Balance of Plant Network Simulation for a Boiling Water Nuclear Reactor

Summary

A network model for system-level nuclear power production was developed using the Python package Cortix (https://cortix.org). The system consists of various computational modules, namely reactor, turbines, condenser, and pump. These modules are effectively Python modules connected through Cortix using either the MPI library (message passing interface; mpi4py) or the Python multiprocessing package (using pipes) to send messages between modules executed in parallel. The modules solve a system of coupled ordinary differential equations (ODE's) evolving in time to account for neutron density, reactivity feedback, nuclear heat generation, heat transfer and power generation. Extensive use of solvers and data arrays is made through NumPy and SciPy.

Abstract

Nuclear reactors are extremely complicated pieces of equipment to model due to the sheer number of interconnected variables which affect its performance. Power, heat generation, xenon buildup and overall reactivity all simultaneously affect each other. Additionally, the reactor's balance of plant can further affect the operation of the reactor itself through changes in the inflow and outflow properties of whatever coolant is being used. While there are many computational models which allow exploration of disturbances on the reactor itself, there are few time-dependent models which exam the operation of the system as a whole. Our research attempts to fill this gap by developing and solving a time-dependent model of the energy, nuclides and steam produced by a boiling water reactor, which then exchanges information on the properties of inflow and outflow light water to models of the balance of plant unit operations (namely pumps, turbines and condensers).

We first developed a point-reactor model for the boiling water reactor, which solves a number of coupled ODEs to produce time-dependent solutions for the following variables: average nuclear fuel temperature, state and composition of the liquid water and steam entering and leaving the reactor, total nuclear heat generation and absorption by the coolant, burnup of fuel and the density of neutrons, delayed neutron emitters and fission product poisons within the reactor. The model is realized into a computational module (Python) using SciPy tools for solving the ODEs.

Similarly, other models for the turbines, pumps and condensers, found within the reactor's balance of plant (BOP), were implemented as Python modules. In order to effectively link the complicated system together, we employed the network simulation library Cortix to link the developed models in such a way that they could easily and seamlessly pass time-dependent information between each other. This is accomplished by using parallel processing while executing each module in a separate process or thread. Message passing (MPI, mpi4py or Python multiprocessing, pipes) is used for exchanging time-stamped information between modules in the network.

Once fully developed, the BOP system took information about the state of the outflow coolant generated by the BWR model and used it to determine the amount of usable power produced by

the turbines and the state of the subcooled liquid which would be passed back into the BWR. This system was then verified against existing operating data for the balance of plant for commercial BWR's.

With models for all the individual unit operations completed and then properly linked together into a complex network using Cortix, the whole network simulation could be used to examine the effects that startup, shutdown and malfunctions could have on the feedback of the entire system. The final version of our work is an excellent simulation tool for that can be used to explore coupled phenomena in a practical and tractable computational environment. (https://github.com/dpploy/cortix-nb)