UNENE 2010 – Class Project

Objective:

To develop a simple reactor model to demonstrate fundamental characteristics.

Purpose:

It will be used by UNENE as a learning tool for students in the M.Eng. Classes.

Long Term Vision:

The intent is to model steady state and slow transients, assuming uniform and unidirectional flow, as follows:

Reactor Physics

- Point kinetics, reactivity coefficients (fuel temperature, coolant temperature, coolant void) as data inputs
- Delayed neutrons, photo-neutrons
- Xenon production & decay
- Decay power transient model
- Tabular model of shutdown and stepback reactivity versus time
- Trips, setback, stepback initiated from external control routine
- Inputs: initial power, external reactivity transient, fuel and coolant temperature, coolant void
- Output: neutron power (total including decay power)

Channel Thermohydraulics

- Equilibrium (single temperature, pressure, velocity)
- Single and two-phase water/steam
- Conservation of mass and energy, not momentum
- Energy added per segment from fuel, from/to pressure tube
- Fixed flux shape
- Unidirectional forward flow, same in all segments

- 12-segment model, one per bundle, quasi-steady-state, one reactor-average channel
- Inputs: inlet temperature, flow, heat transfer from fuel & from/to pressure tube
- Outputs: outlet temperature, quality, pressure drop

System Thermohydraulics

- Equilibrium (single temperature, pressure, velocity)
- Single and two-phase water/steam
- Conservation of mass and energy, not momentum
- Unidirectional forward flow, same in all nodes
- Quasi-steady-state
- Heat removal from steam generators
- Constant secondary side (or input data table if changes)
- Pump head/flow curve as input
- Inputs: core pressure drop, core outlet temperature & quality
- Outputs: flow, inlet temperature

Fuel and Pressure-Tube Heat Transfer

- Single bundle-average pin (for each channel segment)
- One-dimension radial heat transfer
- Fuel and sheath heat conduction
- Convection from sheath to coolant
- Convection from coolant to pressure-tube
- Radiation from pressure-tube to calandria tube [optional]
- Conduction from calandria tube to moderator (fixed moderator input temperature)
- Inputs: coolant conditions, reactor power
- Outputs: fuel & sheath temperature, heat transfer to coolant from fuel and from/to pressure tube, pressure-tube temperature, calandria tube temperature [optional]

“Control”

- Since CANDUs have a near-zero power coefficient, the steady state will not be stable. A very simplified “control” (feedback) model is needed – *not* a simulation of RRS. The routine should look at the power error and output the appropriate compensating reactivity to hold it “steady”. Choice of time constant (lag) will be important. The routine should be tested or integrated with the physics module so it can be demonstrated to work.

- The module should also have triggers so it can flag a reactor trip to the physics routine if external parameters passed to it exceed setpoints – e.g. if the flow (external input) decreases below a certain value (data), it sends a 'trip' flag to the physics routine (external output).
- Inputs: reactor power setpoint, reactor power, values of trip variables
- Outputs: feedback reactivity to stabilize power, trip flags

Engineering Model Integrator

- Confirm & match engineering models
- Confirm interfaces, match inputs and outputs
- Review models
- Define test cases (separate & integrated) & evaluate

Numerical Model Integrator

- Choose language (e.g. MATLAB, FORTRAN, C++, Excel.)
  - Note: MATLAB chosen March 14 2010.
- Choose solution scheme
- Set up programme structure
- Define data structure, inputs & outputs (no hardwired data!)
- Define graphical outputs
- Define numerical tests
- Review code

Ground-rules & Advice

- OK to consult with colleagues, but state their contribution
- Do NOT use proprietary data or models; this will be an open model
- Test runs for each module

Future

- Add LWR model

Suggested Scope – This Year

- Form groups to develop the five modules, but do not try to integrate
  - Physics, system thermohydraulics, channel thermohydraulics, fuel, control
- Complete each module, test, verify, and run sample cases
- Required: Common computer language, hooks for accepting/reading data as per flowchart
- Adjust scope if it turns out to be too complex or too simple
- Suggested time lines – by end of:
  - Weekend 1 – form groups, assign roles, choose *common* software. Ideally each group should have someone knowledgeable (or keen) in the technical area, and someone with programming experience.
  - Weekend 2 – equations defined, method defined, inputs/outputs defined, programme block structure defined; short *in*formal discussion in class. Sanity check to ensure scope is achievable; adjust if not.
  - Weekend 3 – code written, tested, sample cases defined; short *in*formal discussion in class
  - Weekend 4 – final report submitted, formal presentation to class