the case with the higher number of horizons, shown in Fig. 15, employs a reduced control cost through minimizing control input and overshooting to achieve the desired power. The modification in reactivity resultant from control drum rotation is immediate, while the reactor also exhibits several delayed responses. This scenario tends to rotate the control drum more slowly due to the delayed effect's contribution to power alteration. Even though most of the power change occurs instantly, cases with extended horizons experience a larger RMS error of about 0.045%. These results suggest that it is crucial to leverage a specific time period in the optimization problem to obtain stable and accurate results.

4.2.4 Cost function weight

For the cost function given in Eq. (41), there are two weighting factors involved. These factors within $\mathbf{W}_v$ and $\mathbf{W}_u$ serve to manipulate the significance of the cost of control inputs. It is generally the case that a smaller weight allows for the use of a more aggressive control input to attain greater

accuracy. Taking into consideration that these weights do not have an identifiable physical unit makes it challenging to intuitively assess what the magnitude of the weights used should be. In this section the weight factor is modified to be 0.1 and 10 times that of the base case.

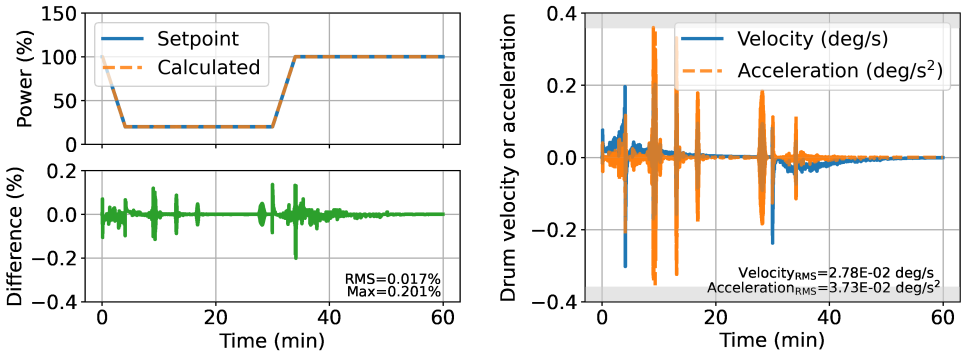


Figure 16. 2D core simulation results with 10 times smaller control input weights

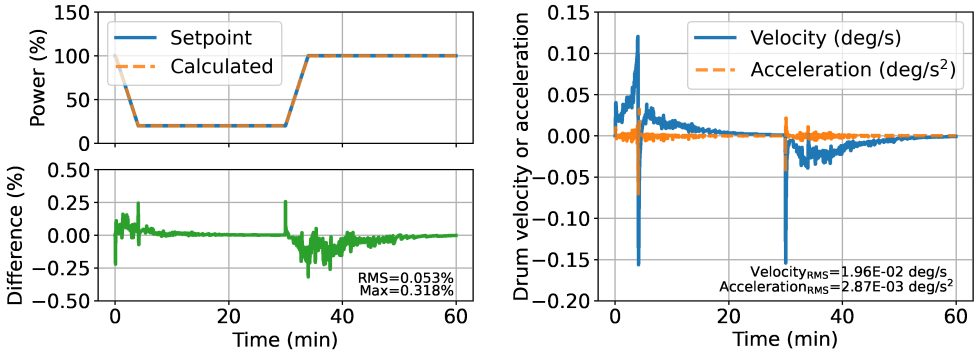


Figure 17. 2D core simulation results with 10 times larger control input weights

Table 8. Comparison with different prediction and control horizons

Description	Tracking difference (%)		Normalized control cost	
	RMS	Max	Velocity	Acceleration
Weights ×0.1	0.017	0.201	1.37	7.31
Base case	0.017	0.170	1.00	1.00
Weights ×10	0.053	0.318	0.96	0.56

When using a smaller weight for control as depicted in Fig. 16, it leads to a much higher cost for control inputs with minor changes in the tracking error. Conversely, utilizing a larger weight in a simulation as shown in Fig. 17 reduces accuracy while reducing the costs for the control inputs. The MPC controller, when operating with a smaller weight, tends to compute aggressive control inputs, which may cause a degree of oscillation in the control inputs and system.

The choice of the number of horizons and control input weights is not absolute, considering the varying objectives. However, in this work, the authors have opted to use the number of horizons

and weights adopted in the base case for the remaining studies. This decision appears to achieve a satisfactory balance between the accuracy of the simulation and the incurred control input costs.

Determination of the weights and N_p and N_c represent factors that must be tuned in the MPC controller. From these results we see that while they do have some effect, the overall effects on the result are not significant.

4.2.5 Control input constraints

An advantage of the MPC controller is its capability to consider constraints while determining an optimal control input. These constraints are typically due to mechanical or physical limitations. In the base case simulation, neither the control drum velocity nor acceleration hits the constraints as defined in Table 2. The reactivity worth of control drum is large enough to manage the power for this simulation, implying that minor and slow rotation of the drums is sufficient to accomplish the objective. In this section, the performance of the MPC controller is further analyzed by modifying the constraints.

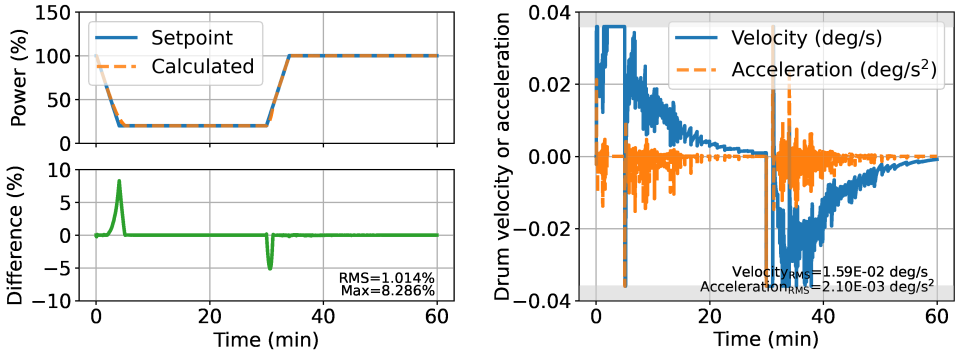


Figure 18. 2D core simulation results with 10 times smaller control input constraints

Table 9. Comparison with 10 times smaller input constraints

Description	Tracking difference (%)		Normalized control cost	
	RMS	Max	Velocity	Acceleration
Base case	0.017	0.170	1.00	1.00
Constraints ×0.1	1.014	8.286	0.78	0.41

Since the base case does not hit the constraints, only stricter or smaller constraints are tested. Fig. 18 exhibits the simulation results with constraints reduced by a factor of 10. The figure illustrates that the control drum velocity and acceleration are regulated within ±0.036 deg/s and ±0.036 deg/s^2, respectively. These stricter constraints hinder the use of quicker or more aggressive actions, thus reducing the power prediction accuracy. The maximum power tracking error is 8.286%, which is a marked increase compared to the base case results. Strict constraints tend to degrade the controller's accuracy. In such scenarios, it is preferable to utilize the control drums more to ensure sufficient reactivity worth with small and slow rotation. Results such as this demonstrate the usefulness of the simulation capability here because if better results from the controller are not achievable with the system constraints, this indicates that the controlled system should be redesigned. Further a

simulation capability can determine what controlled system design points or operating envelope should be.

4.2.6 Control drum reactivity worth

A wide range of the control drum reactivity worth is tested for the 2D microreactor load-follow simulation in this section. The reactivity worth varies from −60% to 60% compared to the reference value listed in Table 3. The tracking error and normalized control cost are summarized in Table 10. The tracking error increases as the drum worth moves away from the base case. As mentioned in Section 4.2.2, the larger drum worth (i.e. $df_{\bar{n}}/dD_1$) makes the controller undershoot the control input causing smaller control cost but larger tracking error. Conversely, as the drum worth becomes smaller, the larger control cost is limits performance of the controller. Essentially, this means the controller with a sufficiently small drum worth may overshoot the reference trajectory such that the system and control action exhibit increasingly oscillatory behavior. This situation is clearly shown in Fig. 19. The control inputs with the 60% smaller drum worth hit the constraints many times while the power and control input oscillate. If the constraints are not used, the simulation may diverge due to the aggressive control inputs. This result also indicates estimated limits for how accurate the control rod worth must be for the state-space model. We conclude that being within 30% of the actual drum worth would allow for a sufficiently performant controller.

Table 10. Comparisons with different control drum reactivity worths

Description	Tracking difference (%)		Normalized control cost	
	RMS	Max	Velocity	Acceleration
Drum worth −60%	0.106	0.790	4.90	37.87
Drum worth −30%	0.022	0.326	1.00	1.48
Base case	0.017	0.170	1.00	1.00
Drum worth +30%	0.031	0.172	1.00	0.88
Drum worth +60%	0.049	0.226	0.99	0.80
Position-dependent drum worth	0.019	0.166	1.00	1.03

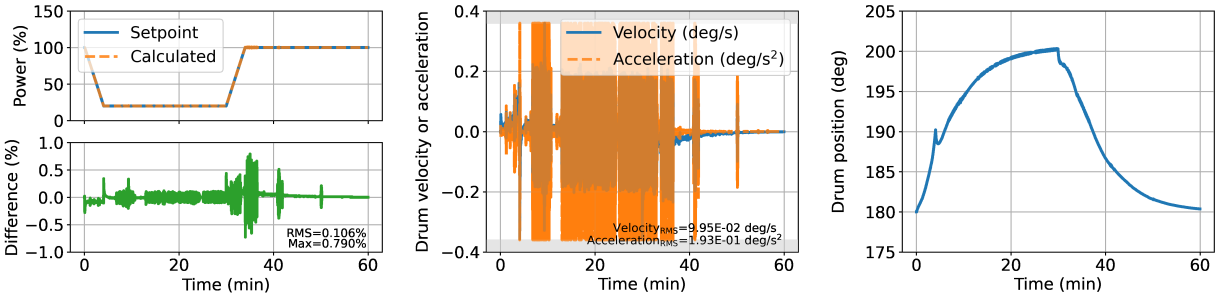


Figure 19. 2D core simulation results with 60% lower control drum reactivity worth

An additional comparison is made for the position-dependent control drum worth. As presented in Fig. 5a, the control drum worth may vary depending on the position. The results with the position-dependent drum worth is also listed in Table 10. The tracking error and the control costs are not much different from those of the base case which uses the constant control drum worth. We note that this a conclusion based on this result would be flawed. This is because the control drum

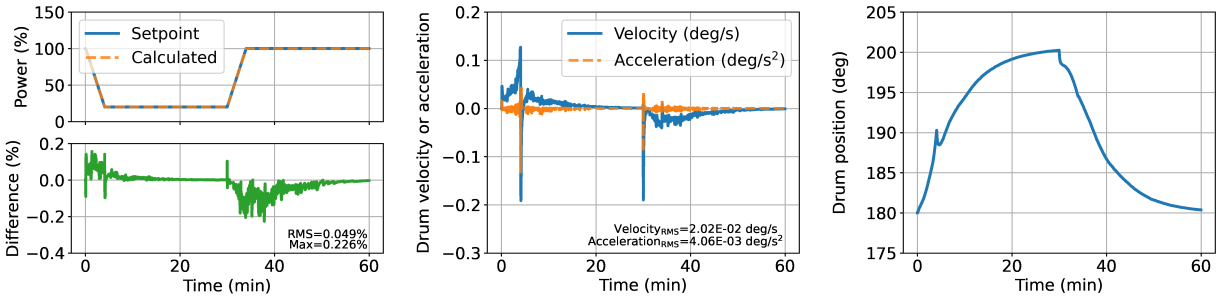


Figure 20. 2D core simulation results with 60% higher control drum reactivity worth

differential worth is fairly flat in the operation range which is 180 - 205 deg. As shown in Fig. 5a, the slope of the reactivity between the ranges is monotonic. If the control drum is initially located in a different rotation (such as 220 deg), the position-dependent worth may be essential to have accurate and stable results.

In this section, the sensitivity test results with different control drum reactivity worths are presented. The simulation results show that it is important to have an accurate estimate of the control drum worth to have accurate and stable simulation results. However, the controller with ±30% error in the control drum worth still predicts fairly accurate and stable results. The ±30% of error in the control drum worth is a quite large margin therefore the MPC controller is reasonably robust.

4.2.7 Point kinetics parameters

A range of values in the point kinetics parameters are tested in this section. The purpose of this study is to gain some insight into how accurate the coefficients in the state-space representation must be compared to the real system. As mentioned in Section 4.1, the point kinetics parameters may have some error since the parameters were not calculated from homogenization by weighting solutions from PROTEUS transport calculation. Instead, medium values or typical values were chosen for the reduced order model. In reality, these parameters can be determined during startup physics testing, but will change to some degree as the reactor operates. Therefore, in practice, it may be difficult to calculate the point kinetics parameters for every configuration and condition of the real reactor. Consequently, we assume that the point kinetics parameters used in the controller will generally have some amount of error. As a result, it is important to check how much the MPC controller is sensitive to the selection of the point kinetics parameters. Tables 11 to 13 summarize the sensitivity test results for the delayed neutron fractions, decay constants, and generation time, respectively.

Table 11. Comparisons with different delayed neutron fractions

Description	Tracking difference (%)		Normalized control cost	
	RMS	Max	Velocity	Acceleration
β_i −60%	0.046	0.375	0.99	0.62
β_i −30%	0.020	0.145	1.00	0.84
Base case	0.017	0.170	1.00	1.00
β_i +30%	0.019	0.267	1.00	1.24
β_i +60%	0.027	0.403	1.24	2.78

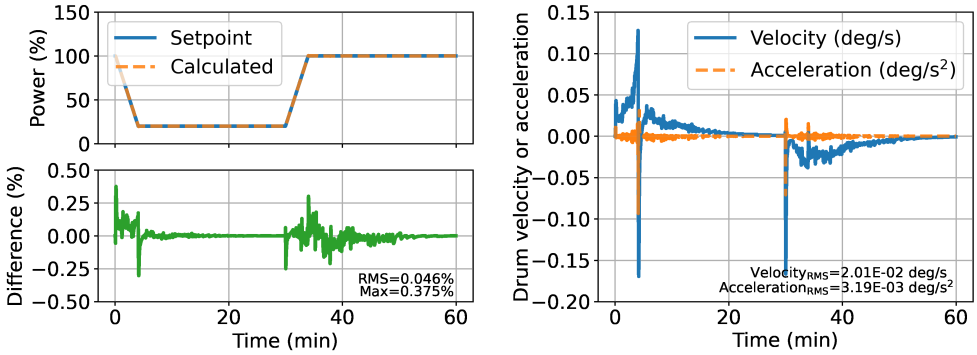


Figure 21. 2D core simulation results with 60% lower β_i

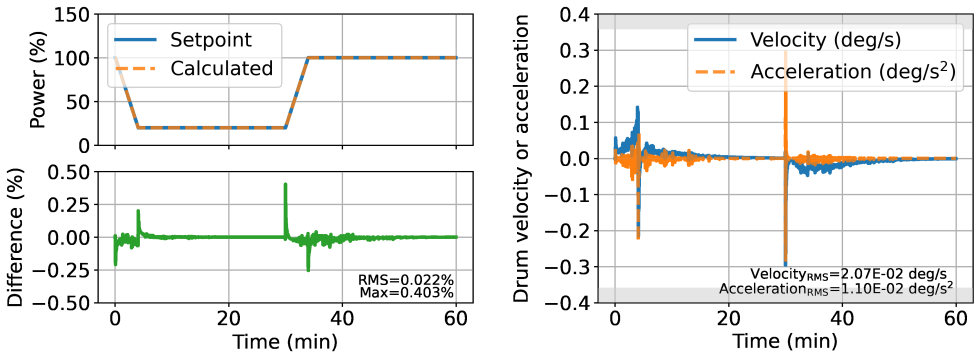


Figure 22. 2D core simulation results with 60% higher β_i

Figs. 21 and 22 present the simulation results for 60% lower and higher β_i cases, respectively. The control cost and the tracking error increase as β_i increase. This means that the reduced order model underestimates the effect of the prompt neutron so that the MPC controller overshoots the control input to compensate for the underestimated prompt neutron effect. On the contrary, the reduced order model overestimates the effect of the prompt neutron as the β_i decreases. The MPC controller undershoots the control inputs in this case. The small β_i reduces the control cost but it increases the tracking error.

Table 12. Comparisons with different decay constants

Description	Tracking difference (%)		Normalized control cost	
	RMS	Max	Velocity	Acceleration
λ_i −60%	0.032	0.224	1.03	1.36
λ_i −30%	0.021	0.176	1.01	1.11
Base case	0.017	0.170	1.00	1.00
λ_i +30%	0.016	0.165	1.00	0.94
λ_i +60%	0.017	0.160	0.99	0.88
λ_i +90%	0.019	0.155	0.99	0.84

Table 12 summarizes the sensitivity results for the decay constant. The decreased λ_i means the delayed neutron has a longer half-life so that the effect of the prompt neutron is underestimated.

The MPC controller overshoots the control input due to the underestimated prompt neutron effect. When using 30% increased λ_i, the simulation results appear to have better accuracy in terms of the RMS tracking error, and use less amount of control cost. This result is likely interpreted that increasing λ_i makes the overall reduced order model be more accurate than the equations with the base parameters since not all parameters used in the reduced order model are accurate. However, the RMS tracking error increases again if the λ_i is increased by 60% and 90%. Thus, we conclude that the 30% increase in λ_i merely provides some easier controllability in the dynamics (it becomes easier to control since the dynamics are "slower"). However, the λ_i are fairly well known, and only change by fissionable isotope, thus these parameters will generally have less error than others.

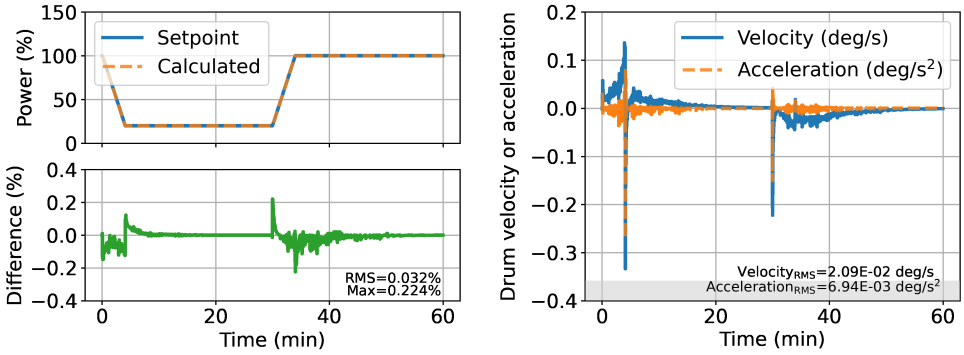


Figure 23. 2D core simulation results with 60% lower λ_i

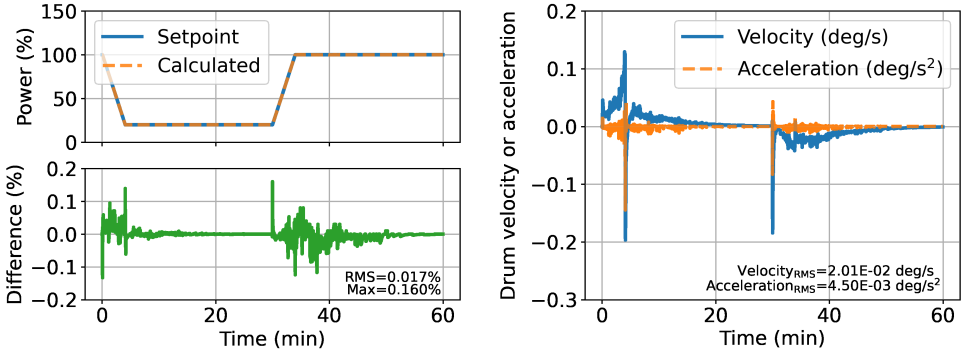


Figure 24. 2D core simulation results with 60% higher λ_i

Changing the prompt neutron generation time is observed to make little difference in the control result. The results are summarized in Table 13. There are very small difference between these simulations so that the results in the table have the same value depending on cases. The neutron time derivative in Eq. (5) shows that Λ exists at the denominator of the entire right hand side equations. Therefore, all $\partial f_{\bar{n}}/\partial x$ (i.e. $\partial f_{\bar{n}}/\partial \bar{n}$ ... $\partial f_{\bar{n}}/\partial D_1$) change with the same fraction (30% or -30%) when Λ changes in the sensitivity test. If the temperature feedback or the drum reactivity worth in Eq. (11) have more complicated form rather than the simple linear equation, the sensitivity results per Λ may be different.

Generally, we would expect the control system to be insensitive to Λ as it influences the fastest dynamics in the system and its effect decays away quickly with time. Should the reactivity insertions

Table 13. Comparisons with different generation time

Description	Tracking difference (%)		Normalized control cost	
	RMS	Max	Velocity	Acceleration
Λ −30%	0.017	0.170	1.00	1.00
Base case	0.017	0.170	1.00	1.00
Λ +30%	0.017	0.170	1.00	1.00

become much closer to \$1, this parameter would matter quite significantly, but for small reactivity insertions over long time periods, the physics of the prompt neutron generation time become "instantaneous" over a broad range of values (likely ± an order of magnitude).

4.2.8 Temperature reactivity coefficients

The sensitivity results for the temperature reactivity coefficients are summarized in Table 14. Since the reacitivty feedback effect from the temperature change is very slow compared to neutronics, all cases have almost identical control costs. As presented in Fig. 10, the temperature changes much more slowly than the power or neutron density. A large control cost is usually used when prompt action is required to reduce the power error. There is only a slight difference in the performance of the controller depending on the reactivity coefficients. When the reduced order model has smaller reactivity coefficients such as Fig. 25, the calculated power tends to change more slowly than the setpoint power. Note that the difference in the plot is calculated by *Calculated* − *Setpoint*. Due to the underestimated temperature coefficient, the required control input is also underestimated. In other words, the change in temperature introduces negative reactivity, so the control input is

Table 14. Comparisons with different temperature coefficients

Description	Tracking difference (%)		Normalized control cost	
	RMS	Max	Velocity	Acceleration
α_f, α_m −60%	0.048	0.294	1.00	1.00
α_f, α_m −30%	0.030	0.221	1.00	1.00
Base case	0.017	0.170	1.00	1.00
α_f, α_m +30%	0.019	0.170	1.00	1.00
α_f, α_m +60%	0.035	0.186	1.00	1.00

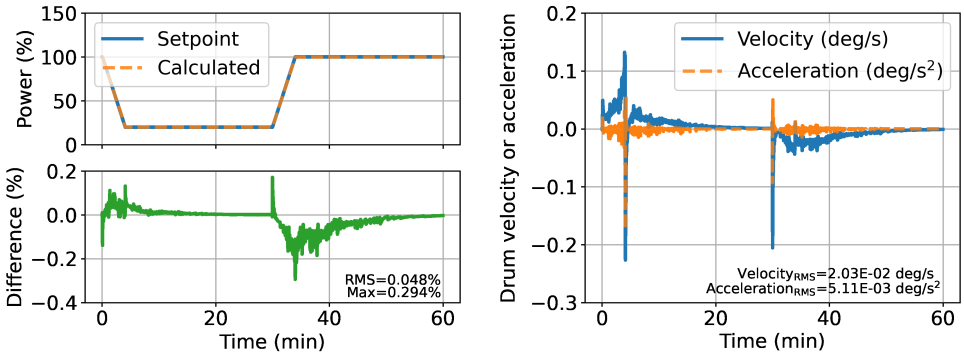


Figure 25. 2D core simulation results with 60% lower α_f and α_m

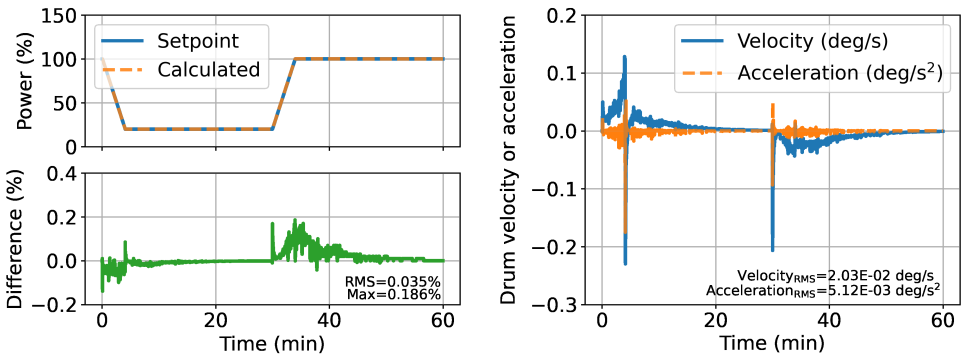


Figure 26. 2D core simulation results with 60% higher α_f and α_m

underestimated if the negative reactivity is not sufficiently accounted for. On the other hand, the required control input is overestimated when using the overestimated temperature coefficients as shown in Fig. 26. The calculated power with the 60% larger temperature coefficient tends to go ahead of the setpoint power.

In summary, the temperature coefficients have a significant margin for achieving accurate load follow results. The control cost is even less sensitive to variations in temperature coefficients, primarily because the rate of temperature change tends to be slower compared to the power.

4.2.9 Heat capacity

The sensitivity results for the heat capacity are summarized in Table 15. Changing the heat capacity or the mass leads to the same results since both heat capacity and the mass appear together on the left hand side of Eqs. (7) to (9). Therefore, additional sensitivity test on the mass is not performed. It can be assumed that changing the mass has the same effect as changing heat capacity. Similar to the sensitivity test on the temperature coefficients in Section 4.2.8, the temperature changes slower than the power or neutron density. Therefore, the control cost does not have a big impact according to the choice of the heat capacity as shown in Table 15. The error in the heat capacity primarily influences the speed of temperature change depending on the value as shown in Figs. 27 and 28. In the figures, *ROM* indicates temperatures calculated from the reduced order model. However, ±30% perturbed heat capacity or mass, again do not make the temperature feedback effect as fast as the neutronics so that tracking error does not have significant impact from the heat capacity or mass.

In other words, the mc_p terms of the thermal problem in the HTGR are so much slower than the delayed neutrons that relatively large changes in their values such as 30% are not sufficient to change

Table 15. Comparisons with different heat capacity

Description	Tracking difference (%)		Normalized control cost	
	RMS	Max	Velocity	Acceleration
$c_{p,f}, c_{p,m}, c_{p,c}$ −30%	0.020	0.171	1.00	1.00
Base case	0.017	0.170	1.00	1.00
$c_{p,f}, c_{p,m}, c_{p,c}$ +30%	0.022	0.192	1.00	1.00

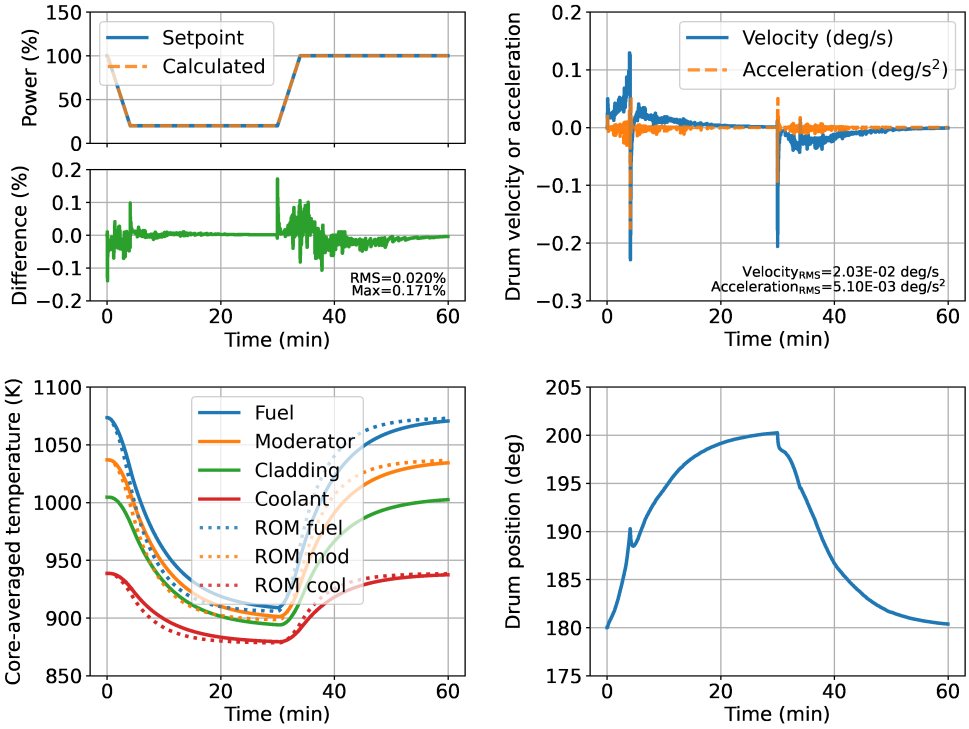


Figure 27. 2D core simulation results with 30% lower heat capacity

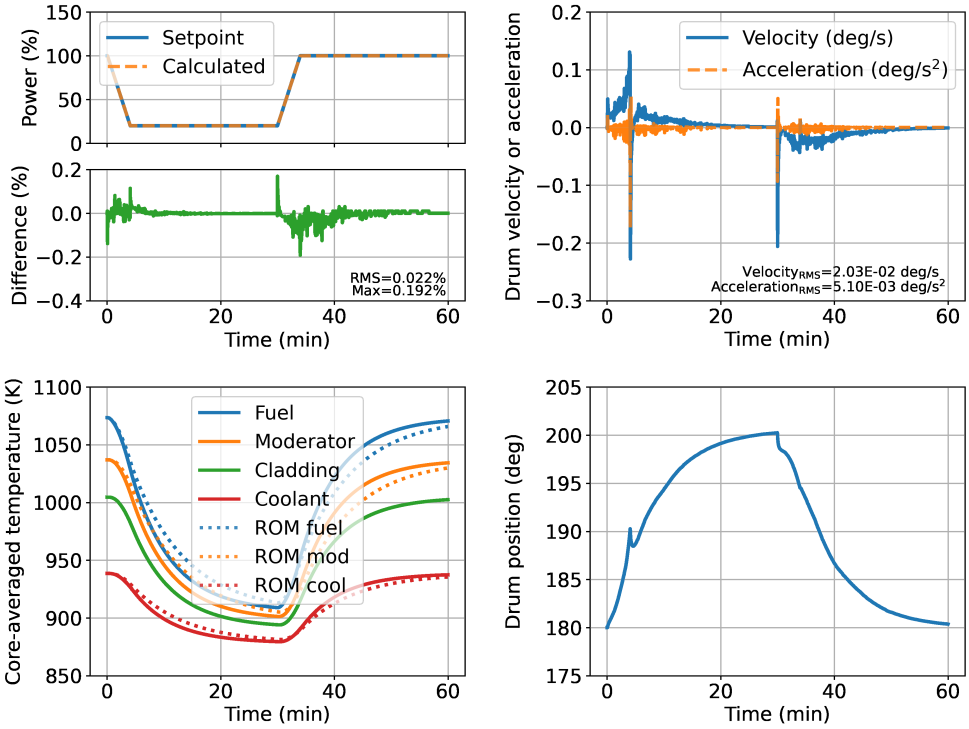


Figure 28. 2D core simulation results with 30% higher heat capacity

the time scales between the thermal dynamics and neutron dynamics in any meaningful way. 30% difference in "far away" is still "far away".

4.2.10 Ramp rate and power level

A wide range of ramp rates and different scenarios are tested in this section. It is expected that a slow transient scenario is easier to calculate the control input for than a fast transient scenario. Table 16 presents the sensitivity results with 5%/min ramp rate to 30%/min ramp rate including the base case (20%/min ramp rate). Figs. 29 to 31 show the detailed simulation results. As expected, the simulation results have a higher tracking error when using the faster ramp rates however the overall change in error is less than 0.1%. For the 5%/min of ramp rate case, the maximum tracking error is only 0.097% and the normalized control cost for the acceleration is only 32% of the base case. On the other hand, the case with 30%/min ramp rate has 0.384% of maximum tracking error and 63% higher control costs than that of the base case. From the sensitivity test on the ramp rate,

Table 16. Comparisons with different scenario

Description	Tracking difference (%)		Normalized control cost	
	RMS	Max	Velocity	Acceleration
Ramp rate 5%/min	0.012	0.097	0.61	0.32
Ramp rate 10%/min	0.014	0.112	0.75	0.54
Ramp rate 20%/min (Base case)	0.017	0.170	1.00	1.00
Ramp rate 30%/min	0.021	0.384	1.27	1.63
Power 100%→140%→100%	0.015	0.140	0.40	0.24

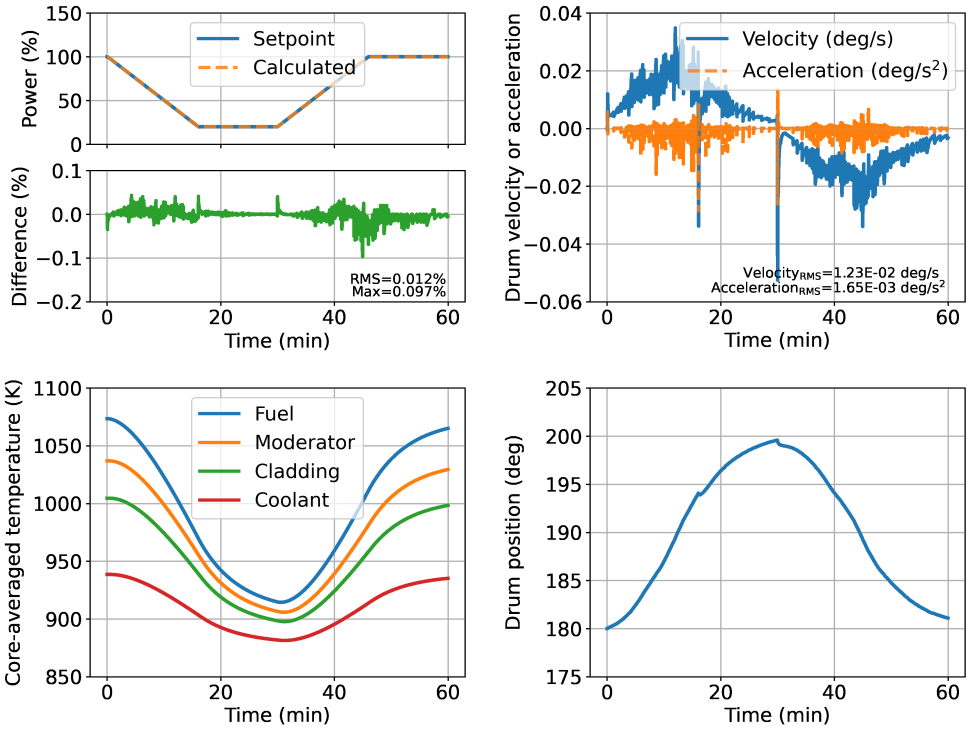


Figure 29. 2D core simulation results with 5% ramp rate

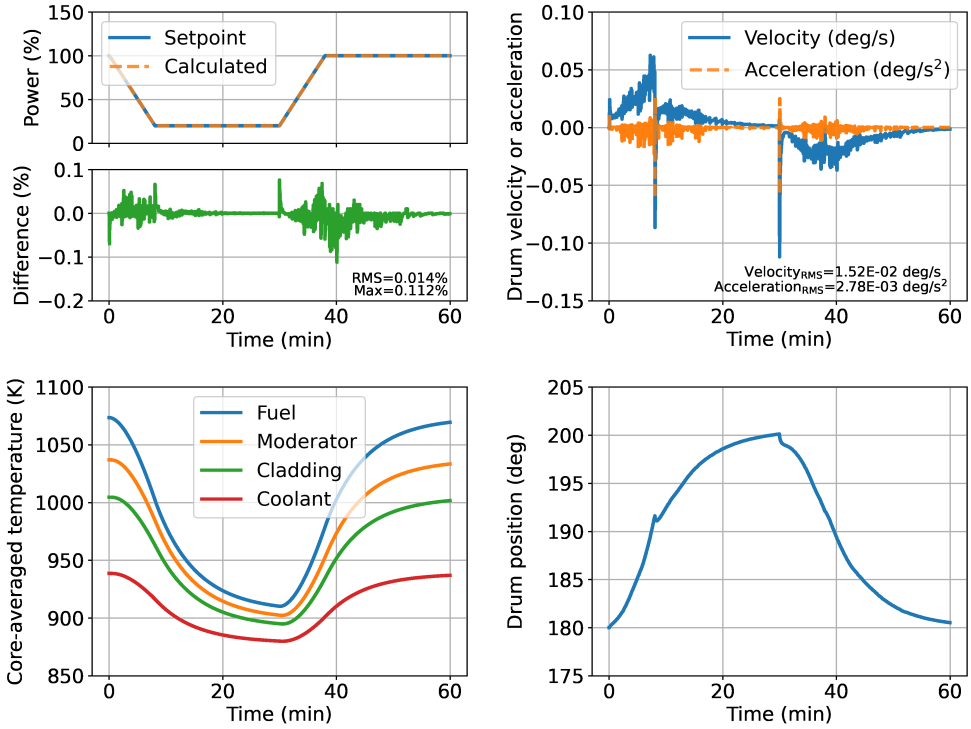


Figure 30. 2D core simulation results with 10% ramp rate

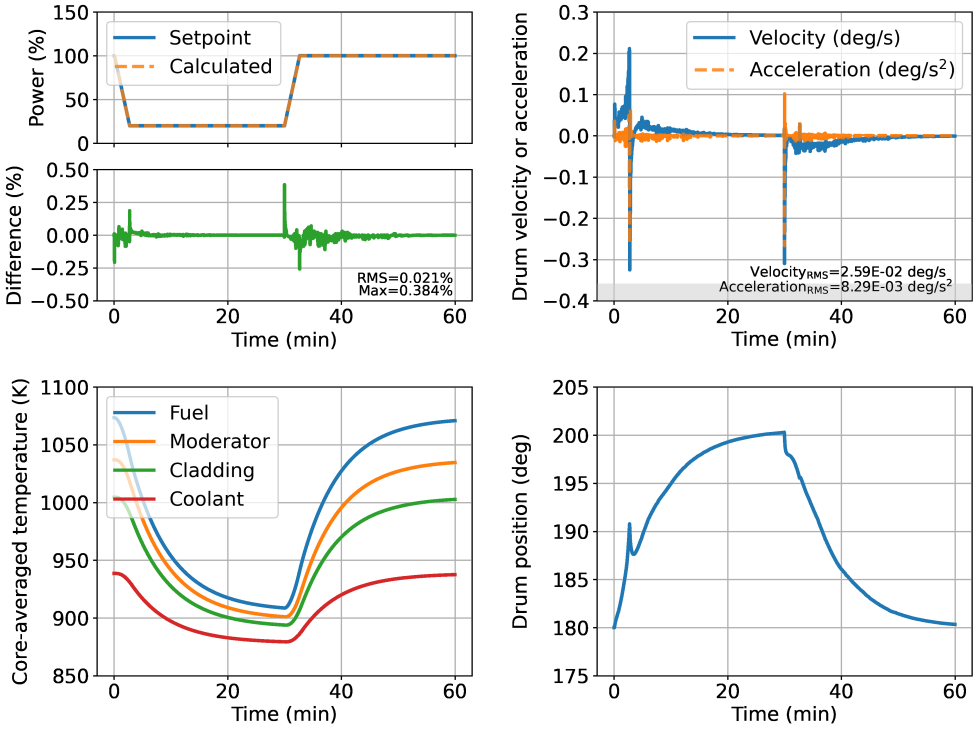


Figure 31. 2D core simulation results with 30% ramp rate

it is concluded that the MPC controller can calculate accurate control inputs to follow the setpoint for essentially the whole range of required or allowable ramp rates.

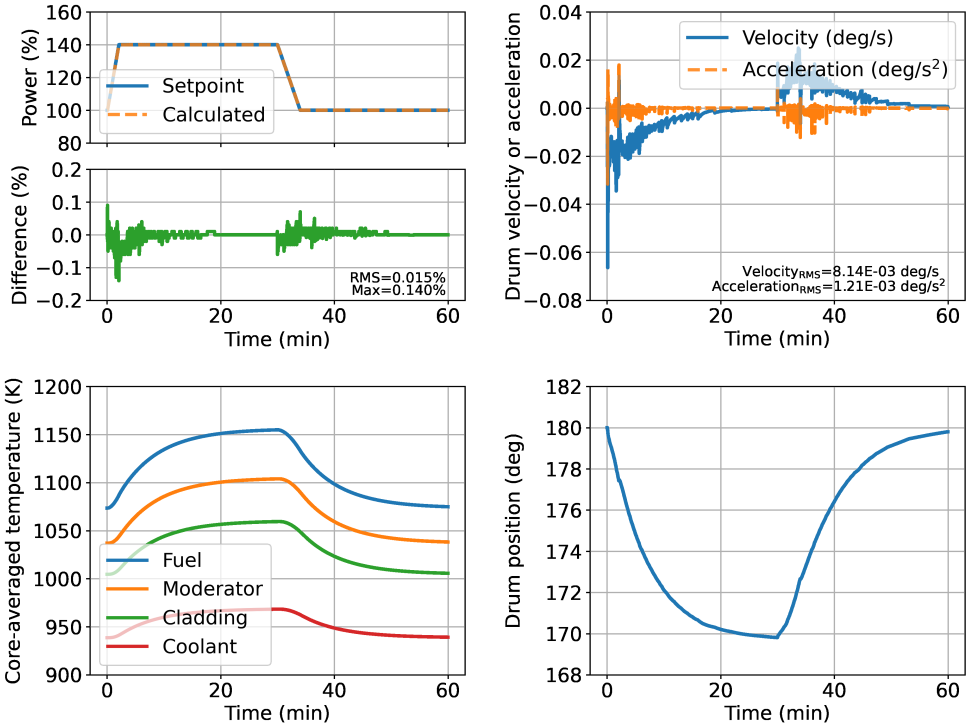


Figure 32. 2D core simulation results with maximum 140% power level

Another scenario considering an uprated power is also simulated and presented in Fig. 32. The scenario has 100% power initially, increases the power to 140%, and then comes back to the original power. This test case illustrates the potential of this analysis capability to be used to assess the control system performance for a potential future power uprate. In this simulation, it is confirmed that the code system does work properly at the higher power. The control drum is rotated out to introduce positive reactivity from the control drum in order to raise the power level. From the simulation results, it is also confirmed that the control drum decusping function works well in the other direction as well. There is no noticeable difference in the tracking error or the control cost compared to that of the base case. The simulation has accurate results and does not have a significant overhead on the control cost.

4.3 Discussion on Numerical Results

In Section Section 4, we demonstrate the load-follow simulation for a 3D microreactor similar to the HTGR. The MPC controller predicts highly accurate control inputs, thus enabling the prescribed power to be followed precisely. The RMS error is 0.027%, and the maximum error reported is 0.234% for the nominal operating case.

A potential concern with the MPC controller involves its need for a model that represents the plant to a sufficient level of accuracy. Obtaining a highly accurate reduced-order model may be impractical, given the inherent and inevitable presence of model error and uncertainty in model

parameters. Therefore, a series of sensitivity tests were performed on the MPC method and model parameters, as described in Section Section 4.2.

The sensitivity tests confirm that the MPC controller is generally robust, and allows for a considerable margin when selecting parameters for inclusion in the reduced order model. Even with an error range as much as ±60% in the point kinetics parameters, temperature reactivity coefficients, and lumped TH models, the maximum tracking error remains below 0.5%. The control drum reactivity worth exhibits the most sensitivity compared to other variables—this is expected. For instance, a simulation with a 60% decrease in control drum worth results in significant oscillation in the control inputs, and saturates the controller (inputs are limited by constraints). However, simulations with control drum worth values ranging from 30% lower to 60% higher showed accurate results and resulted in a maximum tracking error of less than 0.326%.

The sensitivity test demonstrates that overshooting in the control input can negatively impact the precision and stability of simulation. The MPC controller exceeds the control input when parameters such as underestimated control drum worth, overestimated decay constant β_i, and underestimated decay constant λ_i are used. Temperature-based parameters, like the temperature reactivity coefficient and heat capacity, could also produce similar effects. However, these parameters are less sensitive than others due to the slow rate of the temperature change relative to the power or neutron density. While undershooting the control input can diminish tracking accuracy, it does not cause significant solution oscillation.

5. CONCLUSIONS

This report presents the development and assessment of a new simulation capability that combines the MPC controller with high-fidelity simulation for autonomous load follow operation of a HTGR type microreactor. The high fidelity neutronics code, PROTEUS, is used to emulate the actual plant model, making the "plant" simulation more realistic in the absence of a physical system. The simplified TH and control drum decusping method have both been implemented in PROTEUS. A reduced order model, based on the point kinetics equation and lumped TH models, was also derived and incorporated in the MPC controller. The adaptive MPC controller is deployed for computing the control input to adhere to the given power scenario. This adaptive MPC sequentially linearizes the nonlinear reduced order model for use in the MPC optimization problem, thereby enhancing accuracy of the controller for nonlinear systems.

In the assessment of numerical results, the PROTEUS/MPC code system is evaluated for the load follow operation of a 3D microreactor with a ramp rate at 20%/min of the nominal power. The simulation reveals that the error in the tracking power is under 0.234%, while the control inputs remain within the established constraints. For further verification, extensive sensitivity tests have been performed by adjusting parameters used in the MPC controller and reduced order model. The conclusions drawn confirm that the MPC controller offers a sizable margin for adjusting parameter values, thus validating the feasibility of autonomous control with the MPC controller. For additional verification, a comprehensive series of sensitivity tests was conducted by modifying parameters within the MPC controller and the reduced order model. The results of these tests confirm that the MPC controller allows a significant range of flexibility or uncertainty in the parameters. This flexibility provides robust assurance of the autonomous control feasibility via the MPC controller.

Lastly, we identify a few ways in which such a capability may add value to reactor designers. With being able to simulate the controller comapred with high fidelity simulation, analysts can gain confidence that a particularly controller is sufficiently robust compared with best estimate simulation methods in the absense of physical systems and experimental data. Further, this capability can be used to identify design flaws for targets, particularly as it relates to the choice of control drum drive motors and reactivity worth curves. This simulation capability also presents a tool to assess system and controller performance for design or operating scenarios that may happen such as after a power uprate or some other significant design change.

Future areas of research related to this can include considering other transient scenarios, expanding the simulation model to include the balance of plant system components, development of a multiple-input, multiple-output control problem, introduction of noise, control of a different variable (e.g. core average temperature inestead of core power)

ACKNOWLEDGEMENTS

This work was supported by funding received from the DOE Office of Nuclear Energy's Nuclear Energy University Program under contract number DE-NE0008887.

REFERENCES

[1] N. Stauff, C. Lee, and C. Filippone. "Core Design of the Holos-Quad Microreactor." Technical Report ANL/NSE-22/4, Argonne National Lab.(ANL), Argonne, IL (United States) (2022).

[2] "HolosGen LLC." URL http://www.holosgen.com.

[3] S. Choi, S. Kinast, and B. Kochunas. "Point Kinetics Model Development with Predictive Control for Multi-Module HTGR Special Purpose Reactors." Technical Report NURAM-2020-006-00, University of Michigan, Ann Arbor, MI (United States) (2020).

[4] Q. Shen and B. Kochunas. "Preliminary Passive Feedback Model Development and Integration." Technical Report NURAM-2021-0004-00, University of Michigan, Ann Arbor (2021).

[5] E. Shemon, M. Smith, and C. Lee. "PROTEUS-SN methodology manual." Technical report, Argonne National Lab.(ANL), Argonne, IL (United States) (2014).

[6] C. H. Lee, Y. S. Jung, H. Connaway, and T. Taiwo. "Simulation of TREAT Cores Using High-Fidelity Neutronics Code PROTEUS." In *Proc. M&C 2017*, pp. 16–20 (2017).

[7] Y. S. Jung and W. S. Yang. "A Consistent CMFD Formulation for the Acceleration of Neutron Transport Calculations Based on the Finite Element Method." *Nuclear Science and Engineering*, **volume 185**(2), pp. 307–324 (2017).

[8] G. Yang, M. K. Jaradat, W. S. Yang, and C. H. Lee. "Development of coupled PROTEUS-NODAL and SAM code system for multiphysics analysis of molten salt reactors." *Annals of Nuclear Energy*, **volume 168**, p. 108889 (2022).

[9] B. Kochunas, B. S. Collins, D. Jabaay, K. S. Kim, A. Graham, S. Stimpson, W. A. Wieselquist, K. T. Clarno, S. Palmtag, T. Downar, et al. "VERA core simulator methodology for PWR cycle depletion." In *Proc. M&C 2015* (2015).

[10] C. E. García, D. M. Prett, and M. Morari. "Model predictive control: Theory and practice—A survey." *Automatica*, **volume 25**(3), pp. 335 – 348 (1989).

[11] L. Wang. *Model predictive control system design and implementation using MATLAB®*. Springer Science & Business Media, London, UK (2009).

[12] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd. "OSQP: An operator splitting solver for quadratic programs." *Mathematical Programming Computation*, **volume 12**(4), pp. 637–672 (2020).

[13] R. Findeisen and F. Allgöwer. "An introduction to nonlinear model predictive control." In *21st Benelux meeting on systems and control*, volume 11, pp. 119–141. The Netherlands: Technische Universiteit Eindhoven Veldhoven (2002).

[14] V. Adetola, D. DeHaan, and M. Guay. "Adaptive model predictive control for constrained nonlinear systems." *Systems & Control Letters*, **volume 58**(5), pp. 320–326 (2009).

APPENDIX A

Draft Journal Article of Report

High-Fidelity Microreactor Load Follow Simulations with Model Predictive Control

Sooyoung Choi[1], Qicang Shen[1], Chang-ho Lee[2], Claudio Filippone[3], and Brendan Kochunas[1*]

[1]Department of Nuclear Engineering and Radiological Sciences, University of Michigan
2355 Bonisteel Blvd, Ann Arbor, MI 48109, USA

[2]Nuclear Science and Engineering Division, Argonne National Laboratory
9700 S. Cass Ave, Lemont, IL 60439, USA

[3]HolosGen LLC
9207 Enterprise Ct Suite D, Manassas Park, VA 20111, USA

Abstract

This paper presents the application of a Model Predictive Control (MPC) controller for the autonomous load follow operation of a High-Temperature Gas-Cooled Reactor (HTGR) type microreactor. Utilizing the high fidelity neutronics code, PROTEUS, to serve as the actual plant model, the study enhances the reliability of the simulation. The Simplified Thermal Hydraulics/Fluids (STH) solver and control drum decusping methods have been incorporated into PROTEUS. The work also develops a reduced-order model, based on the point kinetics equation and lumped TH models, for integration into the MPC controller. An adaptive MPC is utilized to compute the control input required to adhere to a given power scenario; it enhances accuracy by sequentially linearizing the nonlinear reduced-order model for optimization. Numerical results from the PROTEUS/MPC code system for a 3D microreactor's load-follow operation at a 20% maximum ramp rate show that the tracking power error is minimal (under 0.234%), and the control inputs stay within predetermined constraints. Additional in-depth sensitivity tests on the parameters used in the MPC controller and reduced-order model verify the robustnesss and flexibility of the MPC controller.

Keywords: Model predictive control, load follow, transient, microreactor, high-fidelity simulation

I Introduction

As the global demands on nuclear energy output continue to rise, so does the necessity for more advanced, robust, and efficient methods of controlling and managing nuclear reactors. The move

*Corresponding author, e-mail: bkochuna@umich.edu

towards more autonomous control systems such as Model Predictive Control (MPC) has emerged as a key area of interest within the field of nuclear reactor operation, holding vast potential to revolutionize safety, performance and operational efficiency.

MPC, a control strategy known for efficiently handling multi-input and multi-output systems, has seen increasing application within the fields of the nuclear engineering as well as process industry due to its capacity to predict and optimize system performance. Leveraging a model of the system to predict future behavior, MPC allows for advanced planning and adjustments, thereby enhancing the safety, efficiency, and reliability of nuclear power plants operation.

Given the advantages of MPC, significant efforts have been dedicated to implementing MPC in nuclear reactor applications. Na, Shin, and Kim [1] employed the MPC methodology to design an automatic controller for thermal power control during load follow operations in Pressurized Water Reactor (PWR). Point models, specifically nonlinear point kinetics equations, and lumped thermal hydraulic balance equations, represented the reactor and its systems. In a similar vein, Wang et al. [2] utilized the MPC method for PWR core power control, demonstrating the superior performance of MPC in comparison to Proportional–Integral–Derivative (PID) controllers. Eliasi, Menhaj, and Davilu [3], on the other hand, employed a robust nonlinear MPC technique for PWR applications, incorporating additional constraints to manage Xenon oscillation within a predefined range. In relatively recent work, Naimi et al. [4] enhanced the traditional MPC's performance by integrating intelligent feedback linearization and a dynamic neural network to address the nonlinearity of reactor models.

There have been numerous applications of MPC to nuclear power plants. However, to the best of the authors' knowledge, no previous research has demonstrated a real plant model as a high-fidelity neutron transport simulation with multiphysics coupling, particularly with respect to Gen-IV microreactors. Historically, plant models have most commonly been depicted by nonlinear or linear point dynamics models.

In this research, a High-Temperature Gas-Cooled Reactor (HTGR)-type microreactor is modeled using multiphysics transient simulations. A high-fidelity neutron transport code, PROTEUS [5], is coupled to an adaptive MPC controller to achieve this. The MPC controller is employed to ascertain the optimized control drum rotation for load-following operations. The outcomes of this paper offer a more realistic and dependable depiction of load-following results.

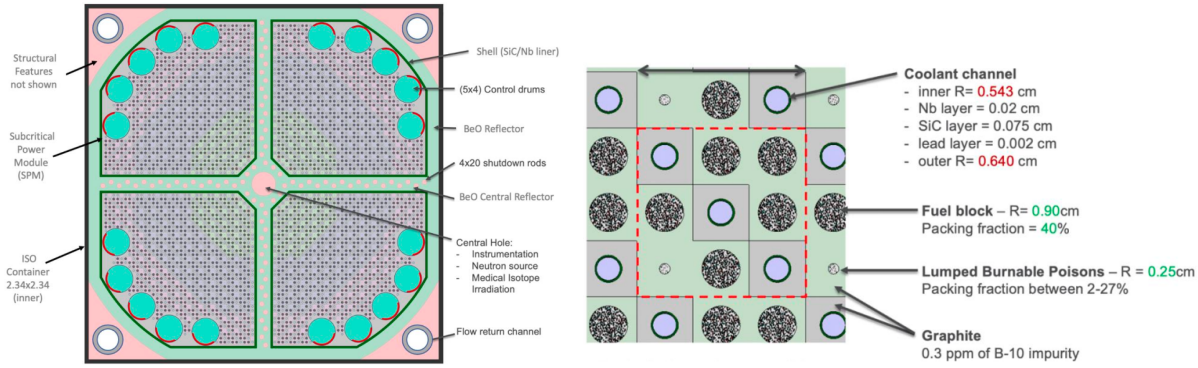
Section II presents the reactor model employed in this work, while Section III presents the methodologies implemented in PROTEUS and the MPC controller. Some of the preliminary results for load-following operations using the PROTEUS/MPC procedure can be found in Section IV. It's worth noting that this research is an extension of the authors' previous work [6].

II HTGR Model

The authors hold a particular interest in HTGR-type microreactors, with the Holos-Quad microreactor being an ideal candidate for in-depth study. A Gen-IV microreactor concept proposed by HolosGen LLC [7], the radial layout of Holos-Quad (Gen 2+) is illustrated in Fig. 1a [8].

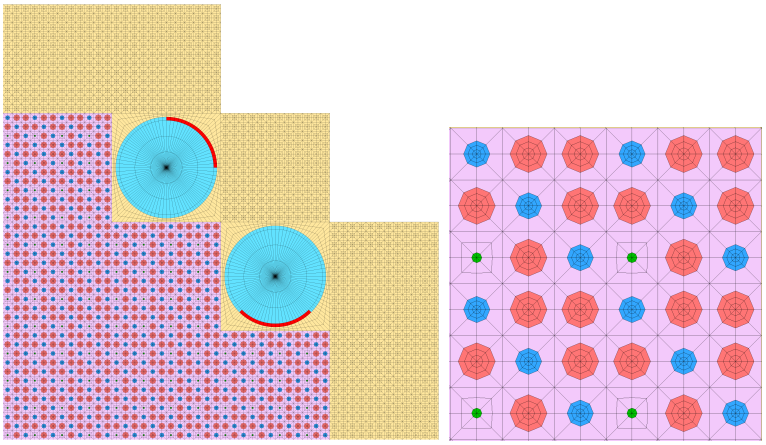
Although the design is fundamentally HTGR type, the reactor has unique features, such as Sub-

critical Power Modules (SPMs) and control drums. These rotating control drums are strategically located in the radial reflector region and are employed for load-follow operation. As depicted in Fig. 1b, a unit-cell is defined as a 3x3 pin array. Each unit-cell contains five fuel compacts, three helium coolant channels, and one burnable absorber.



(a) Radial layout (b) Unit-cell

Figure 1: Holos-Quad (Gen 2+) model [8]



(a) Radial layout (b) Unit-cell

Figure 2: Simplified microreactor model

Table 1: Comparison between Holos-Quad and simplified model

Parameter	**Holos-Quad (Gen 2+)**	**Simplified Microreactor**
Power (MW)	22.00	2.42
# of fuel compacts	2300	480
Active core height (cm)	380	200
Power density (W/m)	2517	2517
# of coolant channels	1528	288
Core coolant mass flow rate (g/s)	21896	3085
Inlet temperature (K)	863	863
Estimated outlet temperature (K)	1123	1014

While the Holos-Quad microreactor is suitable for the focus of this research, its size combined with a long-running transient presents a notable challenge for computational resources. Consequently, we have decided to modify the Holos-Quad model into a simplified, yet representative model.

The modified microreactor model and its unit-cell model are visually represented in Figs. 2a and 2b. In Table 1, we compare some design parameters between the Holos-Quad and the simplified microreactor.

The restructured model presumes quarter symmetry and is smaller in size than the Holos-Quad while maintaining the linear power density. The control drum has a 90-degree arc of absorber material. Although there are two control drums in the quarter core, only one will be operational in this work to facilitate problem simplification. Stated differently, one control drum's absorber faces the core peripheral region, while the other control drum can rotate as needed to reach the desired power level.

III Methodology

The objective of our study is to develop a control strategy and demonstrate the load-follow operation performance of the Holos-Quad microreactor design. To accomplish this, we employ the PROTEUS code for high-fidelity, multiphysics calculations in order to best represent an actual plant operation. For the load follow operation, the MPC controller calculates the control input, namely the control drum rotation, for every time step. The subsequent subsections provide detailed descriptions of the individual components incorporated in the calculation.

III.A Plant Model – PROTEUS

III.A.1 Neutronics solver

The PROTEUS code [5] developed by Argonne National Laboratory (ANL) is a high-fidelity neutron transport code based on unstructured finite element meshes that allow users to model complex and unconventional geometry in reactor problems like microreactors and Gen-IV reactors. PROTEUS has several methods for the transport calculation. In this work the MOCEX solver, which uses 2D MOC radially and the discontinuous Galerkin finite element method axially with extruded geometry [5], is used. The transient fixed source problem solver is used by relying on the isotropic approximation of the angular flux time derivative. The transport solution is accelerated by the consistent Coarse Mesh Finite Difference (CMFD) [9]. The study uses a 6 energy group macroscopic cross-section library, formulated in ISOTXS format, and was generated from a Serpent Monte Carlo simulation.

III.A.1 Thermal Hydraulics/Fluids feedback solver for HTGR

To ensure reliable load-follow simulation, it is important to accurately incorporate Thermal Hydraulics/Fluids (TH) feedback. A prior research study coupled PROTEUS and Systems Analysis

Module (SAM) using a MOOSE-based wrapper for a molten salt reactor [10]. However, coupling PROTEUS and SAM requires significant effort and computational time, hence this study implements Simplified Thermal Hydraulics/Fluids (STH) solver for HTGR

The STH solver divides the system into four regions - fuel compact, graphite moderator, cladding, and helium coolant. The coolant channels in each unit cell, as presented in Figs. 1b and 2b, are consolidated into a single coolant channel. The solver assumes single-phase flow in an axial flow channel subject to constant pressure. The governing equation for one coolant channel is as follows:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho v}{\partial z} = 0 \,, \tag{1}$$

$$\frac{\partial \rho h}{\partial t} + \frac{\partial \rho h v}{\partial z} = q_f \frac{n_f}{n_c} + q_c = q \,, \tag{2}$$

where ρ is the density; v is the velocity; t is the time; z is the axial coordinate; h is the enthalpy. q_f is the power density per fuel compact; n_f is the number of fuel compact per unit cell; n_c is the number of coolant channel per unit cell; q_c is the heat deposited in the coolant; and q is the total heat source.

The fuel compact, moderator, and cladding in each unit cell are individually homogenized and made to cylindrical geometry such as Fig. 3.

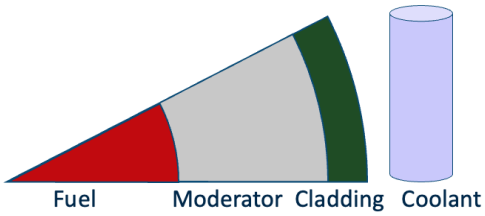


Figure 3: STH solver cylindrical geometry

The STH solver solves the radial heat transfer equation in the 1D cylindrical coordinate:

$$\rho c_p \frac{\partial T}{\partial t} = \frac{\partial}{\partial x} k(T) \frac{\partial T}{\partial x} + q \,, \tag{3}$$

where c_p is the heat capacity; T is the temperature; and k is the thermal conductivity. The boundary condition is applied at the cladding wall as follows:

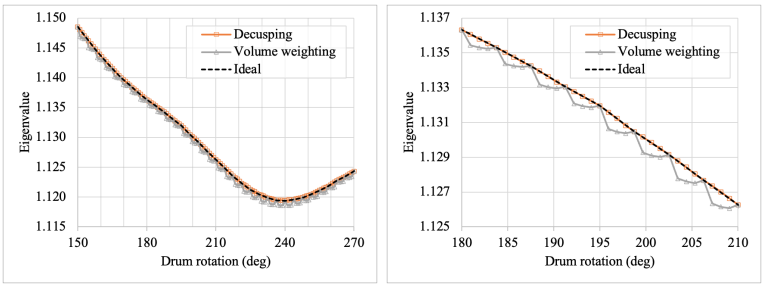
$$q'' = -k \frac{\partial T}{\partial x}\bigg|_w = h_w (T_w - T_b) \tag{4}$$

where q'' is the heat flux; h_w is the heat transfer coefficient at the surface; T_w is the wall temperature; and T_b is the coolant temperature. Eq. (3) is defined in each region of the geometry so that a system

of equations is made for the geometry. The system of equations is solved with the finite difference method. Since the unit cell geometry is approximated to the cylindrical coordinate, the accuracy may be degraded. The thermal conductivity and heat transfer coefficient can be tuned to preserve the desired temperature of each region at the steady-state condition. The desired temperature may come from a solver using a higher-order method as the finite element.

III.A.1 Control drum modeling

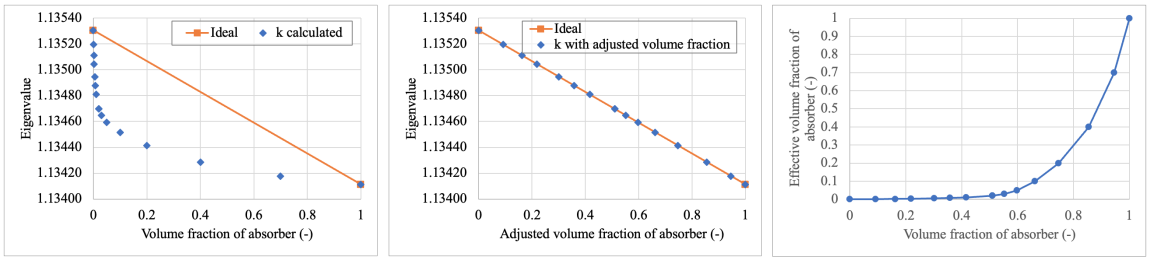
The thermal power level of a microreactor is controlled by the control drums that may rotate continuously in space. In the computational model, the mesh is finite, and it is generally quite difficult to change the mesh during the transient simulation. Therefore, we calculate an effective material or cross section for the mesh that is partially spanned by the absorber material. Simply homogenizing it based on volume fraction may lead to significant error and cause the so-called control rod or drum cusping issues. To mitigate the drum cusping issue, the control drum was divided into 96 sectors, however, this discretization was still not sufficient to completely eliminate the cusping effect. As shown in the Fig. 4, the volume weighting method causes a "cusping" effect in the eigenvalue as a function of the control drum rotation. The drum cusping behavior does not represent a realistic situation. Futhermore, it makes the reactivity control difficult since the differential reactivity worth does not have a monotonic slope.



(a) Eigenvalue from 150 to 270 deg

(b) Eigenvalue from 180 to 210 deg (Zoomed in)

Figure 4: Eigenvalue as a function of control drum rotation with different methods



(a) Eigenvalue vs. volume fraction of absorber

(b) Eigenvalue vs. adjusted volume fraction of absorber

(c) Control drum decusping function

Figure 5: Control drum decusping function

To reduce the cusping effect, a homogenization correction factor has been generated and used in this

work. The way to generate the correction factor is similar to that in [11], but with consideration of the control drum geometry. First, 15 eigenvalue calculations were performed with different arc lengths of control drum absorbers from 90 deg to 93.75 deg. In this calculation, the starting point of the absorber was fixed and only the length was changed. The volume weighting homogenization was used when calculating cross section of the mesh which is partially spanned by the absorber. Recall that the control drum geometry has 96 azimuthal sectors so each sector is 3.75 deg. Fig. 5a shows the eigenvalue as a function of the volume fraction. In the figure, a 0 of volume fraction means the length of absorber is 90 deg while 1 means the length of absorber is 93.75 deg. Based on the assumption that the eigenvalue should vary linearly, the adjusted volume fractions were calculated as Fig. 5b. Additional eigenvalue calculations are not necessary to calculate the volume adjustment fraction. Instead, the x-axis was shifted to locate eigenvalues on the ideal line. Fig. 5c shows the adjusted or effective volume fraction as a function of the model volume fraction. This function is used to calculate the effective volume fraction when absorber material spans a mesh partially. Even though the calculation is based on perturbing the volume fraction of one side of the absorber, the function is used for both sides of the absorber. With this drum decusping function, the eigenvalue changes smoothly as shown in Fig. 4. The function was made based on series of 2D core calculations, however, it turned out that the same function behaved well for a 3D core simulation as well. This approach has limitations in that this function is not general, nevertheless the procedure may be repeated for any particular reactor configuration. A more general method may need additional research but the decusping function behaves well for the microreactor.

III.B Reduced Order Models and MPC Controller

This section describes the methodology of the controller. In this work, the MPC is used for the controller. Since the MPC relies on the mathematical model, it is necessary to set up reasonable models. Section III.B.2 describes reduced order models for neutronics and TH feedback. Section III.B.2 describe how to define the state-space equation from the reduced order models. Section III.B.2 describes the theroy of MPC.

III.B.2 Reduced order models

The neutron kinetics is represented by the point kinetics equations with normalized forms:

$$\frac{d\bar{n}(t)}{dt} = \frac{\rho(t) - \beta_t}{\Lambda}\bar{n}(t) + \frac{\sum_{i=1}^{6}\lambda_i\bar{c}_i(t)}{\Lambda} \,, \tag{5}$$

$$\frac{d\bar{c}_i(t)}{dt} = \lambda_i\bar{n}(t) - \lambda_i\bar{c}_i(t) \,, \quad i = 1\dots6 \,, \tag{6}$$

where $\bar{n}(t)$ is the normalized neutron density at time t; $\rho(t)$ is the reactivity; β_t is the total delayed neutron fraction; Λ is the neutron generation time; λ_i is the delayed neutron time constant for i-th group; and $\bar{c}_i(t)$ is the normalized precursor density for i-th group.

Since the plant model represented by PROTEUS has the TH feedback, it is better to have the lumped TH feedback model for a reduced order model for more accurate MPC prediction. The lumped TH model has three temperatures for the fuel, graphite moderator, and helium coolant:

$$m_f c_{p,f} \frac{dT_f(t)}{dt} = q_f P_r \bar{n}(t) - K_{f\to m}\left(T_f(t) - T_m(t)\right) \ , \tag{7}$$

$$m_m c_{p,m} \frac{dT_m(t)}{dt} = (1-q_f) P_r \bar{n}(t) + K_{f\to m}\left(T_f(t) - T_m(t)\right) - K_{m\to c}\left(T_m(t) - T_c(t)\right) \ , \tag{8}$$

$$m_c c_{p,c} \frac{dT_c(t)}{dt} = K_{m\to c}\left(T_m(t) - T_c(t)\right) - K_c\left(T_c(t) - T_{in}\right) \ , \tag{9}$$

where subscripts f, m, and c denote the fuel, moderator, and coolant, respectively; m_x is the mass of material x; $c_{p,x}$ is the heat capacity; $T_x(t)$ is the temperature; q_f is the fraction of heat generated from fuel; P_r is the rated power; $K_{x\to y}$ is the heat transfer coefficient from material x to y; and T_{in} is the inlet coolant temperature.

The rotation of the control drum is also one of the state variables. The following is the equation for the control drum:

$$\frac{dD_1(t)}{dt} = V_1(t) \ , \tag{10}$$

where $D_1(t)$ is the position of control drum; and $V_1(t)$ is the velocity of the drum. The controller determines the velocity.

The reactivity model has the reactivity feedback from temperature, and the reactivity change from control drum:

$$\rho(t) = \rho_b + \alpha_f(t) T_f(t) + \alpha_m(t) T_m(t) + \alpha_c(t) T_c(t) + W_1(t) D_1(t) \ , \tag{11}$$

where ρ_b is the base reactivity; $\alpha_x(t)$ is the temperature coefficient of the material x; and $W_1(t)$ is the differential reactivity worth of the control drum.

These equations for the reduced order models can be written in a system of equations as follows:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \ , \tag{12}$$

where $\mathbf{x}(t)$ and $\mathbf{u}(t)$ are the state vector and input vector, respectively. $\mathbf{x}(t)$ contains the state variables introduced above, and it is defined as follows:

$$\begin{aligned} \mathbf{x}(t) &= \begin{bmatrix} x_1 & \dots & x_{N_x} \end{bmatrix}^T \\ &= \begin{bmatrix} \bar{n}(t) & \bar{c}_1(t) & \dots & \bar{c}_6(t) & T_f(t) & T_m(t) & T_c(t) & D_1(t) \end{bmatrix}^T \ , \end{aligned} \tag{13}$$

where N_x is the number of state variables.

For simplicity, we assume a single input case for the system therefore N_u is 1 here. With this assumption $\mathbf{u}(t)$ is

$$\mathbf{u}(t) = \begin{bmatrix} u_1 & \dots & u_{N_u} \end{bmatrix}^T = \begin{bmatrix} V_1(t) \end{bmatrix}^T \ . \tag{14}$$

The system of equations, $\mathbf{f}$, is defined as follows:

$$\begin{aligned} \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) &= \begin{bmatrix} f_1 & \dots & f_{N_x} \end{bmatrix}^T \\ &= \begin{bmatrix} f_{\bar{n}} & f_{\bar{c}_1} & \dots & f_{\bar{c}_6} & f_{T_f} & f_{T_m} & f_{T_c} & f_{D_1} \end{bmatrix}^T \\ &= \begin{bmatrix} \frac{d\bar{n}(t)}{dt} & \frac{d\bar{c}_1(t)}{dt} & \dots & \frac{d\bar{c}_6(t)}{dt} & \frac{dT_f(t)}{dt} & \frac{dT_m(t)}{dt} & \frac{dT_c(t)}{dt} & \frac{dD_1(t)}{dt} \end{bmatrix}^T \end{aligned} \ . \tag{15}$$

This form may be helpful for the linearization of the equations.

For extended periods of load-follow operation, the impact of the Xenon effect is crucial. Xenon, which builds up over several hours, carries substantial reactivity worth [12]. Regrettably, PROTEUS does not possess functionality that allows modeling of Xenon for transient simulations. Consequently, it was not possible to include the Xenon model in the reduced order model either.

III.B.2 State-space representation

State space representation is a mathematical model of a physical system expressed as a function of input, output, and state variables related by first-order differential equations or difference equations. All system variables may be represented by a linear combination of the state variable and system inputs by the ordinary differential equations.

It is useful to define the state-space model based on the linearization of the nonlinear equation. The state-space model is used in the control theory. The state-space model is defined as follows:

$$\begin{aligned} \dot{\mathbf{x}}_s(t) &= \mathbf{A}_s \mathbf{x}_s(t) + \mathbf{B}_s \mathbf{u}_s(t) \\ \mathbf{y}_s(t) &= \mathbf{C}_s \mathbf{x}_s(t) \ , \end{aligned} \tag{16}$$

where $\mathbf{A}_s$ is the system matrix with dimensions $N_x \times N_x$; $\mathbf{B}_s$ is the input matrix with dimensions $N_x \times N_u$; and $\mathbf{C}_s$ is the output matrix with dimensions $N_y \times N_x$. $\mathbf{x}_s(t)$, $\mathbf{u}_s(t)$, and $\mathbf{y}_s(t)$ are the state vector, input vector, and output vector of state-space model, respectively. The column vectors, $\mathbf{x}_s(t)$, $\mathbf{u}_s(t)$, and $\mathbf{y}_s(t)$ have length N_x, N_u, and N_y, respectively.

In the state-space representation, it is inherently assumed that $\mathbf{x}_s(t)$, $\mathbf{u}_s(t)$, and $\mathbf{y}_s(t)$ are the differences from nominal condition where the linearization is done. In other words, the vectors are defined as follow:

$$\begin{aligned} \mathbf{x}_s(t) &= \mathbf{x}(t) - \mathbf{x}(t_n) \\ \mathbf{u}_s(t) &= \mathbf{u}(t) - \mathbf{u}(t_n) \\ \mathbf{y}_s(t) &= \mathbf{y}(t) - \mathbf{y}(t_n) \ , \end{aligned} \tag{17}$$

where t_n is the time at the nominal condition where the linearization is done.

An arbitrary function $g(z)$ can be linearly approximated at a nominal state z_n as follows:

$$g(z) - g(z_n) \approx \frac{dg(z_n)}{dz}(z - z_n) \ . \tag{18}$$

Similarly, Eq. (12) is approximated at nominal time t_n as follows:

$$\dot{\mathbf{x}}(t) - \dot{\mathbf{x}}(t_n) = \mathbf{A}_s(\mathbf{x}(t) - \mathbf{x}(t_n)) + \mathbf{B}_s(\mathbf{u}(t) - \mathbf{u}(t_n)) \ . \tag{19}$$

where

$$\mathbf{A}_s = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{t=t_n} = \left.\begin{bmatrix} \frac{\partial \mathbf{f}}{\partial x_1} & \cdots & \frac{\partial \mathbf{f}}{\partial x_{N_x}} \end{bmatrix}\right|_{t=t_n} = \left.\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_{N_x}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{N_x}}{\partial x_1} & \cdots & \frac{\partial f_{N_x}}{\partial x_{N_x}} \end{bmatrix}\right|_{t=t_n} , \tag{20}$$

$$\mathbf{B}_s = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{t=t_n} = \left.\begin{bmatrix} \frac{\partial \mathbf{f}}{\partial u_1} & \cdots & \frac{\partial \mathbf{f}}{\partial u_{N_u}} \end{bmatrix}\right|_{t=t_n} = \left.\begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \cdots & \frac{\partial f_1}{\partial u_{N_u}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{N_x}}{\partial u_1} & \cdots & \frac{\partial f_{N_x}}{\partial u_{N_u}} \end{bmatrix}\right|_{t=t_n} . \tag{21}$$

Eq. (19) is the same form as Eq. (16). In many cases, t_n is time at an initial steady-state, namely $t_n = 0$. The linearization process can be done either by the analytical method or the numerical method. In the following sections, the state-space model generated at the initial steady-state condition is used for the controller algorithm.

III.B.2 Model Predictive Control

The state-space control problem is solved using the MPC algorithm, a sophisticated technique for process control that complies with a set of constraints [13]. The technique is based on an iterative optimization over a finite horizon for the system, essentially a trajectory optimization. For the purposes of this research, we concentrate on the linear, time-invariant state-space model. To minimize the discrepancy between a desired set-point and the predicted output, the control input is computed for a comparatively short time horizon by evaluating a cost function. This computation is consequently repeated at each ensuing time window.

For completeness, the basic theory of the MPC method is detailed below. The subsequent derivation of the MPC is primarily referenced from [14], presented here to ensure thorough understanding within the context of this work. The state-space model is outlined as follows, featuring a time-discretized linear system.

$$\mathbf{x}_d(k+1) = \mathbf{A}_d\mathbf{x}_d(k) + \mathbf{B}_d\mathbf{u}_d(k), \tag{22}$$

$$\mathbf{y}_d(k) = \mathbf{C}_d \mathbf{x}_d(k) \ , \tag{23}$$

where k is the time step index, and $\mathbf{A}_d$, $\mathbf{B}_d$, and $\mathbf{C}_d$ are the discrete forms of $\mathbf{A}_s$, $\mathbf{B}_s$, and $\mathbf{C}_s$, respectively. These discretized matrices are defined with Tayler expansions as follow:

$$\mathbf{A}_d = \sum_{l=0}^{\infty} \frac{1}{l!} \left(\mathbf{A}_s \Delta t\right)^l \ , \tag{24}$$

$$\mathbf{B}_d = \left[\sum_{l=1}^{\infty} \frac{1}{l!} \mathbf{A}_s^{l-1} \Delta t^l\right] \mathbf{B}_s \ , \tag{25}$$

$$\mathbf{C}_d = \mathbf{C}_s \ , \tag{26}$$

where Δt is the time difference between step k and $k+1$.

Applying a finite difference approximation to Eq. (23) yields

$$\mathbf{x}_d(k+1) - \mathbf{x}_d(k) = \mathbf{A}_d \left(\mathbf{x}_d(k) - \mathbf{x}_d(k-1)\right) + \mathbf{B}_d \left(\mathbf{u}(k) - \mathbf{u}(k-1)\right) . \tag{27}$$

Next, we make use of the following simplifying notation

$$\Delta(\cdot)(k) = (\cdot)(k) - (\cdot)(k-1) \ , \tag{28}$$

to yield

$$\Delta \mathbf{x}_d(k+1) = \mathbf{A}_d \Delta \mathbf{x}_d(k) + \mathbf{B}_d \Delta \mathbf{u}(k) \ . \tag{29}$$

Note that the input to the state-space model is $\Delta \mathbf{u}_d(k)$. The next step is to connect $\Delta \mathbf{x}_d(k)$ to the output, $\mathbf{y}_d(k)$. To do so, a new state variable vector is defined as

$$\mathbf{x}_a(k) = \begin{bmatrix} \Delta \mathbf{x}_d(k)^T & \mathbf{y}(k) \end{bmatrix}^T \ . \tag{30}$$

Note that

$$\begin{aligned} \mathbf{y}(k+1) - \mathbf{y}(k) &= \mathbf{C}_d \left(\Delta \mathbf{x}_d(k+1)\right) \\ &= \mathbf{C}_d \mathbf{A}_d \Delta \mathbf{x}_d(k) + \mathbf{C}_d \mathbf{B}_d \Delta \mathbf{u}(k) \ . \end{aligned} \tag{31}$$

Combining Eqs. (29) and (31) leads to the following discretized state-space model:

$$\overbrace{\begin{bmatrix} \Delta \mathbf{x}_d(k+1) \\ \mathbf{y}_d(k+1) \end{bmatrix}}^{\mathbf{x}_a(k+1)} = \overbrace{\begin{bmatrix} \mathbf{A}_d & \mathbf{O}_{N_x \times N_o} \\ \mathbf{C}_d \mathbf{A}_d & \mathbf{I}_{N_o \times N_o} \end{bmatrix}}^{\mathbf{A}_a} \overbrace{\begin{bmatrix} \Delta \mathbf{x}_d(k) \\ \mathbf{y}_d(k) \end{bmatrix}}^{\mathbf{x}_a(k)} + \overbrace{\begin{bmatrix} \mathbf{B}_d \\ \mathbf{C}_d \mathbf{B}_d \end{bmatrix}}^{\mathbf{B}_a} \overbrace{\Delta \mathbf{u}_d(k),}^{\Delta \mathbf{u}_a(k)} \tag{32}$$

$$\mathbf{y}_a(k) = \overbrace{\begin{bmatrix} \mathbf{O}^T & \mathbf{I}_o \end{bmatrix}}^{\mathbf{C}_a} \overbrace{\begin{bmatrix} \Delta \mathbf{x}_d(k) \\ \mathbf{y}_d(k) \end{bmatrix}}^{\mathbf{x}_a(k)} \tag{33}$$

where $\mathbf{O}_{N_x \times N_o}$ is the zero matrix with dimension $N_x \times N_o$; and $\mathbf{I}_{N_o \times N_o}$ is the identity matrix with dimension $N_o \times N_o$. $\mathbf{A}_a$, $\mathbf{B}_a$, and $\mathbf{C}_a$ are called the augmented model which will be used in the design of the predictive control.

Assuming that at the sampling instant k, where $k > 0$, the state variable vector $\mathbf{x}(k)$ is available through measurement, then the state $\mathbf{x}(k)$ provides the current plant information. The future control trajectory is denoted by

$$\Delta\mathbf{u}_a(k),\ \Delta\mathbf{u}_a(k+1)\ \ldots\ \Delta\mathbf{u}_a(k+N_c-1), \tag{34}$$

where N_c is called the control horizon – dictating the number of parameters used to capture the future control trajectory. With the information given in $\mathbf{x}(k)$, the future state variables are predicted for N_p number of samples, where N_p is called the prediction horizon. N_p is also the length of the optimization window. We denote the future state variables as

$$\mathbf{x}_a(k|k),\ \mathbf{x}_a(k+1|k)\ \ldots\ \mathbf{x}_a(k+N_p|k), \tag{35}$$

where $\mathbf{x}(k+m|k)$ is the predicted state variable at $k+m$ with given current plant information $\mathbf{x}(k)$. The control horizon N_c is chosen to be less than (or equal to) the prediction horizon N_p.

The state-space model for the prediction horizons can be denoted as follows:

$$\mathbf{Y} = \mathbf{F}\mathbf{x}_a(k) + \mathbf{\Phi}\Delta\mathbf{U}, \tag{36}$$

where

$$\mathbf{Y} = \begin{bmatrix}\mathbf{y}_a(k+1|k)^T & \ldots & \mathbf{y}_a(k+N_p|k)^T\end{bmatrix}^T\ , \tag{37}$$

$$\Delta\mathbf{U} = \begin{bmatrix}\Delta\mathbf{u}_a(k)^T & \ldots & \Delta\mathbf{u}_a(k+N_c-1)^T\end{bmatrix}^T\ , \tag{38}$$

$$\mathbf{F} = \begin{bmatrix}\mathbf{C}_a\mathbf{A}_a \\ \vdots \\ \mathbf{C}_a\mathbf{A}_a^{N_p}\end{bmatrix}\ , \tag{39}$$

$$\mathbf{\Phi} = \begin{bmatrix} \mathbf{C}_a\mathbf{B}_a & 0 & 0 & \cdots & 0 \\ \mathbf{C}_a\mathbf{A}_a\mathbf{B}_a & \mathbf{C}_a\mathbf{B}_a & 0 & \cdots & 0 \\ \mathbf{C}_a\mathbf{A}_a^2\mathbf{B}_a & \mathbf{C}_a\mathbf{A}_a\mathbf{B}_a & \mathbf{C}_a\mathbf{B}_a & \cdots & 0 \\ \vdots & & & & \\ \mathbf{C}_a\mathbf{A}_a^{N_p-1}\mathbf{B}_a & \mathbf{C}_a\mathbf{A}_a^{N_p-2}\mathbf{B}_a & \mathbf{C}_a\mathbf{A}_a^{N_p-3}\mathbf{B}_a & \cdots & \mathbf{C}_a\mathbf{A}_a^{N_p-N_c}\mathbf{B}_a \end{bmatrix}\ . \tag{40}$$

For a given set-point signal (or reference trajectory), the objective of the predictive control system is to bring the predicted output as close as possible to the set-point signal. One of the advantages of MPC is that MPC can uses preview capability for set-point signal. If an application allows anticipating the signal, the MPC controller with signal previewing can improve reference tracking. This objective is then translated into an input to find the "best" control parameter vector $\Delta\mathbf{U}$ such that an error function between the set-point and the predicted output is minimized. The cost function J, which reflects the control objective, is defined as follows:

$$J = (\mathbf{R}-\mathbf{Y})^T(\mathbf{R}-\mathbf{Y}) + \Delta\mathbf{U}^T\mathbf{W}_u\Delta\mathbf{U} + \mathbf{U}^T\mathbf{W}_v\mathbf{U}\ . \tag{41}$$

In Eq. (41), the first term is linked to the objective of minimizing the errors between the predicted output $\mathbf{Y}$ and the set-point signal $\mathbf{R}$. The vector $\mathbf{R}$ contains the set-point information as follows:

$$\mathbf{R} = \begin{bmatrix} \mathbf{r}(k)^T & \mathbf{r}(k+1)^T & \dots & \mathbf{r}(k+N_p-1)^T \end{bmatrix}^T , \tag{42}$$

$$\mathbf{r}(k) = \begin{bmatrix} r_1(k) & r_2(k) & \dots & r_{N_o}(k) \end{bmatrix}^T , \tag{43}$$

where $r_1(k)$ to $r_{N_o}(k)$ are the set-point signals corresponding to output vector. If an application does not allow previewing the set-point signals for some reason, $\mathbf{r}(k)$ can then be used for the rest of the prediction horizon.

The second term reflects the consideration given to the size of $\Delta\mathbf{U}$. This term reflects the consideration to minimize the control effort. $\mathbf{W}_u$ is a diagonal matrix in the form that $\mathbf{W}_u = w_u\mathbf{I}_{N_c\times N_c}$ ($w_u \geq 0$) where w_u is used as a weighting parameter for the desired closed-loop performance.

The third term also reflects the consideration to minimize the control effort but the third term is to reduce $\mathbf{u}_a$ instead of $\mathbf{\Delta u}_a$. $\mathbf{U}$ is defined as follows:

$$\begin{aligned} \mathbf{U} &= \begin{bmatrix} \mathbf{u}_a(k)^T & \dots & \mathbf{u}_a(k+N_c-1)^T \end{bmatrix}^T \\ &= \mathbf{C}_1\mathbf{u}_a(k-1) + \mathbf{C}_2\Delta\mathbf{U} , \end{aligned} \tag{44}$$

where $\mathbf{C}_1 = \mathbf{1}_{N_c\times 1} \otimes \mathbf{I}_{N_u\times N_u}$; $\mathbf{1}_{N_c\times 1}$ is the 1-vector with dimension $N_c \times 1$; $\mathbf{C_2} = \mathbf{T} \otimes \mathbf{I}_{N_u\times N_u}$; and $\mathbf{T}$ is the lower triangle with all non-zero elements are 1 and dimension $N_c \times N_c$. $\mathbf{W}_v$ in the third term is a diagonal matrix in the form that $\mathbf{W}_v = w_v\mathbf{I}_{N_c\times N_c}$ which w_u is used as a weighting parameter.

The cost function is rewritten as a function of $\Delta\mathbf{U}$ as follows:

$$J = \Delta\mathbf{U}^T\mathbf{H}\Delta\mathbf{U} + 2\Delta\mathbf{U}^T\mathbf{L} + c , \tag{45}$$

where

$$\mathbf{H} = \mathbf{\Phi}^T\mathbf{\Phi} + \mathbf{W}_u + \mathbf{C}_2^T\mathbf{W}_v\mathbf{C}_2 , \tag{46}$$

$$\mathbf{L} = \mathbf{\Phi}^T\left(\mathbf{F}\mathbf{x}_a(k) - \mathbf{R}\right) + \mathbf{C}_2^T\mathbf{W}_v\mathbf{C}_1\mathbf{u}_a(k-1) , \tag{47}$$

and c is the remainder which is not multiplied to $\Delta\mathbf{U}$. The remainder, c, is not used when solving the quadratic programming problem.

One of the other advantages of MPC is that design constraints can be considered in the optimization. To incorporate the design constraints into the control problem, it is necessary to translate the constraints into linear inequalities. The constraints are taken into consideration for each moving horizon window. Since the MPC problem is formulated and solved in the framework of the receding horizon control, the constraints on the rate of change are expressed as

$$\Delta\mathbf{u}^{min} \leq \Delta\mathbf{u}_a(k) \leq \Delta\mathbf{u}^{max}, \tag{48}$$

where the superscripts min and max denote the minimum and the maximum constraints, respectively. The constraints are defined within the control horizon so that it can be expressed in terms of function $\Delta\mathbf{U}$ as follows:

$$\Delta\mathbf{U}^{min} \leq \Delta\mathbf{U} \leq \Delta\mathbf{U}^{max}, \tag{49}$$

where $\Delta\mathbf{U}^{min}$ and $\Delta\mathbf{U}^{max}$ are column vectors with N_c elements of $\Delta\mathbf{u}^{min}$ and $\Delta\mathbf{u}^{max}$, respectively.

The constraints then need to be decomposed into two parts to reflect the lower and the upper limit. This is expressed as follows:

$$\begin{bmatrix} -\mathbf{C}_1 \\ \mathbf{C}_1 \end{bmatrix} \Delta\mathbf{U} \leq \begin{bmatrix} -\Delta\mathbf{U}^{min} \\ \Delta\mathbf{U}^{max} \end{bmatrix}, \tag{50}$$

where $\mathbf{I}$ is the identity matrix, and its size depends on N_c and $\mathbf{u}$.

This procedure also applies to the control input and output constraints. All constraints are expressed in terms of $\Delta\mathbf{U}$. The constraints for the velocity, which $\mathbf{U}$ defined in Eq. (51), is written as follows:

$$\mathbf{U}^{min} \leq \mathbf{C}_1\mathbf{u}_a(k-1) + \mathbf{C}_2\Delta\mathbf{U} \leq \mathbf{U}^{max} . \tag{51}$$

Using Eq. (36), the output constraints are expressed in terms of $\Delta\mathbf{U}$ as:

$$\mathbf{Y}^{min} \leq \mathbf{F}\mathbf{x}(k) + \mathbf{\Phi}\Delta\mathbf{U} \leq \mathbf{Y}^{max}. \tag{52}$$

Finally, the MPC in the presence of constraints is proposed as finding $\Delta\mathbf{U}$ that minimizes the quadratic cost function

$$J = \Delta\mathbf{U}^T\mathbf{H}\Delta\mathbf{U} + 2\Delta\mathbf{U}^T\mathbf{L} + c , \tag{45 revisited}$$

subject to the inequality constraints:

$$\mathbf{M}\Delta\mathbf{U} \leq \mathbf{N} , \tag{53}$$

where

$$\mathbf{M} = \begin{bmatrix} -\mathbf{C}_2 \\ \mathbf{C}_2 \\ -\mathbf{C}_1 \\ \mathbf{C}_1 \\ -\mathbf{\Phi} \\ \mathbf{\Phi} \end{bmatrix} ; \ \mathbf{N} = \begin{bmatrix} -\mathbf{U}^{min} + \mathbf{C}_1\mathbf{u}(k-1) \\ \mathbf{U}^{max} - \mathbf{C}_1\mathbf{u}(k-1) \\ -\Delta\mathbf{U}^{min} \\ \Delta\mathbf{U}^{max} \\ -\mathbf{Y}^{min} + \mathbf{F}\mathbf{x}(k) \\ \mathbf{Y}^{max} - \mathbf{F}\mathbf{x}(k) \end{bmatrix} . \tag{54}$$

To solve this quadratic programming problem, the Operator Splitting solver for Quadratic Programs (OSQP) solver is used [15]. The OSQP solver is a numerical optimization package for solving convex quadratic programs. The solver is very efficient and accurate for these kind of convex quadratic programming problems. The OSQP solver is not applicable to the more complicated nonlinear optimization problem for nonlinear MPC. However, the optimization problem with a nonlinear model can be handled with the adaptive MPC which is not much different from the standard MPC but the state-space model is updated during simulation by linearizing the nonlinear model every time step. A detailed algorithm is presented in Section III.B.2.

Although the optimal parameter vector $\Delta\mathbf{U}$ contains the controls $\Delta\mathbf{u}_a(k)$, $\Delta\mathbf{u}_a(k+1)$, $\ldots$, $\Delta\mathbf{u}_a(k+N_c-1)$, with the receding horizon control principle, the first sample of this sequence, i.e. $\Delta\mathbf{u}(k)$,

is implemented only while ignoring the rest of the sequence. When the next sample period arrives, the more recent measurement is taken to form the state vector $\mathbf{x}(k+1)$ for calculation of the new sequence of control signal. This procedure is repeated in real time to give the receding horizon control law. Fig. 6 presents the calculation flow of the standard MPC.

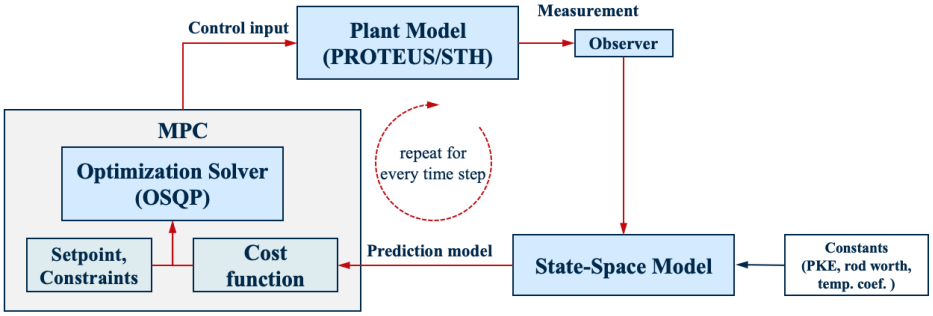


Figure 6: Calculation flow with standard MPC

III.B.2 Adaptive MPC

The theory of MPC is presented in Section III.B.2. The standard MPC described in Section III.B.2 is based on the state-space model which is linearized or defined at a particular time such as an initial steady-state condition. However, the reduced order model described in Section III.B.2 is not necessary to be linear. Ignoring the nonlinearity of the reduced order model may degrade the accuracy of reference tracking. Applying nonlinear MPC [16] may resolve the issue. However, the nonlinear MPC is computationally expensive and a nonlinear constraint is rarely necessary for many applications. Therefore, this work uses the adaptive MPC [17]. The adaptive MPC controller adjusts its prediction model at run time to compensate for nonlinear or time-varying characteristics. Figs. 6 and 7 present the calculation flow with the standard MPC and the adaptive MPC.

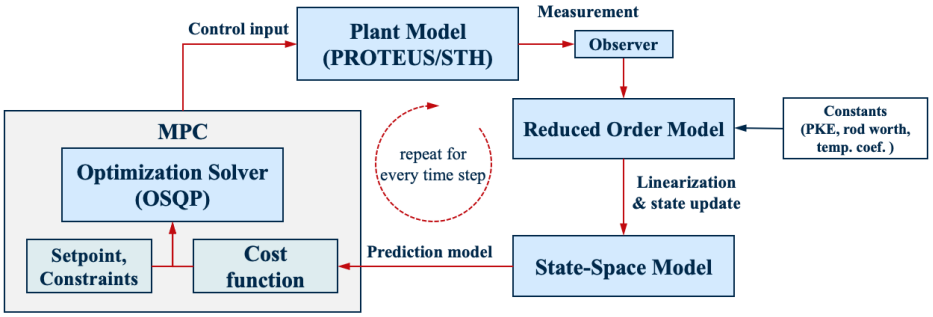


Figure 7: Calculation flow with adaptive MPC

The only different thing between the standard MPC and the adaptive MPC is that the state-space model is linearized during the simulation in the adaptive MPC. To do this, the reduced order model defined in Eq. (12) needs to be solved during the simulation as well as the plant model and the state-space model. The state-space model is updated every time step or as needed by using the exact same process in Section III.B.2. The MPC problem is updated repeatedly with the updated state-space

model. Compared to the standard MPC, the adaptive MPC requires additional calculations in solving the reduced order model and repetitive setups of the state-space model and MPC problem. However, the overhead is negligible and is readily manageable with modern computing power while the accuracy of the adaptive MPC may be comparable to that of the nonlinear MPC as long as the linearization is repeated with a short time interval. Therefore, the adaptive MPC can be a good trade-off between the standard MPC and the nonlinear MPC.

III.B.2 Model mismatch and observer

The MPC optimization is conducted based on the reduced order model. It is nearly impossible to directly use the plant model or a high-fidelity model in the MPC optimization due to the complexity and computational cost. Instead, the reduced order model is used as described in Section III.B.2. The issue is that the reduced order model also approximates the plant or high-fidelity model. It means that the solutions from the plant model and the reduced order model are different. Therefore, there should be a process to correct the difference. This is done by an observer.

Before explaining the observer, it is necessary to mention the measurement. It is necessary to consider which state variable can be obtained from the plant or high-fidelity model. In practice, it is not feasible to measure some state variables (e.g. the delay neutron precursors and core-averaged temperature for each material). The power level or relative neutron density (i.e. $\bar{n}$) may be the only parameter that can be obtained from the real plant. In this sense, it is assumed that only the power level or relative neutron density is obtained from the plant model in the simulation and the observer is used to correct the neutron density.

The measurement correction process done by the observer is quite straightforward. Once the measurement is obtained from the plant or high-fidelity model, the output vector, $\mathbf{y}_d(k)$ in Eqs. (32) and (33) is corrected as follows:

$$\mathbf{y}'_d(k) = \mathbf{y}_d(k) + \mathbf{K}\left(\tilde{\mathbf{y}}(k) - \mathbf{y}_d(k)\right) \tag{55}$$

where $\mathbf{y}'_d(k)$ is the corrected output vector; $\tilde{\mathbf{y}}(k)$ is the output vector from the plant; and $\mathbf{K}$ is the observer matrix with dimension $N_o \times N_o$. When the optimization problem is a single output case and the only power can be measured from the plant, the $\mathbf{K}$ would be $\mathbf{1}_{1\times 1}$. What the observer is doing is simply replacing the output from the state-space model with measurable output from the plant. The corrected output vector, $\mathbf{y}'_d(k)$, is then used in the subsequent MPC optimization instead of $\mathbf{y}_d(k)$.

An additional correction is necessary for the adaptive MPC since the adaptive MPC algorithm solves the reduced order model separately and the measurement correction needs to be done to the reduced order model as well. The observer for the state-space model simply corrects the output vector. However, simply updating the neutron density in Eq. (12) may lead to an unphysical neutron time derivative. The corrected neutron density is not calculated from the reduced order model so the neutron balance is not necessary to meet. An additional calculation to preserve the time derivative of neutron density is necessary to resolve the issue. In other words, the following relation needs to meet:

$$\frac{d\bar{n}(t)}{dt} = \frac{d\bar{n}'(t)}{dt} \,, \tag{56}$$

where $\bar{n}'(t)$ is the corrected neutron density which is the same as the neutron density of the plant

model. To preserve the time derivative with the corrected neutron density, an additional parameter is needed or some variables are needed to be modified. This will be handled by updating the base reactivity. Using Eq. (5), the above equation is written in following:

$$\frac{\rho(t)-\beta_t}{\Lambda}\bar{n}(t)+\frac{\sum_{i=1}^{6}\lambda_i\bar{c}_i(t)}{\Lambda}=\frac{\rho'(t)-\beta_t}{\Lambda}\bar{n}'(t)+\frac{\sum_{i=1}^{6}\lambda_i\bar{c}_i(t)}{\Lambda}\ , \tag{57}$$

where $\rho'(t)$ is the corrected reactivity corresponding to $\bar{n}'(t)$ and it is defined as a function of corrected and uncorrected neutron densities:

$$\rho'(t)=(\rho(t)-\beta_t)\,\frac{\bar{n}(t)}{\bar{n}'(t)}+\beta_t\ . \tag{58}$$

The correction on the reactivity may be assumed that the base reactivity, ρ_b, is updated to have $\rho'(t)$. The updated neutron density and the reactivity are used in the subsequent calculations.

IV Numerical Results

IV.A 3D microreactor results

In this section, the load-follow simulation results for the HTGR-type microreactor described in Section II are presented. As mentioned in Section III.B.2, one of the advantages of the MPC is an ability to consider constraints in the optimization problem. Three kinds of constraints are applied to the drum rotation rate $\mathbf{u}$, and drum rotation acceleration $\Delta\mathbf{u}/\Delta t$. It should be noted that these constraints are assumed because they have not been determined in the reactor design, yet. The numerical values of the constraints are listed in Table 2. There is no specific constraint for the control drum rotation since the drum can rotate over 0 degree or 360 degree. The constraints for $\mathbf{u}$ and $\Delta\mathbf{u}/\Delta t$ are simply chosen to have $\pm 0.1\%$/s and $\pm 0.1\%/\text{s}^2$, respectively. These constraints are corresponding to ± 0.36 deg/s and ± 0.36 deg/s^2, respectively. In reality, the constraints should be first based on the mechanical performance of the drum rotation system, and secondarily on technical specification limits for the operation that should satisfy safety and operational performance requirements.

Table 2: Applied constraints to microreactor control

Parameters	**Constraints**
Control drum rotation (deg)	$0 \le \mathbf{s} \le 360$
Control drum rotation rate (deg/s)	$-0.36 \le \mathbf{u} \le 0.36$
Control drum rotation acceleration (deg/s^2)	$-0.36 \le \Delta\mathbf{u}/\Delta t \le 0.36$

Table 3 presents parameters used in the reduced order model. The delayed neutron fraction and the delayed neutron time constants were extracted from the 6-group cross section library of PROTEUS. A typical generation time for thermal reactor was used. Therefore, the point kinetics parameters are not necessary to be accurate. The feature to generate the point kinetics parameters in PROTEUS is not ready to use yet. In Section IV.B, the sensitivity test results are presented to determine the effect of these parameters. The temperature coefficients and control drum differential worth were determined from a series of 2D core simulations.

Table 3: Parameters used in the reduced order model simulation

Parameter	Value	Unit	Parameter	Value	Unit
β	0.01181	-	Λ	2.627E-5	s
β_1	0.00041	-	λ_1	0.01334	1/s
β_2	0.00213	-	λ_2	0.03274	1/s
β_3	0.00204	-	λ_3	0.12078	1/s
β_4	0.00457	-	λ_4	0.30278	1/s
β_5	0.00187	-	λ_5	0.84949	1/s
β_6	0.00078	-	λ_6	2.85300	1/s
α_f	-4.0	pcm/K	c_f	977.0	J/kg/K
α_m	-0.2	pcm/K	c_m	1697.0	J/kg/K
α_c	0.0	pcm/K	c_c	5190.0	J/kg/K
W_1	-33.3	pcm/deg	-	-	-

Table 4: Power level setpoint

Time period (sec)	Description for power level
0 - 5	Maintain 100%
5 - 245	Decrease to 20% with ramp rate of -20%/min
245 - 1800	Maintain 20%
1800 - 2040	Increase to 20% with ramp rate to 20%/min
2040 - 3600	Maintain 100%

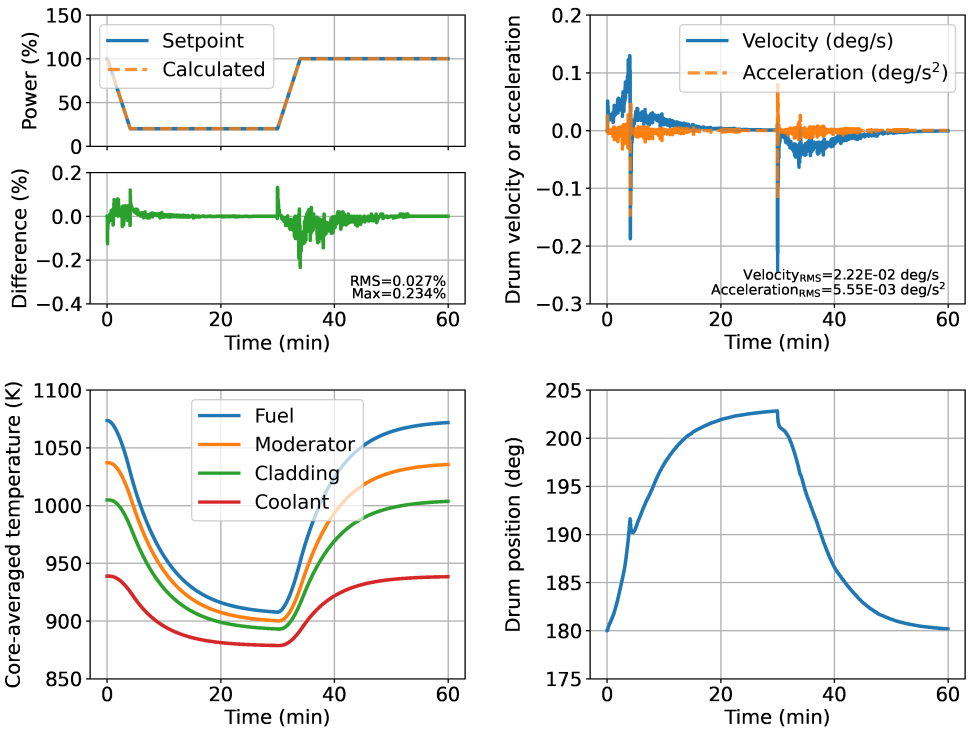


Figure 8: 3D microreactor load-follow simulation results

Table 4 present the results for the 3D microreactor load follow simulation. The prescribed power scenario starts at 100%, decreases to 20%, and then increases back to 100%. The ramp rate between 100% and 20% power is ±20%/min which is the fastest ramp rate requirement for nuclear reactors [18]. As depicted in Fig. 7, the plant is represented by PROTEUS coupled with STH. The adaptive MPC is used for the simulation. One second of the time interval is used for both PROTEUS simulation and MPC control. The initial position of the control drum is 180 deg which the center of the control drum absorber faces south as shown in the Fig. 2a. In the simulation, 40 computing cores were used.

The objective of the controller is to predict control inputs to have accurate power compared to setpoint while minimizing control costs and being regulated within the given constraints. Fig. 8 presents the load-follow simulation results. The prescribed power scenario starts at 100%, decreases to 20%, and then increases back to 100%. The calculated power with the MPC controller is very accurate. The RMS tracking error over the entire simulation is 0.027%. The maximum tracking error is 0.234%. The core-averaged temperature is also calculated for fuel, moderator, cladding, and coolant. The temperature changes very slowly than power. The power level decreases to 20% and maintains the level between 4 min and 30 min. In this period, the MPC controller needs to keep solving the optimization problem to compensate for the temperature feedback effect by adding negative reactivity with the control drum. In the same period, the control drum keeps rotating. Similar behavior is repeated when the power increases back to 100% but with opposit direction. During the load-follow simulation, the control drum rotates between 180 deg and 203 deg.

To determine the cost for input control, RMS velocity and RMS acceleration are calculated and shown in the figure. The RMS velocity and RMS acceleration are 2.22E-2 deg/s and 5.55E-3 deg/s^2, respectively. It is not straightforward to determine whether these costs are large or small. The same parameters are calculated for several other 2D core simulations and then compared to each other in the next section.

IV.B Sensitivity Test

This section presents the sensitivtiy test results on numerous parameters including the MPC method, control drum worth, kinetics parameters, TH parameters, ramp rate and the different scenario. Table 5 summarizes the tracking error and the noramlized control cost. The normalized control cost indicates the RMS velocity and RMS acceleration of the control drum over the entire simulation but these are normalized by the costs of 2D core simulation. The normalization is performed since it is not straightforward to determine whether the control costs are expensive or not based on the absolute values. Therefore, relative or normalized control costs are calculated and presented.

Table 5: Summary of sensivitiy test based on 2D simulation

Description	Tracking difference (%)		Normalized control cost	
	RMS	Max	Velocity	Acceleration
3D core simulation	0.027	0.234	1.09	1.09
2D core simulation (Base case)	0.017	0.170	1.00	1.00
Standard MPC	0.180	1.196	0.89	0.40
Drum worth −60%	0.106	0.790	4.90	37.87
Drum worth −30%	0.022	0.326	1.00	1.48
Drum worth +30%	0.031	0.172	1.00	0.88
Drum worth +60%	0.049	0.226	0.99	0.80
Position-dependent drum worth	0.019	0.166	1.00	1.03
β_i −60%	0.046	0.375	0.99	0.62
β_i +60%	0.027	0.403	1.24	2.78
λ_i −60%	0.032	0.224	1.03	1.36
λ_i +60%	0.017	0.160	0.99	0.88
λ_i +90%	0.019	0.155	0.99	0.84
Λ −30%	0.017	0.170	1.00	1.00
Λ +30%	0.017	0.170	1.00	1.00
α_f, α_m −60%	0.048	0.294	1.00	1.00
α_f, α_m −30%	0.030	0.221	1.00	1.00
α_f, α_m +30%	0.019	0.170	1.00	1.00
α_f, α_m +60%	0.035	0.186	1.00	1.00
$c_{p,f}, c_{p,m}, c_{p,c}$ −30%	0.020	0.171	1.00	1.00
$c_{p,f}, c_{p,m}, c_{p,c}$ +30%	0.022	0.192	1.00	1.00
Ramp rate 5%/min	0.012	0.097	0.61	0.32
Ramp rate 30%/min	0.021	0.384	1.27	1.63
Power 100%→140%→100%	0.015	0.140	0.40	0.24

IV.B.2 2D core simulation

For fast running, all sensitivity tests are performed with a 2D core model. The 2D core has the same design and the same size but only the boundary condition at the top and bottom surface of the core is reflective. Therefore, only one axial plan is necessary to simulate the core model. Fig. 9 shows the 2D core simulation results. This comparison is made to confirm that the 2D core model can show similar behavior as the 3D core so that the subsequent sensitivity test still represents the situation of the 3D core.

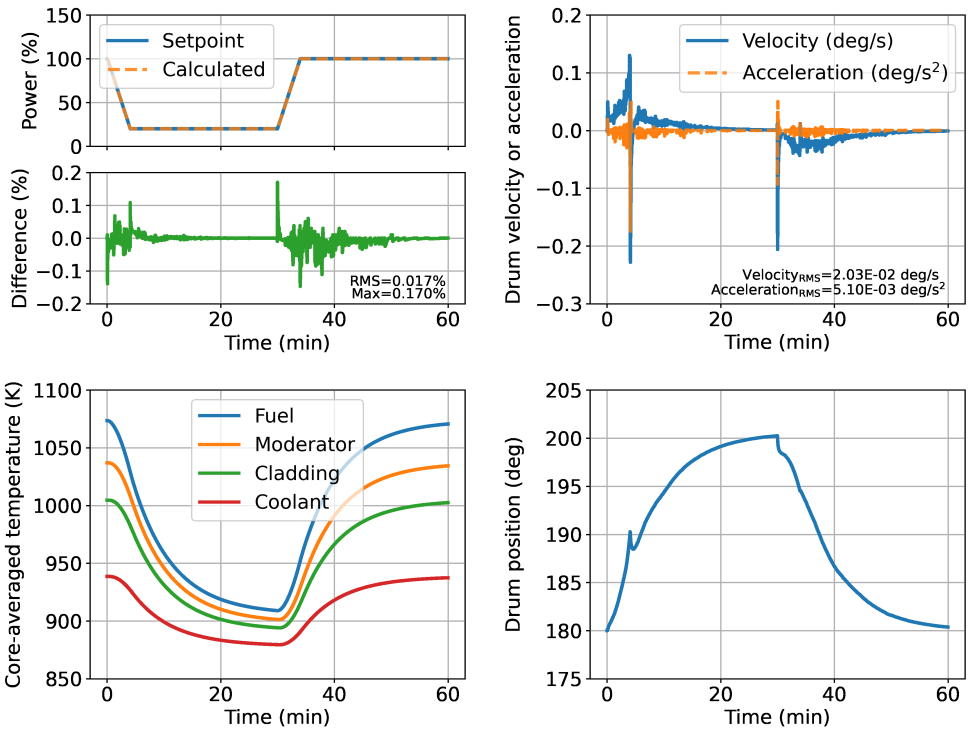


Figure 9: 2D microreactor load-follow simulation results

Table 5 also presents the comparison between the 2D and 3D core simulation. Both 2D and 3D core simulations have a similar level of accuracy. The 2D core has a smaller tracking error, i.e., the maximum error is 0.17%. The normalized control cost in Table 5 indicates the RMS velocity and RMS acceleration of the control drum over the entire simulation but these are normalized by the costs of 2D core simulation. The normalization is performed since it is not straightforward to determine whether the control costs are expensive or not based on the absolute values. Therefore, relative or normalized control costs are calculated and presented. The 3D core simulation shows about 9% higher control costs for both control drum velocity and acceleration than those of the 2D core. It may be more difficult to approximate the plant by the reduced order model since the 3D core may have a more heterogeneous flux profile. In addition, the reduced order model may approximate the 2D core model more accurately since the control drum differential worth and temperature reactivity coefficients were generated based on the 2D core simulation. These are the reasons why the 2D core load follow results have higher accuracy and lower control cost than those of the 3D core simulation. However, these differences are not significant and it can be assumed that the 2D core load follow simulation has a similar behavior as the 3D core. Therefore, the 2D core model is used for the rest of the sensitivity tests.

IV.B.2 Adaptive MPC vs. Standard MPC

In Sections IV.A and IV.B.2, the results are based on the adaptive MPC controller. As discussed in Section III.B.2, the adaptive MPC can consider the nonlinear characteristics of the reduced order model. The impact of considering the nonlinearity can be determined by comparing it to the

standard MPC case. Fig. 10 shows the results with the standard MPC controller.

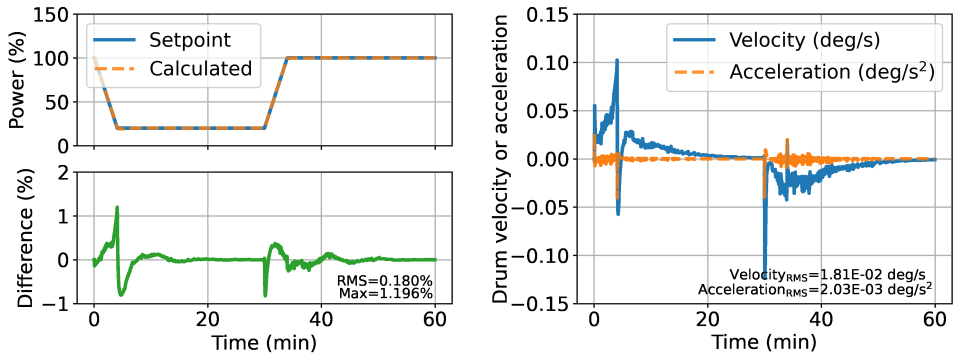


Figure 10: 2D microreactor load-follow simulation results with standard MPC

The standard MPC has less accurate results than the adaptive MPC results. The RMS tracking error is 0.18% and the maximum error is 1.196%. The difference may be acceptable but there is accuracy degradation due to the methodology in the controller. The main difference comes from the time-varying components of the matrix $\mathbf{A}_s$ in the state-space equations (i.e. Eq. (16)). Fig. 11 shows the time-varying components of $\mathbf{A}_s$ which is calculated from the adaptive MPC. Four elements from the adaptive MPC highly depend on time and vary significantly. Especially, $df_{\bar{n}}/dT_f$, $df_{\bar{n}}/dT_m$, and $df_{\bar{n}}/dD_1$ have upto 80% differences compared to those at the initial condition. On the other hand, the state-space model of the standard MPC is generated at the initial steady state and does not vary during the simulation. It means the standard MPC overestimates the magnitude of $df_{\bar{n}}/dT_m$, and $df_{\bar{n}}/dD_1$ between 4 min and 30 min. According to the power difference in Fig. 10, once a tracking error is calculated for some reason, the tracking error does not shrink rapidly. The controller needs to calculate more aggressive control input to eliminate the tracking error. However, the controller cannot do it due to the overestimated $df_{\bar{n}}/dD_1$. The normalized control costs of the standard MPC are smaller than those of the adaptive MPC due to the same reasons – overestimated $df_{\bar{n}}/dD_1$. If the state-space model is linearized at low power, then the controller may overshoot the control input due to underestimated $df_{\bar{n}}/dD_1$, and the power may oscillate – this result is not presented but it can be guessed.

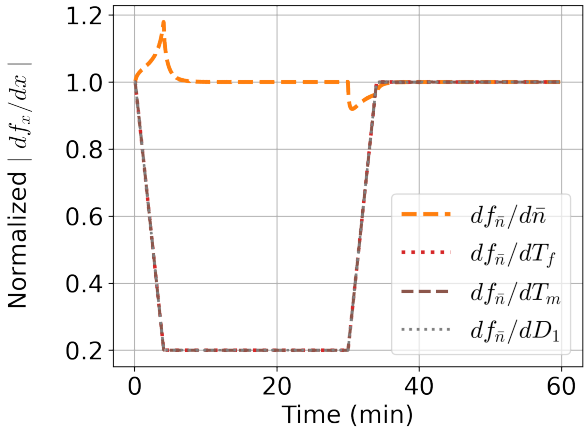


Figure 11: Time-varying components of matrix $\mathbf{A}_s$ from adaptive MPC

IV.B.2 Control drum reactivity worth

A wide range of the control drum reactivity worth is tested for the 2D microreactor load-follow simulation in this section. The reactivity worth varies from -60% to 60% compared to the reference value listed in Table 3. The tracking error increases as the drum worth moves away from the base case. As mentioned in Section IV.B.2, the larger drum worth (i.e. $df_{\bar{n}}/dD_1$) makes the controller undershoot the control input causing smaller control cost but larger tracking error. On the other hand, as the drum worth becomes smaller, the larger control cost is spent. It means the controller with the small drum worth may overshoot the control input so that the solution may oscillate in a severe case. This situation is clearly shown in Fig. 12. The control inputs with the 60% smaller drum worth hit the constraints a lot of times while the power and control input oscillate. If the constraints are not used, the simulation may diverge due to the aggressive control inputs.

An additional comparison is made for the position-dependent control drum worth. As presented in Fig. 4a, the control drum worth may vary depending on the position. The tracking error and the control costs are not much different from those of the base case which uses the constant control drum worth. This is because the control drum differential worth is fairly flat in the operation range which is 180 - 205 deg. As shown in Fig. 4a, the slope of the reactivity between the ranges is monotonic. If the control drum is initially located in a different rotation (such as 220 deg), the position-dependent worth may be essential to have accurate and stable results.

In this section, the sensitivity test results with different control drum reactivity worths are presented. The simulation results show that it is important to have an accurate estimate of the control drum worth to have accurate and stable simulation results. However, the controller with $\pm 30\%$ error in the control drum worth still predicts pretty accurate and stable results. The $\pm 30\%$ of error in the control drum worth is a quite large margin therefore the MPC controller is reasonably robust.

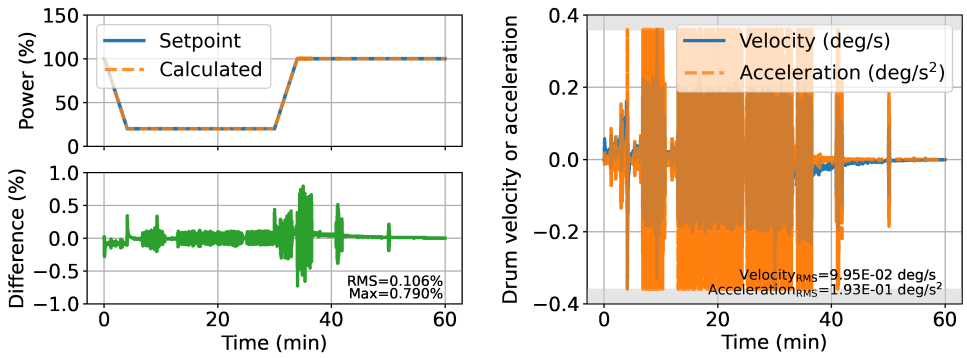


Figure 12: 2D core simulation results with 60% lower control drum reactivity worth

IV.B.2 Point kinetics parameters

A various range of point kinetics parameters is tested in this section. As mentioned in Section IV.A, the point kinetics parameters may have some error since the parameters were not calculated from homogenization by weighting solutions from PROTEUS transport calculation. Instead, medium values or typical values were chosen for the reduced order model. In practice, it may be difficult to calculate the point kinetics parameters for every configuration and condition of the real reactor. The point kinetics parameters used in the controller may have some amount of error. Therefore, it is important to check how much the MPC controller is sensitive to the selection of the point kinetics parameters.

The control cost and the tracking error increase as β_i increase. This means that the reduced order model underestimates the effect of the prompt neutron so that the MPC controller overshoots the control input to compensate for the underestimated prompt neutron effect. On the contrary, the reduced order model overestimates the effect of the prompt neutron as the β_i decreases. The MPC controller undershoots the control inputs in this case. The small β_i reduces the control cost but it increases the tracking error.

The decreased λ_i means the deleted neutron has a longer half-life so that the effect of the prompt neutron is underestimated. The MPC controller overshoots the control input due to the underestimated prompt neutron effect. When using 30% increased λ_i, the simulation results look like to have better accuracy in terms of RMS tracking error, and use less amount of control cost. This result is likely interpreted that increasing λ_i makes the overall reduced order model be more accurate than the equations with the base parameters since not all parameters used in the reduced order model are accurate. However, the RMS tracking error increases again if the λ_i is increased by 60% and 90%.

Changing the prompt neutron generation time does not make any noticeable difference. There are very small difference between these simulations so that the results in the table have the same value depending on cases. The neutron time derivative in Eq. (5) shows that Λ exists at the denominator of the entire right hand side equations. Therefore, all $\partial f_{\bar{n}}/\partial x$ (i.e. $\partial f_{\bar{n}}/\partial \bar{n}$ ... $\partial f_{\bar{n}}/\partial D_1$) change with the same fraction (30% or -30%) when Λ changes in the sensitivity test. If the temperature feedback or the drum reactivity worth in Eq. (11) have more complicated form rather than the simple linear equation, the sensitivity results per Λ may be different.

IV.B.2 Temperature reactivity coefficients and heat capacity

The control cost and tracking accuracy do not have significant impact from choosing the temperature related parameters such as temperature reactivity coefficients and heat capacity is relatively small since the reactivity feedback effect from the temperature change is very slow compared to neutronics. As shown in Fig. 9, the temperature changes relativelty slowely than the power level. Changing the heat capacity and the mass makes the same results since both heat capacity and the mass are multiplied to the left hand side of Eqs. (7) to (9). Therefore, additional sensitivity test on the mass is not performed. It can be assumed that changing the mass has the same effect as changing heat capacity.

IV.B.2 Ramp rate and power level

A wide range of ramp rates and different scenarios are tested in this section. It is expected that a slow transient scenario is easier to calculate the control input for the controller than a fast transient scenario since the slow transient scenario has smooth power change. Table 5 presents the sensitivity results with 5%/min ramp rate to 30%/min ramp rate including the base case. Figs. 13 and 14 show the detailed simulation results. As expected, the simulation results have a higher tracking error when using the fast ramp rate. For the 5%/min of ramp rate case, the maximum tracking error is only 0.097% and the normalized control cost for the acceleration is only 32% of the base case. On the other hand, the case with 30%/min ramp rate has 0.384% of maximum tracking error and 63% higher control costs than that of the base case. From the sensitivity test on the ramp rate, it is concluded that the MPC controller can calculate accurate control inputs to follow the setpoint for a wide range of ramp rates.

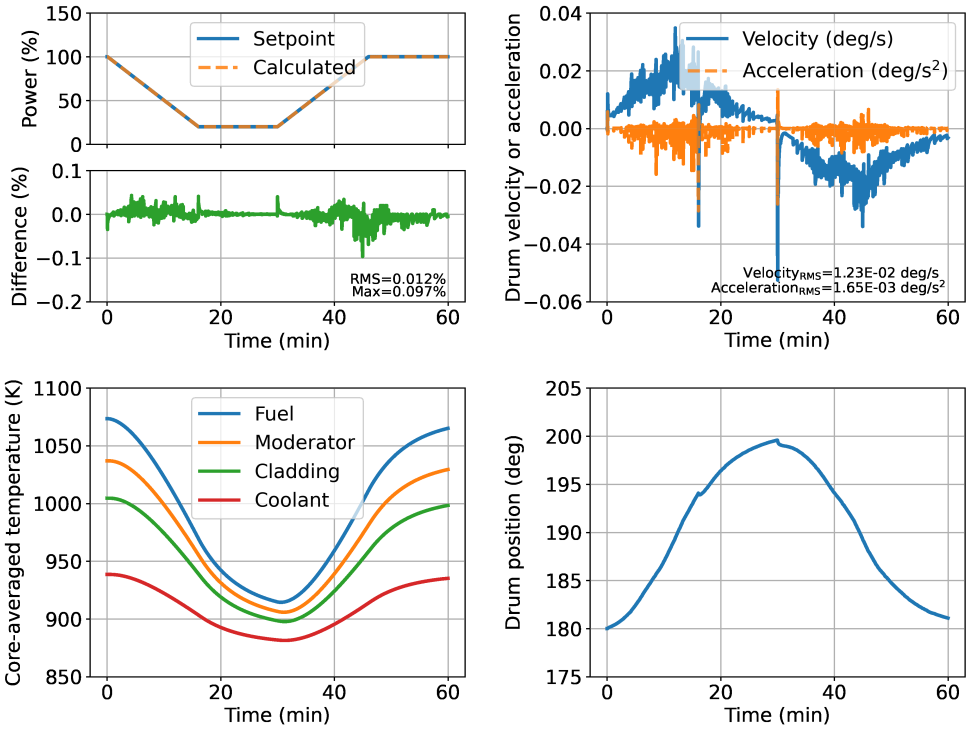


Figure 13: 2D core simulation results with 5% ramp rate

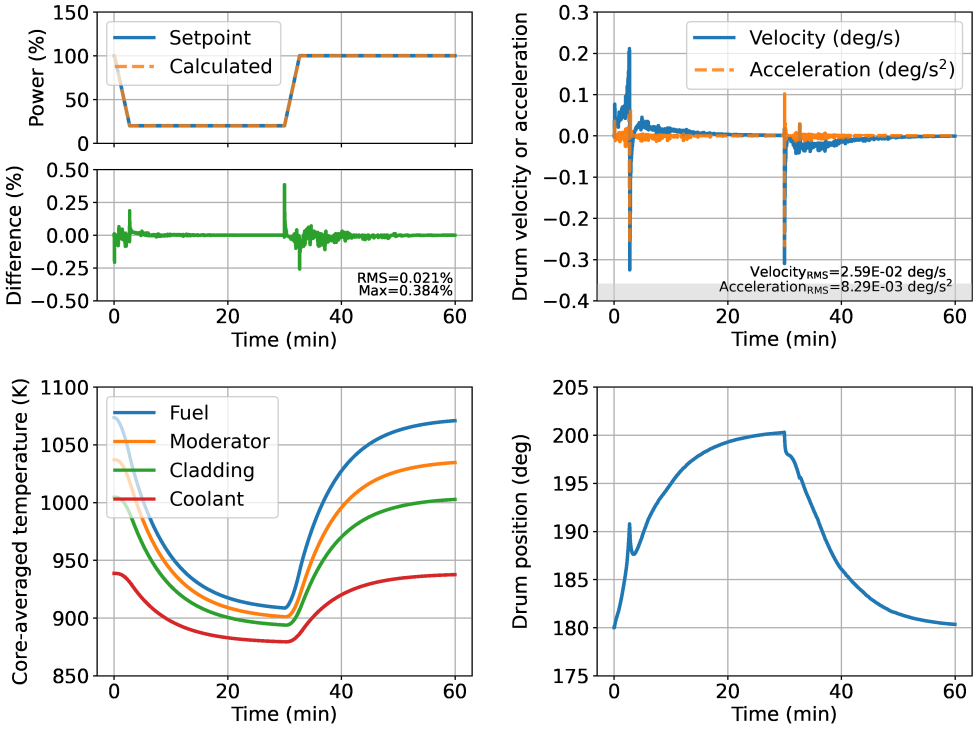


Figure 14: 2D core simulation results with 30% ramp rate

A different scenario with uprating power is also simulated and presented in Fig. 15. The scenario has 100% power initially, increases the power to 140%, and then comes back to the original power. In this simulation, it is confirmed that the code system does work properly even for a higher power. The control drum needs to rotate in the other direction to reduce the negative reactivity from the control drum in order to raise the power level. From the simulation results, it is also confirmed that the control drum decusping function works well in the other direction as well. There is no noticeable difference in the tracking error or the control cost compared to that of the base case. The simulation has accurate results and does not have a significant overhead on the control cost.

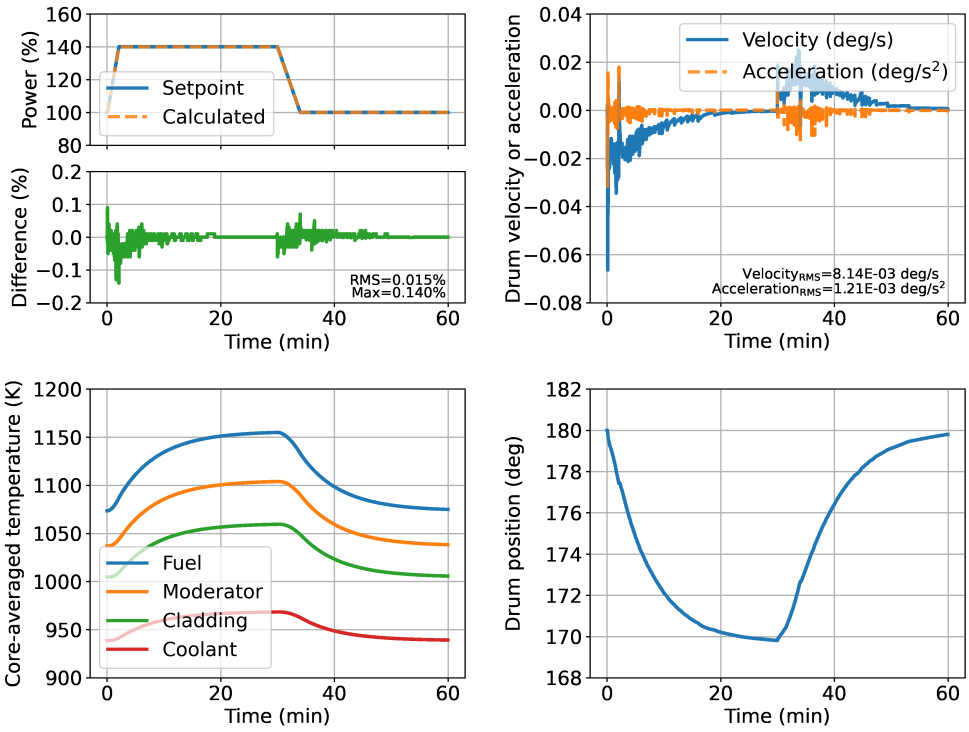


Figure 15: 2D core simulation results with maximum 140% power level

IV.C Discussion on Numerical Results

In Section Section IV, we demonstrate the load-follow simulation for a 3D microreactor similar to the HTGR. The MPC controller predicts highly accurate control inputs, thus enabling the prescribed power to be followed precisely. The RMS error is 0.027%, and the maximum error reported is 0.234%.

A potential concern with the MPC controller involves its need for a model that represents the plant. Obtaining a highly accurate reduced-order model may be impractical, given the inherent and inevitable presence of model error. Therefore, a series of sensitivity tests were performed on the MPC methods and model parameters, as described in Section Section IV.B.

The sensitivity tests confirm that the MPC controller allows for a considerable margin when selecting parameters for inclusion in the reduced order model. Even with an error range as much as

±60% in the point kinetics parameters, temperature reactivity coefficients, and lumped TH models, the maximum tracking error remains below 0.5%. The control drum reactivity worth exhibits more sensitivity than the other variables. For instance, a simulation with a 60% decrease in control drum worth results in significant oscillation in the control inputs, which strike constraints. However, simulations with control drum worth values ranging from 30% lower to 60% higher showed accurate results, as seen in a maximum tracking error less than 0.326%.

The sensitivity test demonstrates that overshooting in the control input can negatively impact the precision and stability of simulation. The MPC controller exceeds the control input when parameters such as underestimated control drum worth, overestimated decay constant β_i, and underestimated decay constant λ_i are used. Temperature-based parameters, like the temperature reactivity coefficient and heat capacity, could also produce similar effects. However, these parameters are less sensitive than others due to the slow rate of temperature change relative to the power or neutron density. While undershooting the control input can diminish tracking accuracy, it does not cause significant solution oscillation.

V Conclusion

This paper presents the practicality of an MPC controller for autonomous load follow operation of a HTGR type microreactor. The high fidelity neutronics code, PROTEUS, is used to emulate the actual plant model, making the simulation more reliable. The simplified TH and control drum decusping method have both been implemented in PROTEUS. A reduced order model, based on the point kinetics equation and lumped TH models, has been established and incorporated in the MPC controller. The adaptive MPC controller is deployed for computing the control input to adhere to the given power scenario. This adaptive MPC sequentially linearizes the nonlinear reduced order model and utilizes it for optimization, thereby enhancing accuracy.

In the assessment of numerical results, the PROTEUS/MPC code system is evaluated for the load follow operation of a 3D microreactor with a ramp rate at 20% of the maximum. The simulation reveals that the error in the tracking power is under 0.234%, while the control inputs remain within the established constraints. For further verification, extensive sensitivity tests have been performed by adjusting parameters used in the MPC controller and reduced order model. The conclusion drawn confirms that the MPC controller offers a sizable margin for adjusting parameters, thus validating the feasibility of autonomous control with the MPC controller. For additional verificaiton, a comprehensive series of sensitivity tests was conducted by modifying parameters within the MPC controller and the reduced order model. The results of these tests confirm that the MPC controller allows a significant range of flexibility for parameter adjustment. This flexibility provides robust assurance of the autonomous control feasibility via the MPC controller.

Acknowledgements

This work was supported by funding received from the DOE Office of Nuclear Energy's Nuclear Energy University Program under contract number DE-NE0008887.

References

[1] Man Gyun Na, Sun Ho Shin, and Whee Cheol Kim. "A model predictive controller for nuclear reactor power". In: *Journal of Korean Nuclear Society* 35.5 (2003), pp. 399–411.

[2] Guoxu Wang et al. "State-Space Model Predictive Control Method for Core Power Control in Pressurized Water Reactor Nuclear Power Stations". In: *Nuclear Engineering and Technology* 49.1 (2017), pp. 134–140. ISSN: 1738-5733.

[3] H Eliasi, MB Menhaj, and H Davilu. "Robust nonlinear model predictive control for nuclear power plants in load following operations with bounded xenon oscillations". In: *Nuclear Engineering and Design* 241.2 (2011), pp. 533–543.

[4] Amine Naimi et al. "Nonlinear model predictive control using feedback linearization for a pressurized water nuclear power plant". In: *IEEE Access* 10 (2022), pp. 16544–16555.

[5] C. H. Lee et al. "Simulation of TREAT Cores Using High-Fidelity Neutronics Code PROTEUS". In: *Proc. M&C 2017*. 2017, pp. 16–20.

[6] Sooyoung Choi et al. "Preliminary Results of Load Follow Simulation for Holos-Quad Microreactor using PROTEUS and Model Predictive Control". In: *M&C 2023*. ANS. 2023.

[7] *HolosGen LLC*. URL: `http://www.holosgen.com`.

[8] N. Stauff, C. H. Lee, and C Filippone. *Core Design of the Holos-Quad Microreactor*. Tech. rep. ANL/NSE-22/4. Argonne National Lab.(ANL), Argonne, IL (United States), 2022.

[9] Yeon Sang Jung and Won Sik Yang. "A Consistent CMFD Formulation for the Acceleration of Neutron Transport Calculations Based on the Finite Element Method". In: *Nuclear Science and Engineering* 185.2 (2017), pp. 307–324.

[10] Gang Yang et al. "Development of coupled PROTEUS-NODAL and SAM code system for multiphysics analysis of molten salt reactors". In: *Annals of Nuclear Energy* 168 (2022), p. 108889.

[11] Brendan Kochunas et al. "VERA core simulator methodology for PWR cycle depletion". In: *Proc. M&C 2015*. 2015.

[12] Sooyoung Choi et al. "Comparative Study for Load-follow Operations of the Holos Microreactor". In: *Proc. M&C 2021*. 2021.

[13] Carlos E. García, David M. Prett, and Manfred Morari. "Model predictive control: Theory and practice—A survey". In: *Automatica* 25.3 (1989), pp. 335–348. ISSN: 0005-1098.

[14] Liuping Wang. *Model predictive control system design and implementation using MATLAB®*. London, UK: Springer Science & Business Media, 2009. ISBN: 978-1-84882-330-3.

[15] Bartolomeo Stellato et al. "OSQP: An operator splitting solver for quadratic programs". In: *Mathematical Programming Computation* 12.4 (2020), pp. 637–672.

[16] Rolf Findeisen and Frank Allgöwer. "An introduction to nonlinear model predictive control". In: *21st Benelux meeting on systems and control*. Vol. 11. The Netherlands: Technische Universiteit Eindhoven Veldhoven. 2002, pp. 119–141.

[17] Veronica Adetola, Darryl DeHaan, and Martin Guay. "Adaptive model predictive control for constrained nonlinear systems". In: *Systems & Control Letters* 58.5 (2009), pp. 320–326.

[18] Sooyoung Choi, Shai Kinast, and Brendan Kochunas. *Point Kinetics Model Development with Predictive Control for Multi-Module HTGR Special Purpose Reactors*. Tech. rep. NURAM-2020-006-00. University of Michigan, Ann Arbor, MI (United States), 2020.