


NE/8975-2021-011-00

Development of Learning-based Model Predictive Control Framework for SMRs

Sooyoung Choi[1], Marisol Garrouste[1], Una Baker[2], Benjamin Lindley[2], and Brendan Kochunas[1]

[1] *University of Michigan*
[2] *University of Wisconsin, Madison*

7/30/2021

NEUP Nuclear Energy University Program
U.S. Department of Energy

This page is intentionally blank.

REVISION LOG

Revision	Date	Affected Pages	Revision Description
0	7/30/2021	All	Initial Release

Document pages that are:

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

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

This work was supported by funding received from the DOE Office of Nuclear Energy's Nuclear Energy University Program under contract number DE-NE0008975.

EXECUTIVE SUMMARY

In this report, we document the development of a reactor dynamics and Learning-Based Model Predictive Control (LBMPC) algorithm for the autonomous reactivity control of a Small Modular Reactor (SMR). The reactor dynamics model includes the Point Kinetics Equations (PKE), Thermal-hydraulics (TH) models, and Xenon dynamics. The position-dependent control rod worth is used to demonstrate a realistic situation. The nonlinearity of the reactor dynamics models causes a model mismatch with the linear state-space model used in the MPC controller, degrading the accuracy of the controller. The LBMPC controller is developed to minimize the error caused by the model mismatch. The Gaussian Process Regression (GPR) algorithm is used to train a way to update the state-space model as reactor condition evolves. In the training, the nonlinear model is successively linearized and the piecewise state-space model information is provided to the GPR. The trained GPR model provides improved state-space models to the MPC controller every time step resulting in better accuracy for reference power tracking.

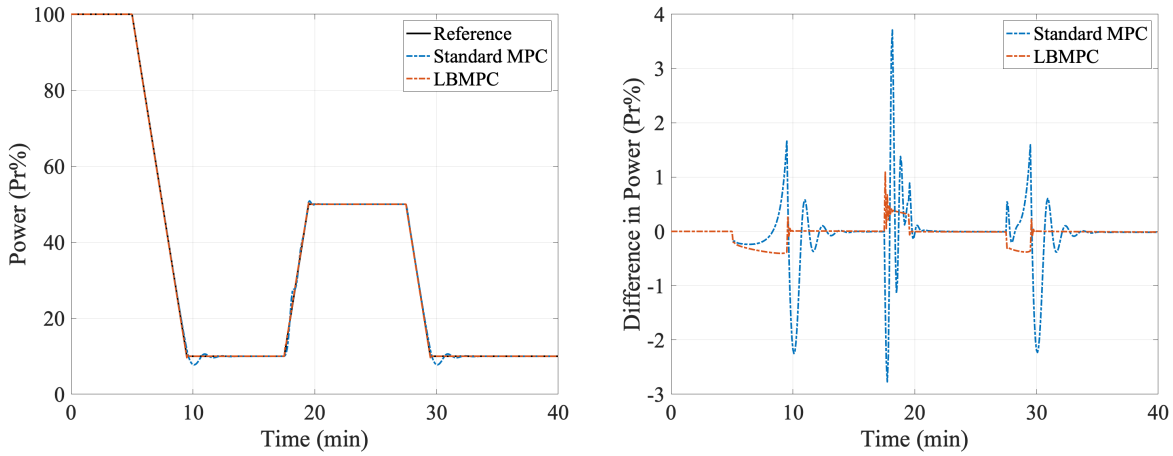

Figure EC.1. Simulation results for the short transient scenario using LBMPC

This page is intentionally blank.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

PKE Point Kinetics Equation

TH Thermal Hydraulics

MPC Model Predictive Control

LBMPC Learning-Based MPC

GPR Gaussian Process Regression

GP Gaussian Process

ML Machine Learning

SVM Support Vector Machine

SMR Small Modular Reactors

FPO Flexible Power Operation

QP Quadratic Programming

1. INTRODUCTION

In the commercial nuclear power sector, Small Modular Reactors (SMR) are becoming more attractive due to their lower capital costs, and presumed advantages for load follow operation. The nuclear power industry needs to adapt to the inherently intermittent renewable generation in the electricity grid. As renewables make up a larger share of the generating capacity, this challenges base load power generation in deregulated markets. The demands for Flexible Power Operation (FPO) capabilities are likely to continue to increase as nations push for reduced carbon emissions by increasing renewable energy generation complemented by nuclear generation. Therefore, FPO capabilities are an increasingly valued and important requirement for future nuclear power plant designs.

The most likely deployment scenarios for SMR assume multi-unit sites; this provides flexibility to replace traditional gigawatt generating stations with grid-appropriate generation and flexible outage planning. Operation of a multi-unit site represents a step function increase in complexity. This project seeks to address these challenges through enhanced automation control strategies for multi-unit SMRs. Specifically, the objective of this project is to develop a hierarchy of automation control strategies for FPO using the NuScale plant as the reference SMR design.

To achieve the overall project objective, several problems must be addressed. The focus of this work is to advance the techniques of model based control to integrate machine learning into the control algorithm that results in an adaptive controller. There are a variety of state-of-the-art control algorithms in the reactor control field. The Model Predictive Control (MPC) [1] is one of the most popular algorithms used widely in industries owing to its high performance and high accuracy. The MPC calculates an accurate control input for the plant as long as the MPC controller has an appropriate mathematical model. However, in practice, an actual reactor model is highly nonlinear and it is rarely possible to represent the reactor as a real-time numerical model to be used in a controller. There is an inevitable mismatch between the actual plant model and the model used in the controller. The mismatch between the controller state-space model of the reactor and the real physics may degrade the accuracy and performance of the controller in reference power tracking. To address this issue, we have investigated several approaches to introduce a learning-based component into the MPC framework. A Machine Learning (ML) technique called Gaussian Process Regression (GPR) is used to improve the mathematical model in the MPC controller resulting in better accuracy for the reference power tracking.

This report is organized as follows: first, we present the nonlinear reactor dynamics models for SMR and derivation of the state-space model. Then, we present point kinetics parameters, temperature coefficients, and Xenon worth from 3 cycles of SMR simulation. We next present the algorithm and theory of MPC to control the reactor which has the nonlinear model. Here, we show a need to improve the MPC controller since accuracy is degraded due to model mismatch. Therefore, we propose a learning-based control algorithm. We show numerical results using the learning-based controller to show accuracy improvement and verify the implementation. Finally, we summarize our conclusion and note the future work of this project.

2. REACTOR DYNAMICS MODELS

In this section, we use the Point Kinetics Equation (PKE) and Thermal Hydraulics (TH) models to model the time-dependent behavior of the SMR. We use a standard form of PKE with 6 delayed groups. The PKE at time t are:

$$\frac{dn(t)}{dt} = \frac{\rho(t) - \beta}{\Lambda} n(t) + \sum_{i=1}^{6} \lambda_i C_i(t) \ , \tag{1}$$

$$\frac{dC_i(t)}{dt} = \frac{\beta_i}{\Lambda} n(t) - \lambda_i C_i(t) \ , \quad i = 1...6 \ , \tag{2}$$

where $n(t)$ is the neutron density; $\rho(t)$ is the reactivity; β is the total effective delayed neutron fraction; Λ is the generation time; λ_i is the i-th group effective delayed neutron precursor decay constant; C_i is the i-th group effective delayed neutron precursor density; and β_i is the i-th group effective delayed neutron fraction.

The TH equations are based on fuel and coolant two temperature models as follow:

$$m_f c_f \frac{dT_f(t)}{dt} = q\kappa n(t) - K_{fc} \left(T_f(t) - T_c(t)\right) \ , \tag{3}$$

$$m_c c_c \frac{dT_c(t)}{dt} = (1-q)\kappa n(t) + K_{fc} \left(T_f(t) - T_c(t)\right) - 2\dot{m}_c c_c (T_c(t) - T_i) \ , \tag{4}$$

where m_f and m_c are the mass of fuel and coolant, respectively; c_f and c_c is the heat capacity of fuel and coolant, respectively; $T_f(t)$ and $T_c(t)$ are the fuel and coolant temperature, respectively; q is the fraction of the heat deposited in the fuel region; κ is the energy release per neutron density; K_{fc} is the heat transfer coefficient from the fuel to coolant; $\dot{m}_c$ is the mass flow rate of coolant; and T_i is the inlet coolant temperature, and T_i is assumed to be constant in this work. In the Eq. (4), it is assumed that $T_c(t) = (T_{out}(t) - T_{in})/2$ where $T_{out}(t)$ is the outlet temperature. The rated power of reactor is $P_r(t)$ and it has following relation:

$$P_r(t) = \kappa n(t) \ . \tag{5}$$

To model the realistic load-following, we introduce model for ^{135}I and ^{135}Xe. The concentration of ^{135}I and ^{135}Xe are

$$\frac{dI(t)}{dt} = \gamma_I \Sigma_f \upsilon n(t) - \lambda_I I(t) \ , \tag{6}$$

$$\frac{dX(t)}{dt} = \gamma_X \Sigma_f \upsilon n(t) + \lambda_I I(t) - \lambda_X X(t) - \sigma_X \upsilon n(t) X(t) \ , \tag{7}$$

where $I(t)$ and $X(t)$ are the densities of ^{135}I and ^{135}Xe, respectively; γ_I and γ_X are the fission yields of ^{135}I and ^{135}Xe, respectively; λ_I and λ_X are the decay constants of ^{135}I and ^{135}Xe, respectively;

σ_X is the microscopic absorption cross section of ^{135}Xe; Σ_f is the macroscopic fission cross section of fuel; and v is the average velocity of thermal neutrons.

The reactivity model includes temperature feedback, ^{135}Xe worth, and control rod worth. With assumption that the reactor is steady-state at the beginning, the equation is:

$$\begin{aligned}\rho(t) &= \rho_f(t) + \rho_c(t) + \rho_X(t) + \rho_r(t) \\ &= \alpha_f \left(T_f(t) - T_f(0)\right) + \alpha_c \left(T_c(t) - T_f(0)\right) - \frac{\sigma_X}{\nu\Sigma_f}\left(X(t) - X(0)\right) + \rho_r(t) \ ,\end{aligned} \tag{8}$$

where α_f and α_c are the temperature coefficients of fuel and coolant, respectively; ν is the number of neutrons produced per fission reaction; and $\rho_r(t)$ is the control rod worth and it is approximated as follows [2]:

$$\rho_r(t) = \frac{W}{(1.0 + exp\left(\frac{r(t)-76.053}{-36.967}\right)} - \frac{W}{(1.0 + exp\left(\frac{r(0)-76.053}{-36.967}\right)} \ , \tag{9}$$

where W is the integral bank worth. The position dependent bank worth is presented in Fig. 1.

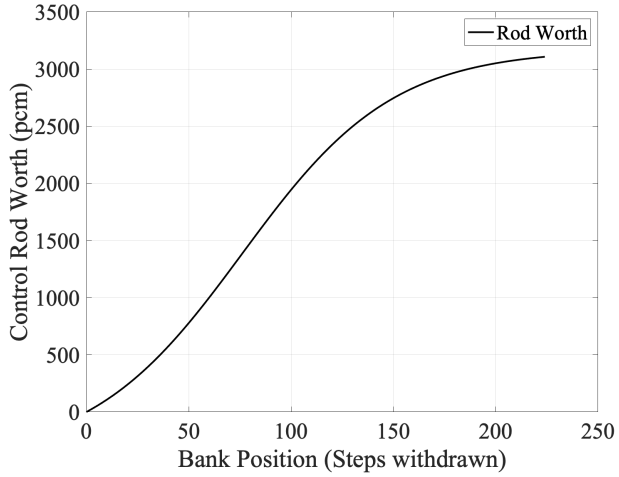

Figure 1. Control rod worth

The equation for rod position, $r(t)$, is

$$\frac{dr(t)}{dt} = u(t) \ , \tag{10}$$

where $u(t)$ is the control rod movement speed.

Eqs. (1) to (4) and (6) to (10) are rewritten in the following form:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \ , \tag{11}$$

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:

$$\begin{aligned}\mathbf{x}(t) &= \begin{bmatrix} x_1 & \dots & x_m \end{bmatrix}^T \\ &= \begin{bmatrix} n(t) & C_1(t) & \dots & C_6(t) & T_f(t) & T_c(t) & I(t) & X(t) & r(t) \end{bmatrix}^T ,\end{aligned} \tag{12}$$

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

$$\mathbf{u}(t) = \begin{bmatrix} u_1 \end{bmatrix}^T = \begin{bmatrix} u(t) \end{bmatrix}^T . \tag{13}$$

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

$$\begin{aligned}\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) &= \begin{bmatrix} f_1 & \dots & f_m \end{bmatrix}^T \\ &= \begin{bmatrix} f_n & f_{c_1} & \cdots & f_{c_6} & f_{T_f} & f_{T_c} & f_I & f_X & f_r \end{bmatrix}^T \\ &= \begin{bmatrix} \frac{dn(t)}{dt} & \frac{dC_1(t)}{dt} & \cdots & \frac{dC_6(t)}{dt} & \frac{dT_f(t)}{dt} & \frac{dT_c(t)}{dt} & \frac{dI(t)}{dt} & \frac{dX(t)}{dt} & \frac{dr(t)}{dt} \end{bmatrix}^T\end{aligned} . \tag{14}$$

Note that $\mathbf{f}$ is the system of nonlinear equations. Especially, Eqs. (1) and (7) have nonlinear terms. 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:

$$\begin{aligned}\dot{\mathbf{x}}_s(t) &= \mathbf{A}_s \mathbf{x}_s(t) + \mathbf{B}_s \mathbf{u}_s(t) \\ \mathbf{y}_s(t) &= \mathbf{C}_s \mathbf{x}_s(t) ,\end{aligned} \tag{15}$$

where $\mathbf{A}_s$ is the system matrix; $\mathbf{B}_s$ is the input matrix; and $\mathbf{C}_s$ is the output matrix; $\mathbf{x}_s(t)$, $\mathbf{u}_s(t)$, and $\mathbf{y}_s(t)$ are the state vector, input vector, and output vector of state-space model, respectively.

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:

$$\begin{aligned}\mathbf{x}_s(t) &= \mathbf{x}(t) - \mathbf{x}(t_n) \\ \mathbf{u}_s(t) &= \mathbf{u}(t) - \mathbf{u}(t_n) \\ \mathbf{y}_s(t) &= \mathbf{y}(t) - \mathbf{y}(t_n) ,\end{aligned} \tag{16}$$

where t_n is the time at the nominal condition.

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

$$g(z) \approx g(z_n) + \frac{dg(z_n)}{dz}(z - z_n) . \tag{17}$$

Similarly, Eq. (11) is approximated at nominal time t_n as follows:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t_n), \mathbf{u}(t_n)) + \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{t=t_n} (\mathbf{x}(t) - \mathbf{x}(t_n)) + \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{t=t_n} (\mathbf{u}(t) - \mathbf{u}(t_n)) \ , \tag{18}$$

where

$$\mathbf{A}_s = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{t=t_n} = \left[\frac{\partial \mathbf{f}}{\partial x_1} \quad \cdots \quad \frac{\partial \mathbf{f}}{\partial x_m}\right]\Bigg|_{t=t_n} = \left[\begin{array}{ccc} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_m} \end{array}\right]\Bigg|_{t=t_n} , \tag{19}$$

$$\mathbf{B}_s = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{t=t_n} = \left[\frac{\partial \mathbf{f}}{\partial u_1}\right]\Bigg|_{t=t_n} = \left[\begin{array}{c} \frac{\partial f_1}{\partial u_1} \\ \vdots \\ \frac{\partial f_m}{\partial u_1} \end{array}\right]\Bigg|_{t=t_n} . \tag{20}$$

Therefore, Eq. (18) becomes

$$\dot{\mathbf{x}}(t) - \dot{\mathbf{x}}(t_n) = \mathbf{A}_s(\mathbf{x}(t) - \mathbf{x}(t_n)) + \mathbf{B}_s(\mathbf{u}(t) - \mathbf{u}(t_n)) \ . \tag{21}$$

Eq. (21) is the same form as Eq. (15). 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.

3. REACTOR DYNAMICS PARAMETERS

The MPACT [3, 4] model for the NuScale reactor was developed, and simulated for cycles 1 to 3. It was assumed that the fuel temperature and coolant temperature are constant over the cycle as 900.00 K and 531.48 K, respectively. At BOC and EOC of each cycle, kinetics parameters, temperature coefficients, and equilibrium Xenon worth were calculated and presented in Tables 1 to 4.

Table 1. Kinetics parameter β and Λ

State	β	Λ
Cycle 1 BOC	7.3847e-03	2.6266e-05
Cycle 1 EOC	5.5089e-03	2.9059e-05
Cycle 2 BOC	6.3079e-03	2.0397e-05
Cycle 2 EOC	5.4761e-03	2.5192e-05
Cycle 3 BOC	6.3336e-03	1.8739e-05
Cycle 3 EOC	5.5502e-03	2.3221e-05

Table 2. Kinetics parameter β_i

State	β_1	β_2	β_3	β_4	β_5	β_6
Cycle 1 BOC	2.3517e-04	1.2595e-03	1.2282e-03	2.8552e-03	1.2760e-03	5.3072e-04
Cycle 1 EOC	1.6538e-04	9.8182e-04	9.0075e-04	2.0653e-03	1.0016e-03	3.9410e-04
Cycle 2 BOC	1.9606e-04	1.0983e-03	1.0419e-03	2.4038e-03	1.1158e-03	4.5200e-04
Cycle 2 EOC	1.6254e-04	9.8193e-04	8.9214e-04	2.0483e-03	9.9961e-04	3.9163e-04
Cycle 3 BOC	1.9661e-04	1.1045e-03	1.0457e-03	2.4132e-03	1.1200e-03	4.5354e-04
Cycle 3 EOC	1.6542e-04	9.9392e-04	9.0530e-04	2.0792e-03	1.0097e-03	3.9664e-04

Table 3. Kinetics parameter λ_i

State	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6
Cycle 1 BOC	1.3357e-02	3.2640e-02	1.2096e-01	3.0425e-01	8.5347e-01	2.8668e+00
Cycle 1 EOC	1.3346e-02	3.1767e-02	1.1789e-01	3.0053e-01	8.5775e-01	2.8272e+00
Cycle 2 BOC	1.3349e-02	3.2188e-02	1.1934e-01	3.0222e-01	8.5560e-01	2.8442e+00
Cycle 2 EOC	1.3350e-02	3.1714e-02	1.1782e-01	3.0062e-01	8.5810e-01	2.8303e+00
Cycle 3 BOC	1.3350e-02	3.2186e-02	1.1937e-01	3.0230e-01	8.5560e-01	2.8456e+00
Cycle 3 EOC	1.3351e-02	3.1746e-02	1.1794e-01	3.0075e-01	8.5791e-01	2.8318e+00

The kinetics parameters vary depending on core conditions since the neutron spectrum and fissionable material change over the cycle and with a reload. The generation time Λ tends to have a larger value at EOC than BOC because of the increased percentage of fissions being provided by plutonium. There is approximately a 10% to 50% difference in prompt neutron lifetime. The variation in the total delayed neutron fraction is as large as 200 pcm–or around 30%. The fuel temperature coefficient is -2.318 pcm/K on average, and almost constant over the cycle. The moderator temperature coefficient of reactivity is relatively more sensitive than the other parameters. The coolant temperature coefficient varies from -4.867 pcm/K to -39.406 pcm/K. The equilibrium Xenon worth is on the order of -2400 pcm, and it does not have a significant variation depending on states.

From these calculations, we are able to estimate the overall variation of the key parameters affecting the reactor dynamics at various points in the reactor lifetime during its approach to equilibrium. The purpose of this is to understand and quantify the sensitivities of the reactor dynamics due to variations in the kinetics parameters. This understanding is necessary to inform the approach for learning based control described later. Essentially, whatever learning-based or adaptive control methodology that is adopted in place of a linear time-invariant model must be able to account for these variations. As will be shown in Section 6 we attempt to bound these effects by explicitly ignoring the xenon dynamics and providing proof of principle results that the GPR corrections to the state-space model can account for the exclusion of xenon effects.

Table 4. Temperature coefficients and Xenon worth

State	α_f (pcm/K)	α_c (pcm/K)	Xenon worth (pcm)
Cycle 1 BOC	-2.149	-4.867	-2561
Cycle 1 EOC	-2.380	-33.165	-2575
Cycle 2 BOC	-2.200	-10.629	-2312
Cycle 2 EOC	-2.467	-38.058	-2447
Cycle 3 BOC	-2.217	-13.597	-2228
Cycle 3 EOC	-2.493	-39.406	-2380

4. MODEL PREDICTIVE CONTROL

The MPC is an advanced method to control a process while satisfying a set of constraints [1]. It is based on iterative finite-horizon optimization of the system (i.e., trajectory optimization). A control input is computed for a relatively short time horizon in the future by evaluating a cost function to minimize an error between a desired set-point and predicted output. This calculation is then repeated at each subsequent instant or time-window.

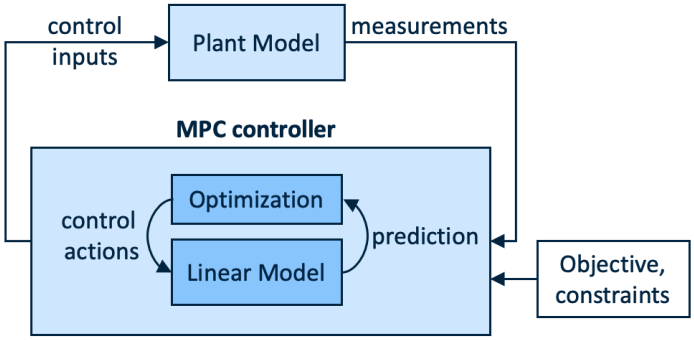

Figure 2. Diagram for reactor control with MPC

Fig. 2 shows a diagram for the reactor control problem using the MPC controller. In the diagram, the plant model is an actual plant or nonlinear model representing the plant. During the operation, measurement data such as power level may be observable. The measurement is used to calculate the objective cost function in the MPC controller. A linear state-space model is used in the MPC controller. It is normally a state-space model as described in Eq. (15). The optimization is a process to find the next control input to minimize the objective cost function. A quadratic cost function is used in the MPC. The cost function estimates an error compared to the reference trajectory and a cost used in the control action. A control input from the MPC controller is used to control the plant.

We briefly review the MPC theory in the following. A detailed description can be found in [5]. First, the continuous time state-space equation in Eq. (15) is converted to the discretized and augmented model [5] defined as:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\Delta\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) \ , \end{aligned} \tag{22}$$

where k is the discrete time index; $\mathbf{x}(k)$ and $\mathbf{y}(k)$ are the state vector and output vector of the discrete time and augmented system, respectively; $\Delta\mathbf{u}(k)$ is input change vector; and $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are the system matrix, input matrix, and output matrix, respectively.

The MPC solves an optimization problem that minimizes the quadratic cost function defined as follows:

$$J = (\mathbf{R}_s - \mathbf{Y})^T(\mathbf{R}_s - \mathbf{Y}) + \Delta\mathbf{U}^T\bar{\mathbf{R}}\Delta\mathbf{U} \ , \tag{23}$$

subject to required constraints i.e.,

$$\begin{aligned} \mathbf{u}_{min} \le \mathbf{u} \le \mathbf{u}_{max} \\ \Delta\mathbf{u}_{min} \le \Delta\mathbf{u} \le \Delta\mathbf{u}_{max} \ , \end{aligned} \tag{24}$$

where the first component $(\mathbf{R}_s - \mathbf{Y})^T(\mathbf{R}_s - \mathbf{Y})$ represents the objective of minimizing the errors between predicted output value and reference trajectory; and the second component $\mathbf{U}^T\bar{\mathbf{R}}\Delta\mathbf{U}$ represents the cost for input change. $\bar{\mathbf{R}}$ is an identity matrix multiplied by a weighting factor to adjust an importance between solution accuracy and the cost required in the input change. $\mathbf{Y}$ is series of output vectors in the prediction horizons, N_p as follows:

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}(k+1|k) & \cdots & \mathbf{y}(k+N_p|k) \end{bmatrix}^T \ , \tag{25}$$

where $\mathbf{y}(k+l|k)$ means the predicted output variable at $k+l$ with given information at time k. In a similar manner, $\Delta\mathbf{U}$ is series of input vectors in the control horizons, N_c:

$$\Delta\mathbf{U} = \begin{bmatrix} \Delta\mathbf{u}(k) & \cdots & \Delta\mathbf{u}(k+N_c-1) \end{bmatrix}^T \ . \tag{26}$$

$\mathbf{Y}$ is calculated as follows:

$$\mathbf{Y} = \mathbf{F}\mathbf{x}(k) + \mathbf{\Phi}\Delta\mathbf{U} \ , \tag{27}$$

where

$$\mathbf{F} = \begin{bmatrix} \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{N_p} \end{bmatrix} \ , \tag{28}$$

$$\mathbf{\Phi} = \begin{bmatrix} \mathbf{CB} & 0 & 0 & \cdots & 0 \\ \mathbf{CAB} & \mathbf{CB} & 0 & \cdots & 0 \\ \mathbf{CA}^2\mathbf{B} & \mathbf{CAB} & \mathbf{CB} & \cdots & 0 \\ \vdots & & & & \\ \mathbf{CA}^{N_p-1}\mathbf{B} & \mathbf{CA}^{N_p-2}\mathbf{B} & \mathbf{CA}^{N_p-3}\mathbf{B} & \cdots & \mathbf{CA}^{N_p-N_c}\mathbf{B} \end{bmatrix}. \tag{29}$$

Minimizing the cost function in Eq. (23) subjects to constraints in Eq. (24) is a representative mathematical optimization problem. The Quadratic Programming (QP) may be used to solve this optimization problem [6]. It should be noted that constraint can be applied to any of state vector, output vector, and input vector. Eq. (24) shows a simple example of the constraint.

So far, the standard MPC theory is derived based on the state-space model. This is why the linearization of the nonlinear model is necessary for the control algorithm. There is nonlinear MPC [7] that may be applied to solve a nonlinear optimization problem, but it is rarely used due to its difficulty and high cost in computing. There is another approach in MPC using step response of plant without constructing an explicit mathematical model. However, the approach requires many data points and it is more suitable for single-input and single-input problem while the MPC based on the state-space model can be readily applied to multiple-input and multiple-output system.

One of the strengths of MPC is its capability to accommodate a constrained problem. In nearly all practical applications, there are constraints imposed by the physical system that must be taken into account. In our application, two constraints are applied to the control rod position, $r(t)$, and control rod speed, $u(t)$. The control rod position is constrained between 0 to 224 steps. The control rod speed is constrained within ± 1 steps$/s$. The constraint for the rod speed is a conservative assumption. In the case of Westinghouse system, the maximum rod speed allowed is ± 1.2 steps$/s$ [8].

Table 5. Applied constraints to Nuscale reactor control

Parameters	Constraints
Control rod position (steps)	$0 \leq r(t) \leq 224$
Control rod speed (steps/s)	$-1.0 \leq u(t) \leq 1.0$

As shown in Fig. 2, it is not necessary to use the same model for the plant and MPC controller. The plant model is normally nonlinear and sometimes it is not possible to approximate the plant as a mathematical model. Therefore, potential mismatches between the two models may cause a significant error. To evaluate this point, a short transient scenario was simulated. The reference power scenario is shown in Fig. 3. The scenario includes 40 min simulation with 20%P_r/min of power ramp rate between transitions. The control input is calculated every 5 sec. The control rod is assumed to be partially inserted (100 steps) at the beginning of the simulation to ensure some degree of negative and positive reactivity from control action. The nonlinear model in Eq. (11) is used as a plant model, and the state-space linear model in Eq. (15) is used in the MPC controller. The state-space model was linearized at steady-state with 100%P_r. Therefore, the state-space model may have a large error when the state is far from the nominal condition.

Fig. 3 shows the power calculated by the MPC controller and the difference compared to the reference scenario. Even though there is a mismatch in the models, the MPC controller still follows

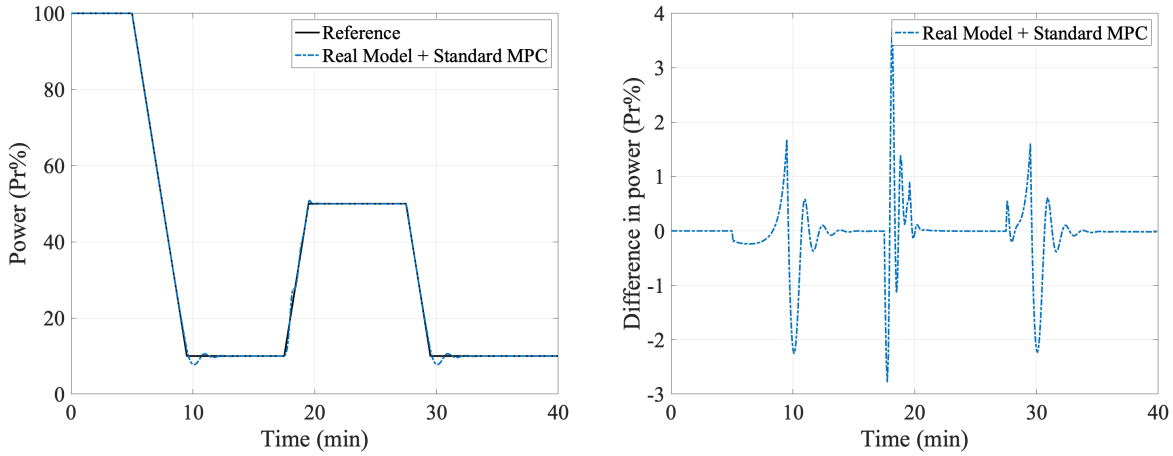

Figure 3. Standard MPC simulation results for power

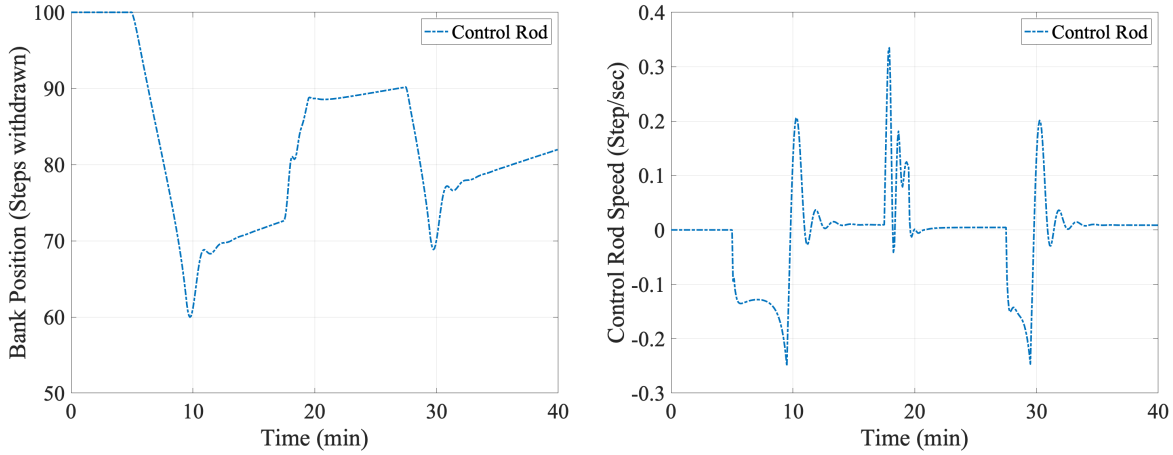

Figure 4. Standard MPC simulation results for control input

the reference power with reasonable accuracy. The maximum error in the power calculation is 3.7%P_r. Most of the errors are less than 2%P_r error. This good performance is due to the repetitive feedback algorithm in the MPC. If the time interval for the control action and measurement feedback is shorter than 5 sec, the error can be reduced further.

There are additional results for several state variables and control inputs through Figs. 4 and 5. The control rod position and rod speed are constrained within the given constraints in Table 5. The reactivity components calculated with Eq. (8) are shown in the Fig. 6. The Xenon worth at HFP condition is an order of -2400 pcm which is one of the significant reactivity components. The half-life of ^{135}I is 6.7 hours so that ^{135}I and ^{135}Xe concentration do not converge with in this 40 min short transient. Since the reactivity from ^{135}Xe varies slowly, appropriate control rod input should be calculated continuously.

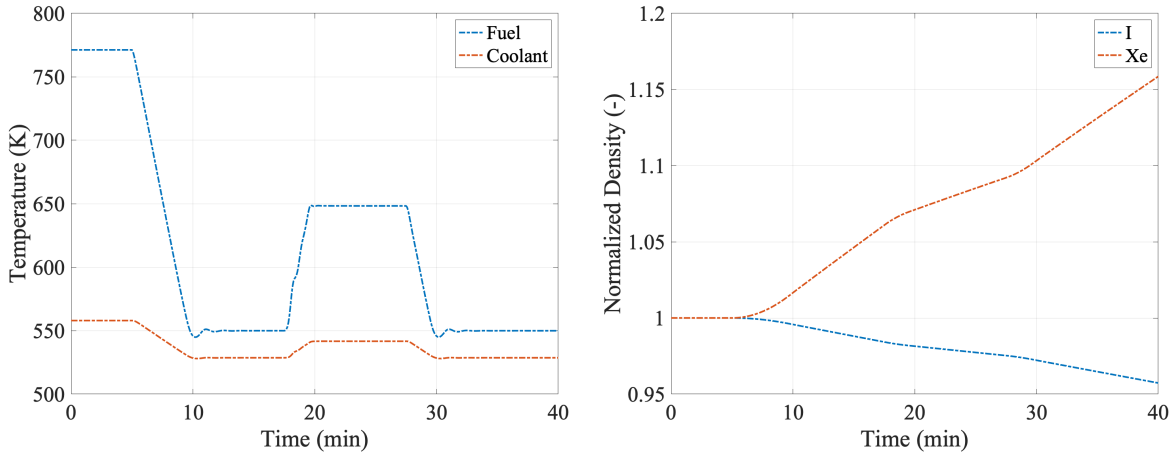

Figure 5. Standard MPC simulation results for temperature and density

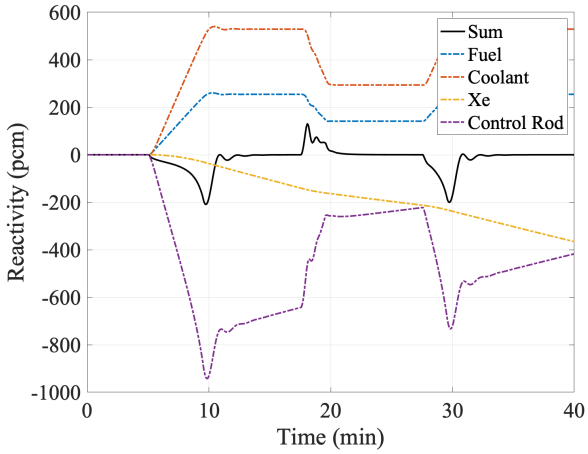

Figure 6. Standard MPC simulation results for reactivity

5. LEARNING-BASED MODEL PREDICTIVE CONTROL

As discussed in Section 4, it is necessary to approximate the plant model as a linear state-space model for the standard MPC. However, the actual plant model is generally nonlinear so that there is a mismatch between the actual plant model and the model used in the MPC controller. The MPC controller behaves well even with the mismatched models as discussed in Section 4, but there is room to improve the accuracy, and this may not necessarily be the case for every transient power scenario or plant design. In this section, we introduce a ML technique to improve the accuracy of the MPC controller. We call the ML based MPC method as Learning-Based MPC (LBMPC) [9, 10].

Fig. 7 describes LBMPC algorithm. The MPC gives a very accurate solution (or control input) as long as the model used in the MPC represents the actual model accurately. Therefore, the focus of LBMPC is to improve the linear model used in the controller by using a ML technique. We expect the MPC controller with an improved linear model will calculate a more reliable control input.

In the ML, it is necessary to provide training data. The measurements (or control input if necessary) from the plant model are used in the ML. The training data can be any state variables, control inputs, measurements, and error. In this work, we used supervised learning as the initial approach. Supervised learning is generally less challenging than other categories of ML from the viewpoint of the amount of training. Therefore, it may be possible to realize the ML in practice. A disadvantage of supervised learning is that it is necessary to provide correct sets of input and output information to ML. Therefore, one must have a good knowledge of the output to train the model. In the case of reactor control, it is necessary to provide training data and a way to improve the linear model. Regression methods can be used to find correlations in the training data and provide the corrections. In this work, we initially adopted GPR because typically in an operating reactor the observations will have Gaussian noise. By using GPR in our calculations we assume that the nature of the mismatch between the real model and the state-space model is due to Gaussian processes–which is not necessarily true, since in some cases we have a specific partial differential equation representing some dynamics. However, insofar as we are concerned with the coefficients in the dynamics model, assuming the coefficients vary as a Gaussian distribution is reasonable.

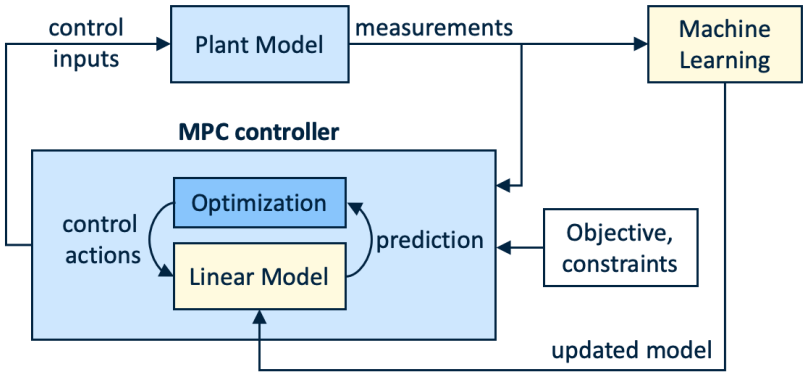

Figure 7. Diagram for reactor control with LBMPC

In the following, we briefly introduce the GPR. A detailed description can be found in [11, 12]. The GPR is a nonparametric approach to regression that is being widely used in the area of ML. Given a set of training data $(w_i, z_i)_{i=1...N}$ the GPR aims to find a function g that can associate w_i with z_i by setting appropriate Gaussian Process (GP) parameters (e.g. mean and variance). Taking noise into consideration for the output z_i, the relationship between the latent function g and z_i can be written as

$$z_i = g(w_i) + \epsilon \ , \tag{30}$$

where ϵ is an independent Gaussian noise. The GPR treats the function g as a collection of random variables. Therefore, a Gaussian distribution of the function g can be completely specified by its mean function $M(w)$ and covariance function $K(w, w')$:

$$g(w) \approx \mathrm{GP}(M(w), K(w, w') \ , \tag{31}$$

where $K(w, w')$ represents the covariance matrix between each data point in w and w^*.

The mean function $M(x)$ is considered to be zero in most real problems. The joint distribution of the observed target z is given as:

$$z \approx \mathrm{GP}(0, K(w, w') + \sigma_n^2 I_n) \ , \tag{32}$$

where σ_n^2 is the variance in the observed noise ϵ, I_n is an identity matrix. Hence, given the test input data w and the test output z, according to the definition of GPR, the joint probability distribution of training output z and the test output z^* under the prior can be written in a matrix form as:

$$\begin{bmatrix} z \\ z^* \end{bmatrix} \approx \left(0, \quad \begin{bmatrix} K(w,w) + \sigma_n^2 I_n & K(w,w^*) \\ K(w^*,w) & K(w^*,w^*) \end{bmatrix}\right) \ . \tag{33}$$

The posterior distribution over z can be obtained by imposing restrictions on the prior joint distribution Eq. (33) given the training and test point, which is expressed as:

$$P(z^*|w,z,w^*) \approx N(\bar{z}^*, \mathrm{cov}(z^*)) \ , \tag{34}$$

where

$$\bar{z}^* = K(w^*,w)[K(w,w) + \sigma_n^2 I_n]^{-1} z \ , \tag{35}$$

$$\mathrm{cov}(w^*) = K(w^*,w) - K(w^*,w)[K(w,w) + \sigma_n^2 I_n]^{-1} K(w,w^*) \ , \tag{36}$$

where z^* and $\mathrm{cov}(w^*)$ are the predicted mean and the variance of output z^*, respectively. It is also observed that the variance $\mathrm{cov}(w^*)$ only depends on the observed inputs, not on w and w^*.

Finally, the GPR model can predict the output of test points based on the mean function, variance, and training points. Consequently, the expected prediction results (mean) and the uncertainty of measurement information (variance) of the real output is given by this unique property of the GPR. There is further discussion about the input and output data sets in Section 6.

6. DEMONSTRATION OF LEARNING-BASED MODEL PREDICTIVE CONTROL

6.1 LBMPC for Time-varying State-space Model

It is important to select appropriate training data for the GPR. A concept of the adaptive MPC [13] is used in the output training data generation. In the adaptive MPC, a nonlinear model is successively linearized for each time step, and the linearized model is used in the MPC controller. In other words, the adaptive MPC uses a series of piecewise linearized models from the nonlinear model. To observe how the model varies during the simulation, the nonlinear model Eq. (11) was linearized during the simulation, and then some of the elements were compared in Fig. 8. Fig. 8 shows the elements that change more than 0.1% compared to those nominal (or initial) values.

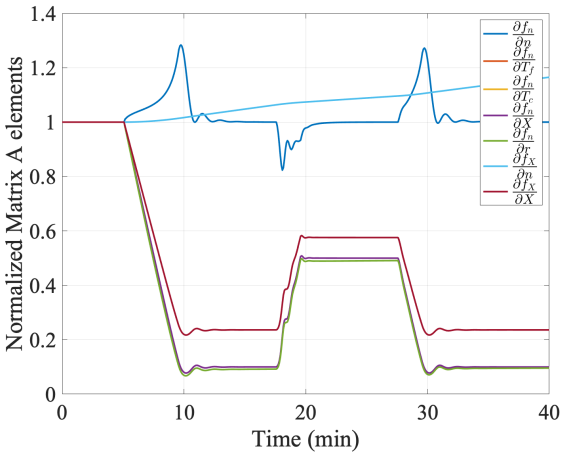

Figure 8. Matrix A elements from successive linearization

These data will be used as output data set since this is the actual linear model at each time step. Some state and control input variables related to the chosen output data set are selected as input data set. The input data, **w**, and the output data, **z** are summarized as follow:

$$\mathbf{w} = \begin{bmatrix} n(t) & \rho(t) & r(t) & \frac{dr(t)}{dt} & X(t) \end{bmatrix}, \tag{37}$$

$$\mathbf{z} = \begin{bmatrix} \frac{\partial f_n}{\partial n} & \frac{\partial f_n}{\partial T_f} & \frac{\partial f_n}{\partial T_c} & \frac{\partial f_n}{\partial X} & \frac{\partial f_n}{\partial r} & \frac{\partial f_X}{\partial n} & \frac{\partial f_X}{\partial X} \end{bmatrix}. \tag{38}$$

Fig. 9 shows the results from LBMPC. There are three simulations for the same reference scenario. In the first generation, the GPR correction is not used in updating the linear model in the MPC controller. Instead, the input and output data defined in Eqs. (37) and (38) are collected during the simulation. At the end of the first simulation, the GPR fitting was performed to make the GPR model. In the second simulation, the GPR model trained in the previous simulation was used to estimate the linear model at every time step. The second generation of the LBMPC controller gives a noticeable improvement in the power calculation. The maximum error compared to the reference trajectory is about $1\%P_r$. The third simulation shows almost identical results with those of the second generation. That means the training data from the first simulation is sufficient to estimate the time-varying linear model.

To verify if the GPR calculates the matrix element accurately, the GPR outputs are compared to the matrix elements from linearization of the nonlinear equation Eq. (11). The comparison results are presented in Fig. 13. In the figure, the "Linearization" means the results from linearizting the nonlinear eqution at the second simulation of the LBMPC. The GPR model was made based on the first simulation, and then the model was used to calculate the matrix element in the second simulation. As presented in the figure, the GPR calculates the matrix elements very accurately.

At the beginning of Section 6, it was discussed that the LBMPC implementation in this work is based on the concept of adaptive MPC. In other word, the LBMPC imitates the adaptive MPC therefore the best accuracy can be achieved by the LBMPC is the that of the adaptive MPC. For verification,

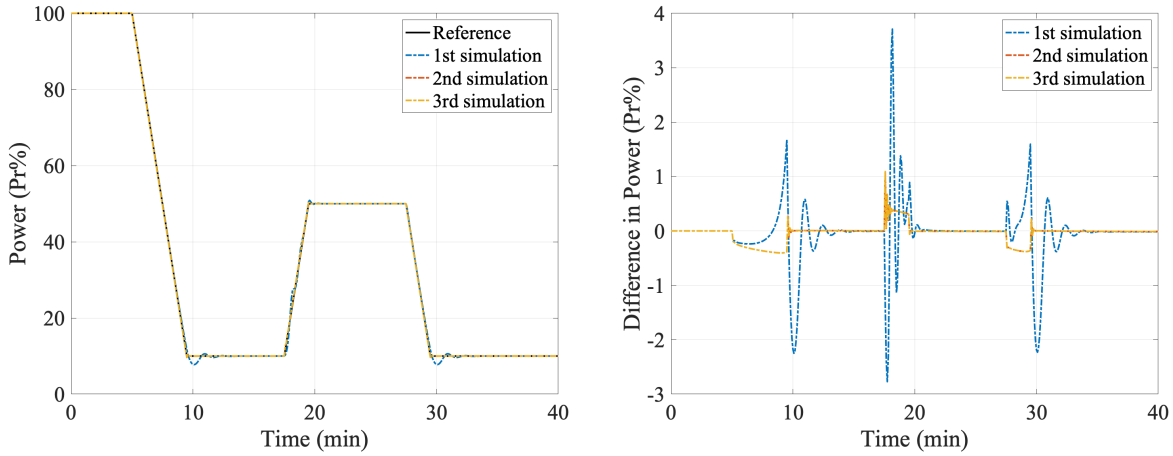

Figure 9. LBMPC simulation results for power

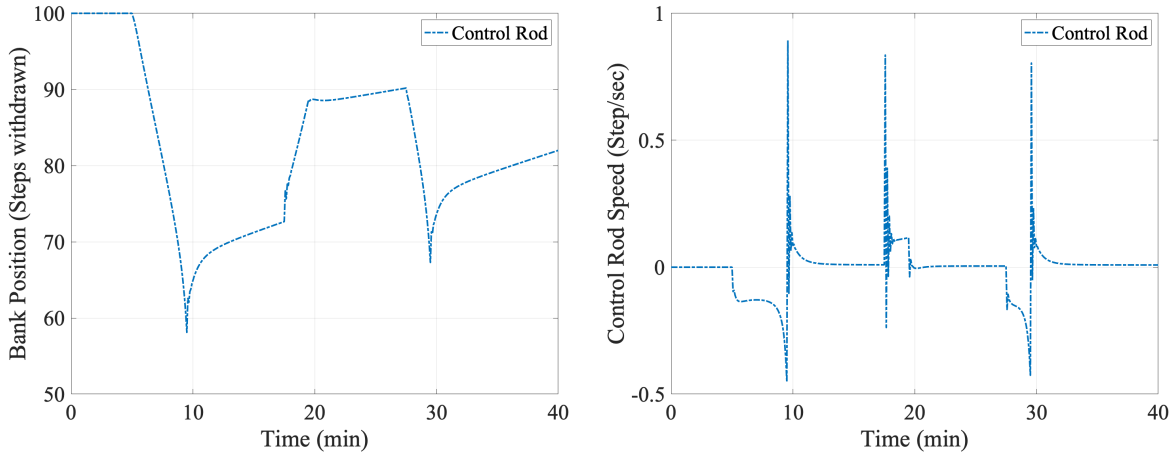

Figure 10. LBMPC simulation results for control input

the adaptive MPC and LBMPC are compared in Fig. 14. As shown in the figure, the LBMPC and the adaptive MPC calculated an almost identical power, and the difference between them is hardly noticeable. Therefore, it is concluded that the LBMPC was successfully implemented and showed the best accuracy as it can.

6.2 LBMPC for Measurement Error Estimation

In Section 6.1, the GPR was used to improve the linear model for the MPC controller by studying the linear model from successive linearization of actual nonlinear model. The approach showed an accurate result, but there is a limitation. The approach requires a knowledge of original mathematical or numerical model since it is necessary to calculate the piecewise state-space models. In practice, it is rarely possible to obtain the nonlinear model representing the actual plant. Therefore, this section uses a different approach for the LBMPC. The idea is to add an correction term in the state-space model for the MPC controller as follow:

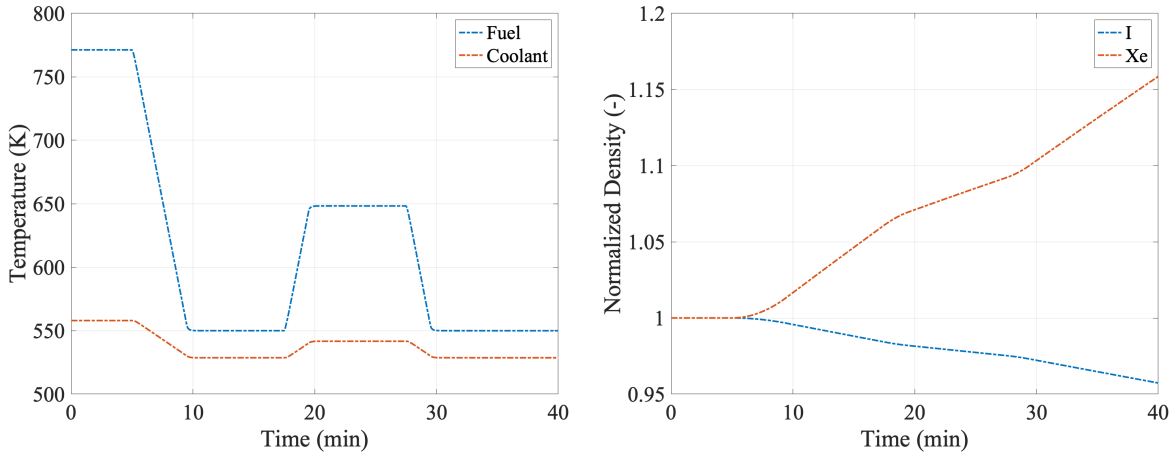

Figure 11. LBMPC simulation results for temperature and density

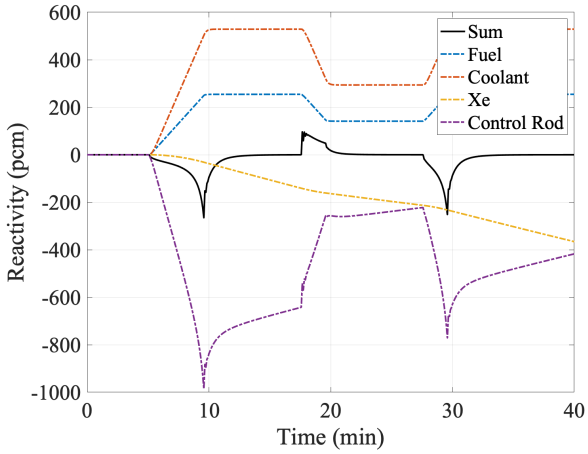

Figure 12. LBMPC simulation results for reactivity

$$\dot{\mathbf{x}}_s(t) = \mathbf{A}_s\mathbf{x}_s(t) + \mathbf{B}_s\mathbf{u}_s(t) + \mathbf{e}(t) \ , \tag{39}$$

where $\mathbf{e}(t)$ is the correction term defined as difference like: $\mathbf{e}(t) = \dot{\mathbf{x}}_{\text{true}}(t) - \dot{\mathbf{x}}(t)$, and $\mathbf{x}_{\text{true}}(t)$ is the time derivative of state vector of actual model with assuming that all state variables are observable. To implement the $\mathbf{e}(t)$ in the MPC, it is necessary to form it as the original state-space model. Therefore, the state-space model is extended to add an hidden variable as follows:

$$\begin{aligned} \dot{\mathbf{x}}_h(t) &= \mathbf{A}_h(t)\mathbf{x}_h(t) + \mathbf{B}_h\mathbf{u}(t) \\ \mathbf{y}_h(t) &= \mathbf{C}_h\mathbf{x}_h(t) \ , \end{aligned} \tag{40}$$

where

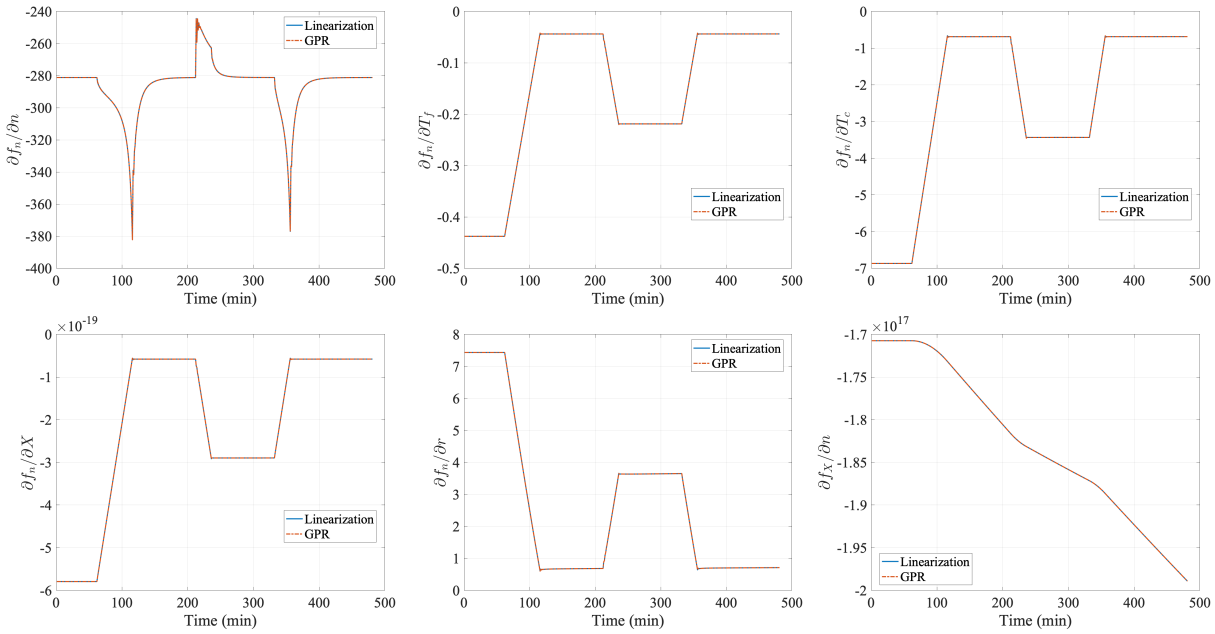

Figure 13. Comparison of GPR outputs

$$\begin{aligned} \mathbf{x}_h(t) &= \begin{bmatrix} \mathbf{x}_s(t) \\ 1 \end{bmatrix}, \mathbf{y}_h(t) = \begin{bmatrix} \mathbf{y}_s(t) \\ 0 \end{bmatrix}, \\ \mathbf{A}_h(t) &= \begin{bmatrix} \mathbf{A}_s & \mathbf{e}(t) \\ \mathbf{o}_m & 1 \end{bmatrix}, \mathbf{B}_h = \begin{bmatrix} \mathbf{B}_s \\ 0 \end{bmatrix}, \mathbf{C}_h = \begin{bmatrix} \mathbf{C} & 0 \end{bmatrix}, \end{aligned} \tag{41}$$

where $\mathbf{o}_m$ is $1 \times m$ null vector.

The training input and output are

$$\begin{aligned} \mathbf{w} &= \left[\mathbf{x}_{\text{true}}(t)\right] \\ \mathbf{z} &= \left[\mathbf{e}(t)\right] \end{aligned}. \tag{42}$$

The result of LBMPC based on the error estimation is shown in Fig. 15. Similarly as Section 6.1, the training data was collected during the first simulation, and then the GPR was used to calculate the output. Unfortunately, the LBMPC with the error correction does not work properly. From the second simulation, significant oscillation was calculated due to the error correction term. There are several guesses about this result. First, this approach should be used when the $\mathbf{e}(t)$ has a monotonic behavior. Since MPC controller calculates the control input by solving cost function for multiple number of prediction horizons, the state-space model should have appropriate predictions for entire prediction horizons as well as a next time interval. However, $\mathbf{e}(t)$ varies a lot as time evolves, so that it seems the MPC controller calculates oscillating power due to underestimated and overestimated control inputs. Second, the $\mathbf{e}(t)$ varies a lot as time evolves, and the behavior is not monotonic so that it is relatively difficult to develop the well trained GPR model. The $\mathbf{e}(t)$ from the GPR model

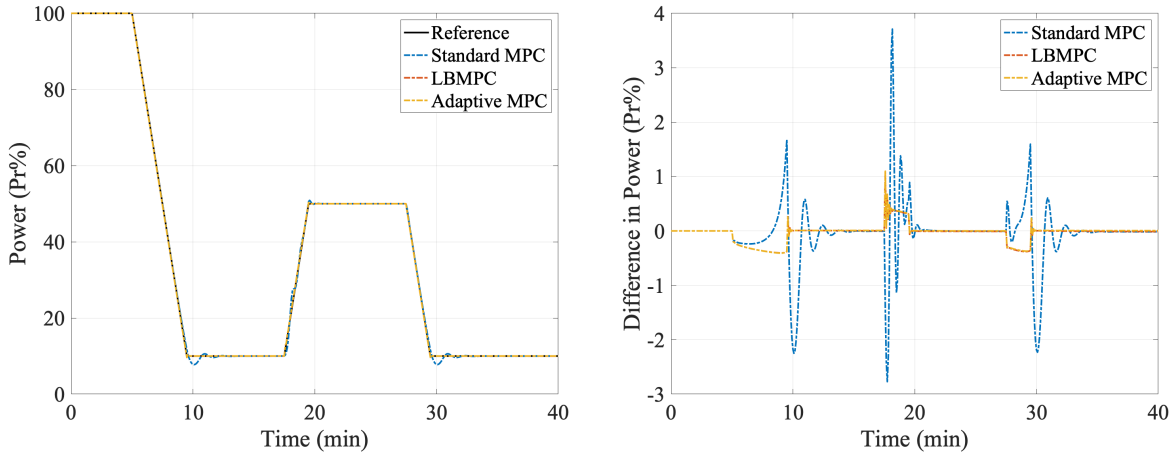

Figure 14. Comparisons of MPCs

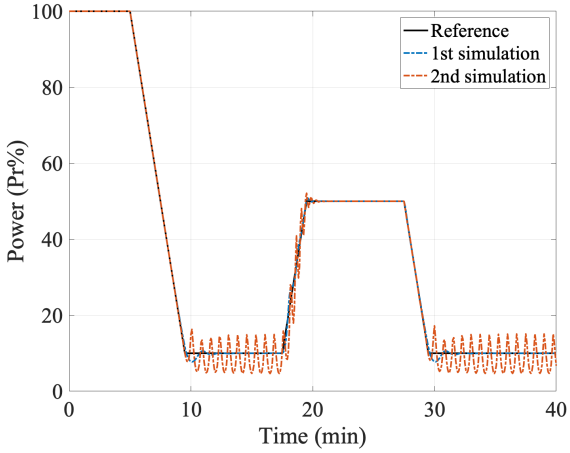

Figure 15. LBMPC with error estimation.

frequently gives a value with very high standard deviations if the GPR model is used for a newly encountered reactor condition.

From this work, it was not possible to realize the LBMPC using the error estimation. However, it is worthwhile to keep developing this approach since it is a more generalized approach and may be used in a wider application. In the future work, this approach may be improved by using a different form of error correction. Instead of adding whole error correction term to the $\dot{\mathbf{x}}(t)$, there may be a better way to decompose the error term into several pieces of functions of state vectors. The Support Vector Machine (SVM) [14] may be used in this approach to classify where an error comes from and which state vector should be adjusted.

7. CONCLUSIONS & FUTURE WORK

In this report, the reactor dynamics model with the PKE, TH model, and Xe dynamics were developed for use with an LBMPC controller to automatically control the SMR. MPACT was used to generate parameters for the PKE and the reactivity model. The parameters were generated for BOC and EOC of 1 – 3 cycles. The generation time and the coolant temperature coefficients were relatively sensitive than other parameters. The MPC controller was developed to simulate the plant model which has nonlinear reactor dynamics. For the standard MPC controller, the state-space model was generated using linearizing the nonlinear model at the initial steady-state model. There was model mismatch between plant model and controller's model. However, the accuracy of the standard MPC controllers was still reasonable with less than $4\%P_r$. To improve the controller accuracy further, the LBMPC controller was developed. In the LBMPC, the GPR was used to train time-varying components of state-space model generated by piecewise linearization of the nonlinear model. The trained GPR model is used to imitate the time-varying state-space model. The updated state-space model for each time step was used in the MPC controller by providing a better estimation for the plant model. As a result, the LBMPC controller improved results resulting less than $1\%P_r$ error in most of time steps.

From this work, several opportunities for future studies were identified.

- The LBMPC controller learning model error is necessary to develop for a more general purpose.
- There is room to refine and improve the nonlinear model to approximate the actual SMR with a higher accuracy. i.e., incorporating measurement noise and models for systems such as pressurizer.

ACKNOWLEDGEMENTS

This work was supported by funding received from the DOE Office of Nuclear Energy's Nuclear Energy University Program under contract number DE-NE0008975.

REFERENCES

[1] 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).

[2] K. Frick and S. Bragg-Sitton. "Development of the NuScale Power Module in the INL Modelica Ecosystem." *Nuclear Technology*, **volume 207**(4), pp. 521–542 (2021).

[3] "MPACT Theory Manual, Version 2.2.0." Technical report, Consortium for Advanced Simulation of Light Water Reactors (CASL) (2016).

[4] B. Kochunas, B. Collins, S. Stimpson, R. Salko, D. Jabaay, A. Graham, Y. Liu, K. S. Kim, W. Wieselquist, A. Godfrey, et al. "VERA core simulator methodology for pressurized water reactor cycle depletion." *Nuclear Science and Engineering*, **volume 185**(1), pp. 217–231 (2017).

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

[6] M. Frank, P. Wolfe, et al. "An algorithm for quadratic programming." *Naval research logistics quarterly*, **volume 3**(1-2), pp. 95–110 (1956).

[7] F. Allgöwer and A. Zheng. *Nonlinear model predictive control*, volume 26. Birkhäuser (2012).

[8] "Westinghouse Technology Systems Manual, Section 8.1: Rod Control System." Technical report, US Nuclear Regulatory Commission (USNRC) (2012). URL `https://www.nrc.gov/docs/ML1122/ML11223A252.pdf`.

[9] P. Bouffard, A. Aswani, and C. Tomlin. "Learning-based model predictive control on a quadrotor: Onboard implementation and experimental results." In *2012 IEEE International Conference on Robotics and Automation*, pp. 279–284. IEEE (2012).

[10] J. Kabzan, L. Hewing, A. Liniger, and M. N. Zeilinger. "Learning-based model predictive control for autonomous racing." *IEEE Robotics and Automation Letters*, **volume 4**(4), pp. 3363–3370 (2019).

[11] B. Jiang, J. Zhou, X. Huang, and P. Wang. "Prediction of critical heat flux using Gaussian process regression and ant colony optimization." *Annals of Nuclear Energy*, **volume 149**, p. 107765 (2020).

[12] C. E. Rasmussen. "Gaussian processes in machine learning." In *Summer school on machine learning*, pp. 63–71. Springer (2003).

[13] 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).

[14] W. S. Noble. "What is a support vector machine?" *Nature biotechnology*, **volume 24**(12), pp. 1565–1567 (2006).