

Overview

My system of interest for applying a hybrid modeling framework would be a recent small modular nuclear reactor (SMR). These systems have been receiving lots of attention lately in the nuclear community due to their size, safety, and reduced costs. At a high level, from what I understand about these reactors, I would aim to combine the first principles/physics based models for fundamental reactor physics such as neutronics and thermal distributions, while using data-driven deep neural networks for improved handling anomalies and uncertainties that may develop between the ideal reactor model and the real-world reactor process. Recent research suggests that this is indeed an area of interest in improving the modeling and simulation of various reactor system designs [[Hybrid mechanistic and neural network modeling of nuclear reactors](#)], [[Prediction of the evolution of the nuclear reactor core parameters using artificial neural network](#)].

Suitability of hybrid modeling for nuclear reactor design

From my surveying of recent SMR design and research, I found that SMRs involve complex interactions between neutronics, thermal hydraulics, and fuel behavior, usually modeled using FP approaches. However, uncertainties often appear to arise, especially in novel designs with advanced coolants like molten salt or gas (SMRs appear to vary considerably in coolant types). In addition, there are spatiotemporally varying parameters within the reactor, similar to the ones we discussed in class for the chemical process system.

Variables and parameters to be predicted by the DNN

In essence, the hybrid framework's ability to handle spatiotemporally varying parameters can be applied to model neutron diffusion and thermal gradients in SMRs, helping to capture some details that are hard to model by a FP only approach. This appears to be particularly relevant for new SMRs with modular, factory-assembled components, where traditional models may lack accuracy. From some searching, it appears there are a few key variables in the reaction process of many SMRs that could be well suited for DNN prediction:

- **Reactivity (ρ):** How the reactor responds to changes, influenced by temperature and control rods.
- **Effective Multiplication Factor (k_{eff}):** the ratio of neutrons produced to those lost, directly related to reactivity via $\rho = \frac{k_{eff} - 1}{k_{eff}}$
- **Power distribution:** Where power is generated within the reactor core
- **Temperature distributions:** Fuel and coolant temperatures, critical for safety

Integrating the DNN with a first principles model

The following is an example first principles point kinetics model for nuclear reactors:

$$\frac{dn}{dt} = \frac{\rho - \beta}{\Lambda} n + \sum_{i=1}^k \lambda_i C_i$$

where n is neutron density,

ρ is reactivity,

β is the delayed neutron fraction,

Λ is the neutron generation time, and

C_i and λ_i are precursor concentrations and decay constants

Reactivity is a measure of the reactor's deviation from criticality, and is influenced by factors like core temperature, coolant temperature, void fraction, control rod positions, and burnup. It can be modeled with a DNN:

$\rho = NN(T_{core}, T_{coolant}, V_{void}, rod\ positions, B, \dots)$ where NN is the neural network.

Effective Multiplication Factor is directly related to reactivity and research has shown that feed forward ANNs can predict it quickly and effectively for a given loading pattern:

$k_{eff} = NN(fuel\ loading, control\ rods, burnup, \dots)$

Aside from the equation above, DNNs could be used to predict other properties as well. Power distribution is the spatial distribution of power density in the reactor core, crucial for ensuring uniform operation and preventing hotspots. My findings suggest that CNNs can be applied to predict 2D or 3D power distributions based on core geometry and operational data:

$P(r, z) = NN(fuel\ type, burnup, control\ rods, \dots)$ where $P(r, z)$ is the power distribution

Finally, temperature distributions for fuel and coolant temperatures can be integrated in a similar way:

$T_{fuel}(r, z) = NN(P(r, z), coolant\ flow, coolant\ properties, \dots)$