SIMULATION-BASED REACTOR CONTROL DESIGN METHODOLOGY for CANDU 9

M.K. Kattan, M.J. MacBeth
Atomic Energy of Canada Limited
446A Second Ave. North
Saskatoon, Saskatchewan, Canada, S7K 2C3

W.F. Chan, K.Y. Lam
Cassiopeia Technologies Incorporated
1071 King Street West, Suite 202
Toronto, Ontario, Canada, M6K 3K2

ABSTRACT

The next generation of CANDU nuclear power plant being designed by AECL is the 900 MWe CANDU 9 station. This design is based upon the Darlington CANDU nuclear power plant located in Ontario which is among the world leading nuclear power stations for highest capacity factor with the lowest operation, maintenance and administration costs in North America. Canadian-designed CANDU pressurized heavy water nuclear reactors have traditionally been world leaders in electrical power generation capacity performance.

This paper introduces the CANDU 9 design initiative to use plant simulation during the design stage of the plant distributed control system (DCS), plant display system (PDS) and the control centre panels. This paper also introduces some details of the CANDU 9 DCS reactor regulating system (RRS) control application, a typical DCS partition configuration, and the interfacing of some of the software design processes that are being followed from conceptual design to final integrated design validation.

A description is given of the reactor model developed specifically for use in the simulator. The CANDU 9 reactor model is a synthesis of 14 micro point-kinetic reactor models to facilitate 14 liquid zone controllers for bulk power error control, as well as zone flux tilt control.

1. INTRODUCTION

In previous CANDU reactors, plant control was performed by control computers, analog devices and relay logic. System control was performed by central dual redundant computers which executed a set of control programs for monitoring, annunciation, and control of plant systems. For the lower level control functions, devices such as analog controllers and programmable logic controllers were utilized. The application programs for the higher level control computers were written in programming languages such as assembler. The lower level device control logic computer applications were written in a symbolic language while the remaining functions were performed by hardwired logic.

The CANDU 9 plant monitoring, annunciation, and control functions are implemented in two evolutionary systems; the distributed control system (DCS) and the plant display system (PDS). The DCS [1] implements most of the plant control functions on a single hardware platform while PDS similarly implements the main control room display and annunciation functions [2]. The DCS communicates with the PDS to provide the main operator interface and annunciation capabilities of the previous control computer designs but is unconstrained by previous computer memory and execution cycle limitations.

The DCS is made up of several partitions with minimum interfacing between each other. A typical partition is illustrated in Figure 1. Plant control functions are assigned to the individual DCS partitions, based on an independence assessment of the control programs. DCS application programs are implemented using a function block language for all hierarchial levels in the control system. The use of a function block programming language (FBL) and the application of software engineering quality assurance procedures lead to an efficient, structured and more auditable software development process.

As the design proceeds, the CANDU 9 project design process requires that system designers be able to update and revise the plant model, under the configuration control of the Process Control Section. The simulation tool and platform selected for this application required several key features, including:

- ability to execute on a PC platform
- upgradable
- non-simulation expert and user friendly
- interfacable to industrial standard devices (TCP/IP,serial,etc)

The functionality of the simulation-linked DCS prototype provides a dynamic mechanism for on-going evaluative design activities by control system designers throughout the entire design project life-cycle. During the project design integration phase, a full DCS partition will be prototyped and interfaced to the CANDU 9 control centre mockup facility[2]. The CANDU 9 plant simulator [3] is used as a verification and validation tool for the displays, annunciations and operator interfaces as well as in the DCS design process. Simulation is used for control strategy development, analysis of overall plant control performance, estimates of tuning parameters for major control loops, etc. During the project phase, the plant control system software for individual partitions will be developed, verified and validated. Each DCS partition, in conjunction with the CANDU 9 plant simulator, will be used to perform the dynamic portion of the DCS validation activity.

The use of simulation provides a systematic means of testing and evaluating the control program applications for completeness and correctness with a convenient report of results. As well, additional testing required as the design progresses and evolves can easily be included in the test suite allowing comprehensive retesting to be completed with a minimum effort while ensuring that a thorough approach has been followed.

The mathematical model of the CANDU 9 plant process simulation consists of 14 reactor zones (Figure 2) with reactor regulation, four heat transport system quadrants with four lumped channels, primary pressure and inventory control system, steam and relief systems, four steam generators, feedwater system, steam turbine and generator, Group 1 electrical system, and shutdown mechanism.

The control and monitoring system model emulates the reactor regulating system (RRS), unit power regulator (UPR); steam turbine control, steam generator pressure control (SGPC), steam generator level control (SGLC), primary pressure and inventory control (P & IC), shutdown system number one and deaerator level and pressure control (DLC & DPC).

The approach taken in the CANDU 9 reactor model is a synthesis of 14 micro point-kinetic reactor models to facilitate 14 liquid zone controllers for bulk power error control, as well as zone flux tilt control. To represent first order effects, the couplings between reactor zones are represented by a simple reactivity exchange among zones, which is directly proportional to the existing zonal flux differences.

2. SIMULATOR STRUCTURE

2.1 Simulation structure

The CANDU 9 simulation software is made up of a library of generic algorithms, consisting of supporting Fortran subroutines that model the CANDU 9 systems using the CASSIM (CASsiopeia SIMulation) block-oriented modeling methodology.

Generic software algorithms were developed to correspond to physical plant components. The generic component can be a process, a logical unit, or equipment such as a valve. The block-oriented modeling technique simulates a large complex process system as a collection of these smaller components, each representing a subsystem of the large system, and each is connected to one another to represent the detailed interactions between the subsystems. More details on the simulator structure and the CASSIM simulation development system are given in Reference [3].

The block-oriented simulation technique allows flexible "connect" or "disconnect" configuration with the RRS control emulator simulation modules (Figure 3). This feature will enhance the dynamic testing of the application software. When the RRS control emulator is "connected", the simulation is standalone, not requiring any DCS input/output signals to function. When the RRS control emulator is "disconnected", reactor control DCS partition hardware will provide the control signals to the zone control valves model, as well as the reactivity devices (absorber/adjuster rods) modelled in the simulator, for reactor regulation. This DCS hardware-in-the-loop reactor control design and testing methodology allows a more in-depth dynamic analysis and evaluation of the DCS implemented reactor control programs under normal and abnormal conditions (e.g. with malfunctions).

2.2 Capabilities of the CANDU 9 Simulator Runtime Software

The runtime software for the CANDU 9 power plant simulator is capable of simulating major transients suitable for overall plant control analysis. The runtime software executes in a single Pentium PC configuration or in a network configuration in which a master Pentium PC is networked with slave PC(s) via Net DDE or TCP/IP communication protocol.

2.3 Simulation User Interface:

The simulation user interface consists of the following functions:

1) Freeze and unfreeze the simulation.

2) Save and load initial conditions.

3) A set of 29 graphical screens to facilitate monitoring, trending, operation of plant equipment status or controller setpoint parameters.

4) Historical time trends for 94 CANDU plant variables such as: neutron flux, PHT pressure, steam generator pressure, steam generator level, feedwater flow, turbine steam flow, ASDV steam flow, CSDV steam flow etc.

3. MATHEMATICAL MODELS

3.1 Modeling Scope

The simulation models in the CANDU 9 simulator include the following process modules:

1. Reactor with 14 liquid zones (Figure 4), complete with flux tilt controls and radial/axial flux mapping

2) Primary heat transport system

3) Pressure and inventory control system

4) Main steam and feedwater system

5) Electrical network

6) Shutdown cooling system

7) Reactor regulating system (RRS), complete with liquid zone controllers and demand power routine

8) Pressure and inventory control program

9) Steam generator level and pressure control

10) Unit power regulator

11) Turbine controls

12) Reactor trip

The mathematical model is capable of exercising various reactor power maneuvers and transients such as reactor power increases or reductions and reactor or turbine trips and recovery. The shutdown system model allows simulation of plant warm-up and cooldown from cold depressurized to full power states.

Plant transients in response to, at this time, seventeen postulated malfunctions can be evaluated. These include such events as failure of feedwater flow control valves, failure of heat transport system feed or bleed valves, failure of setback or stepback functions, etc.

3.2 SIMULATOR REACTOR MODELS

Traditionally, two different approaches have been proposed in the real-time simulation of space-time kinetics. One approach is based on modal kinetic formulation [4]. The other approach is based on Avery's coupled region kinetics theory [5], [6].

In a typical coupled nuclear reactor model, the reactor core is divided into a number of nodes axially and radially. The usual considerations for the choice of the nodes are the core symmetry and the accuracy required in the description of neutron flux distributions, as well as the execution time of the nodal kinetic model.

The nodes interact by coupling coefficients. These coefficients define the probability of a neutron released in a given node producing a fission neutron in another node in the next generation.

The calculation of the coupling coefficients involve the distribution computation of the resultant fast and thermal fluxes which usually require a lot of processing time, and therefore would be impractical to compute in a real-time environment.

The necessity for simpler space-time reactor models suited to the real-time implementation of DCS for reactor control has been recognized in the CANDU 9 project. The simulation need not be highly accurate, but should be sufficiently correct and robust to assist in the DCS implementation of reactor control. The approach taken in the CANDU 9 reactor model is a simpler version of Avery's formulation. It is a synthesis of 14 micro point-kinetic reactor models, taking into account only the first-order effects of the neutron flux coupling amongst zones.

The approach taken is only an approximation of the zone coupling effects and would not be suitable for detailed reactor physics analysis. However, considering only the first-order coupled reactivity effects, and the trade-off between high-fidelity, processing intensive coupling calculations versus low-fidelity, efficient real-time computation, it is found that the latter approach is adequate for the purpose of evaluating DCS reactor control implementation.

The CANDU 9 reactor model consists of the following custom-built algorithms:

- 14 simple point-kinetic models to simulate the space-time CANDU reactor kinetics and the reactivity.
- A decay heat model to simulate the decay heat produced from the fission products.
- A fuel temperature model to simulate the temperature of the fuel elements in the 4 lumped channels.

The 14 point-kinetic models are used respectively to simulate the 14 reactor zones inside the core. Each point-kinetic model will calculate the neutron power based on 6 different neutron delay groups, and the overall change in reactivity. The total delayed neutron fraction is the summation of the neutron fraction of the 6 neutron groups.

$$\beta_I = \Sigma_{i=1}^{6} \beta_i$$

β_I = total delayed neutron fraction

β_i = group i delayed neutron fraction (I=1, 2, 3, 4, 5, 6)

The change in reactivity will be a function of control devices (e.g. adjusters), concentration of xenon, voiding in channels, power changes and safety shutdown devices. The overall reactivity change is expressed as:

$$\Delta K = \Delta K_C + \Delta K_P + \Delta K_V + \Delta K_{XE} + \Delta K_{SDS} + \Delta K_{FUEL} + \Delta K_{DIFF}$$

ΔK = overall neutron reactivity change (K)

ΔK_C = neutron reactivity change due to control devices (K)

ΔK_P = overall neutron reactivity change due to power changes (K)

ΔK_V = overall neutron reactivity change due to channel voiding (K)

ΔK_{XE} = overall neutron reactivity change due to xenon poisoning (K)

ΔK_{SDS} = overall neutron reactivity change due to safety shutdown systems (K)

ΔK_{FUEL} = overall neutron reactivity change due to fuel burnup (K)

ΔK_{DIFF} = overall neutron reactivity change due to coupling between zones (K)

The reactivity change due to control devices consist of the reactivity change due to adjusters, absorbers and liquid zones.

$$\Delta K_C = \Delta K_{ADJ} + \Delta K_{CA} + \Delta K_Z$$

ΔK_C = neutron reactivity change due to control devices (K)

ΔK_{ADJ} = neutron reactivity change due to adjuster banks (K)

ΔK_{CA} = neutron reactivity change due to absorber banks (K)

ΔK_Z = neutron reactivity change due to liquid zones (K)

To represent first order effects, a coupling factor emulating the neutron diffusion from one zone to the next is added, and is calculated proportional to the difference in flux between neighboring zones. The decay heat calculation is based on 3 separate decay product groups, each with a different decay time constant. An arithmetic average flux from the 14 reactor zones is used in the decay heat calculation.

The total heat generated from the fuel as calculated using the average reactor flux is divided equally among the 4 lumped channels. Each lumped channel is assumed to have its own coolant flow, and its own fuel element. The temperature of the fuel element in each lumped channel is calculated, and the lumped fuel sheath temperature will be used in the coolant heat transfer calculation.

In the spatial reactor, the reactivity change due to control devices includes:

- 8 independent lumped adjuster banks with equal reactivity split between 14 zones.
- 2 independent lumped absorber banks with equal reactivity split between 14 zones.
- 14 liquid zones; the individual zone level will affect the respective reactivity change in its designated reactor zone volume.

4. SIMULATOR INTERFACES IN CONTROL CENTRE MOCKUP

At the time of writing this paper, the CANDU 9 simulation is nearing completion of those process models and control emulations listed earlier in Section 3.

4.1 Simulator Interface to the DCS

The CANDU 9 design employs a DCS that is independent of the PDS. Figure 5 illustrates the CANDU 9 control and monitoring design strategy in relation to the design activities of the control centre mockup. The DCS is designed to perform most plant control and data acquisition functions while most plant monitoring functions will be performed via the PDS. A limited group of manual and automatic controls will be performed outside the scope of the PDS and/or the DCS. These controls will be available on the control room panels which are equipped with control devices hardwired to the plant.

The TCP/IP request/transmit blocks running in Labview on the I/O PC request one or more signals from the CASSENG database on the simulator PC via the TCP/IP protocol. The outputs from the TCP/IP Request/Transmit blocks are fed into the Labview I/O driver block which specifies the analog or digital signal to be sent to the DCS. Signals transferred from the DCS to the simulator follow a similar process, but with reverse data flow.

This approach allows the full execution power of the simulator PC to be used and only to be limited to the execution of the simulation program without having to downgrade the real time performance by simultaneously running Labview.

Using a specific liquid zone controller as an illustration (see Figure 3), the liquid zone controller emulator currently receives the following signals as inputs:

RRS: Bulk power error

Average Reactor Flux Calculation Routine: average reactor flux

Average Zone Level Calculation Routine: average zone level

Zone Reactor Kinetic Model: local reactor zone flux and zone level

Currently, *RRS, Average Reactor Flux Calculation Routine and the Average Zone Level Calculation Routine* are emulated in the simulator. During the design integration phase, a DCS control partition will be created for the liquid zone controller. Ultimately, during the project phase, DCS control partitions will be created to accommodate all of *RRS*.

The DCS liquid zone controller will receive the same input signals via TCP/IP data link and the Data Acquisition PC. The Data Acquisition PC is equipped with ISA standard data acquisition cards, which will convert the data from binary to analog and digital signals or vice versa. The terminals of the data acquisition are wired directly to the DCS liquid zone controller input modules and to the control room mockup panel devices. The DCS liquid zone controller will operate on these inputs based on built-in control algorithms and produce a liquid zone control valve demand position output signal. This control signal is sent to the output module, which in turn goes to the data acquisition PC. The same signal is converted to a TCP/IP data protocol and is sent to the "Connect/Disconnect" transfer switch inside the simulator (Figure 3).

The "Connect/Disconnect" transfer switch receives two inputs:

1) the zone control valve demand position signal from the "Zone Controller Emulator";

2) the zone control valve demand position signal from the DCS liquid zone controller (via TCP/IP).

Depending on the setup of the transfer switch, either signal can go through the switch and provide the control input to the zone reactor kinetic model. This "Connect/Disconnect" transfer facilitates switching between the control emulator and the DCS controller during DCS implementation for reactor control for CANDU 9, thus allowing more in-depth dynamic analysis and evaluation of the DCS implemented reactor control programs under normal and abnormal conditions (e.g. with malfunctions).

4.2 Simulator Interface to the PDS

The CANDU 9 mock-up comprises several different inter-communicating software components that run on different computing platforms. The plant display system (PDS) for use in the mock-up and the CTI simulator are two such software components, with the former residing on a network of PCs running the QNX operating system, and the later on a PC running DOS and Windows 3.1. The PDS will store simulator generated plant data. The simulator data will be periodically transferred to the PDS real-time database via a software interface.

The software interface is composed of two parts. The first is responsible for the transfer of data from the simulator to the mockup PDS, and the second is responsible for the transfer of data from the PDS to the simulator. This interface to transfer data from the simulator to the PDS consists of a Windows application that retrieves data from the simulator and transmits it to the mockup PDS real-time database via a DOS to QNX send-receive library of networking routines. This software interface is self configured by reading an ASCII file from disk. This file will supply all information required to map data values from the simulation to specific points in the mockup PDS real-time database. Each addressed point in the mockup PDS real-time database that requires updating will have the update interval specified in the configuration file. Current values for each simulator point will be continually sent to the corresponding mockup PDS real-time database point at this specified rate.

The interface to transfer data from the PDS to the simulator consists of a QNX process running on the PDS real-time database machine and a windows application running on the simulator machine. The QNX process is responsible for reading the output event queue from the PDS real-time database and transmitting any new events to the windows application. These events include such things as requests to raise power or commands for starting/stopping simulator model components. The Windows application will receive this data and use it to update corresponding points in the simulation.

5. SIMULATOR APPLICATION

5.1 DCS Software Design

An overview of the software development and verification process is given in Figure 6. This figure shows how the detailed software design and code evolves from the monitoring and control requirements of the process system, how each stage of the design process is subject to review, and how the design input documentation (DID) is used as the basis for subsequent verification and validation activities. The final step in the software development process is the validation of the software. Software validation is performed by the process control system designers responsible for the preparation of the software functional requirements. Software validation will be a combination of static and dynamic testing. The objective is to ensure the control programs respond and interact in relation to plant changes and other inputs as specified in the DID under transient and steady state conditions.

In static testing, the tester applies a set of constant inputs to the DCS and observes that the DCS displays the same inputs. Any mismatch will indicate a possible incorrect allocation of input/output addresses.

Dynamic testing allows the control system designer to observe the system dynamic behavior and verify that the DCS control programs interact correctly and respond as designed under dynamic conditions. In dynamic tests, the DCS inputs are provided by the plant simulator, which in turn is controlled by the actual control program resident in the DCS using the DCS outputs. The DCS and simulator form a closed loop system to replicate the actual plant installation. Through the use of the simulator interface directly or using test scripts; the tester can manipulate manual inputs such as setpoints, gains or control modes using handswitches or pushbuttons and creating dynamic testing scenarios such as power startup, reduction, cooldown, warm-up etc.

5.2 Design of Operator Interfaces

The simulator is interfaced to the PDS to provide displays and annunciations which allow control centre mock-up evaluation to proceed. In the design integration phase, the goals are to eliminate any risks to successful project completion, to ensure that all tools are in place to conduct necessary design activities, and to provide the work processes to facilitate verification and validation activities.

Furthermore, the CANDU 9 design project has committed to following a human factors engineering (HFE) program plan through all phases of the CANDU 9 design. The simulator will play a significant role in this strategy to allow both the static and dynamic evaluation of operator indications, annunciations, and controls from the CRT based PDS. For example, displays that are developed according to system requirements and appropriate HFE design guides can be evaluated at related simulated station operating states for completeness and correctness of indications. Next, the simulation can be maneuvered to evaluate the operator automatic control input, manual actions, transition state indicators and alarms. Finally, adequacy of the display and display suite for operability during an abnormal or upset state can be tested.

6. CONCLUSIONS

Integration of simulation into the design process provides the CANDU 9 design team with a means to conduct rapid prototyping of concepts, to assess alternate layout and control strategies, to test and exercise control logic over operation state changes and to review human system interface information sets for completeness and correctness. Having a simulation platform which provides the capability to select and customize simulation modules from a library means that designers, expert in the plant design, can configure simulation models to address the design task at hand (i.e., review, assessment, testing, verification, validation etc). The resource load on the project is reduced because design experts prepare the simulation code without requiring detailed knowledge of the simulation techniques.

The simulator runtime software package facilitates a very cost-effective approach for engineering activities such as familiarization with the CANDU 9 design for training new employees or refresher training, overall plant control strategy development as well as validation of plant control software and human system interface display and control systems.

Simulator model enhancements will be on-going, to accommodate the changing needs of the engineering design teams of the CANDU 9 project, and to provide a more complete and more realistic plant state presentations. In terms of DCS software validation, it is anticipated that additional features would be incorporated to accelerate the testing process, to apply strict software configuration control, and to develop facilities to provide traceable and reproducible records of the dynamic tests performed.

7. REFERENCES

[1] Harber J.E., M.K. Kattan, M.J. MacBeth, “Distributed Control Systems for CANDU 9 Nuclear Power Plant”, 17th Annual Conference of CNS, June 1996, Fredericton N.B.

[2] MacBeth M.J. and A. Webster, “Improved Operability of the CANDU 9 Control Centre”, 11th KAIF/KNS Annual Conference, April 1996, Seoul,Korea.

[3] Kattan M.K., M.J. MacBeth and K. Lam, "CANDU 9 Nuclear Power Plant Simulator," 19th CNS Simulation Symposium, Oct 1995, Hamilton, Ontario.

[4] Luxat, J.C., "The Potential of a Generalized Modal Analysis of Postulated Loss-of-Coolant Accidents in CANDU Reactors", Nuclear Technology, Vol. 46, Mid-Dec 1979.

[5] Avery, R., "Theory of Coupled Reactors". Proc. 2nd Int. Conf. on Peaceful Uses of Atomic Energy, Geneva, September 1958, 8/1958. P.182-191

[6] Chou Q.B., De Ruyter M, and Bereznai G.T., "Development of a Low-Cost Spatial CANDU Reactor Simulated Program for Power Plant Control Studies" 1975 Summer Computer Simulation Conference, San Francisco, USA, July 21-23, 1975.

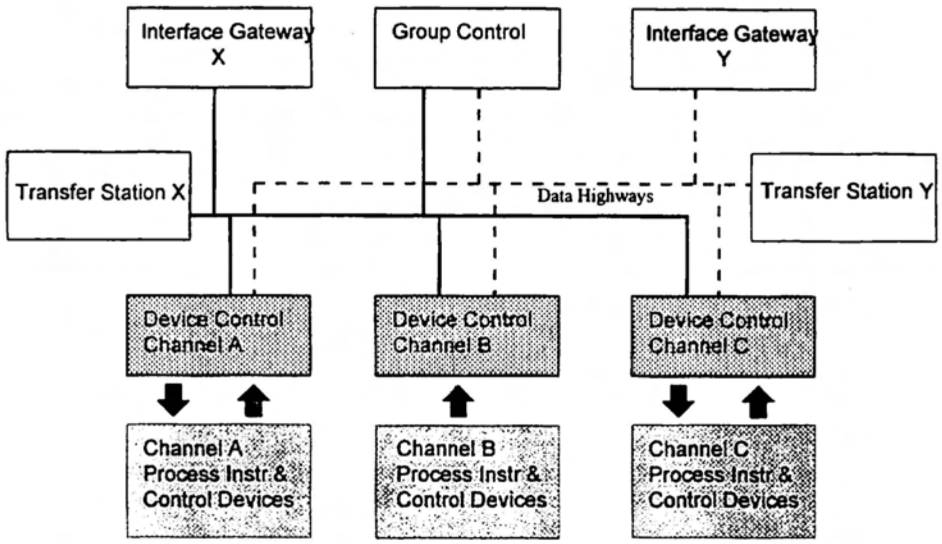


Figure 1 DCS Partition Architecture

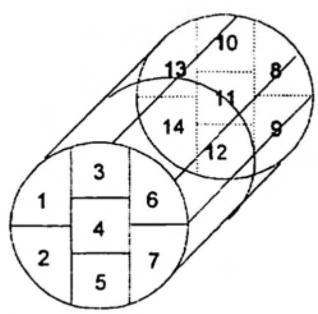


Figure 2 Configuration of Reactor Control Zones

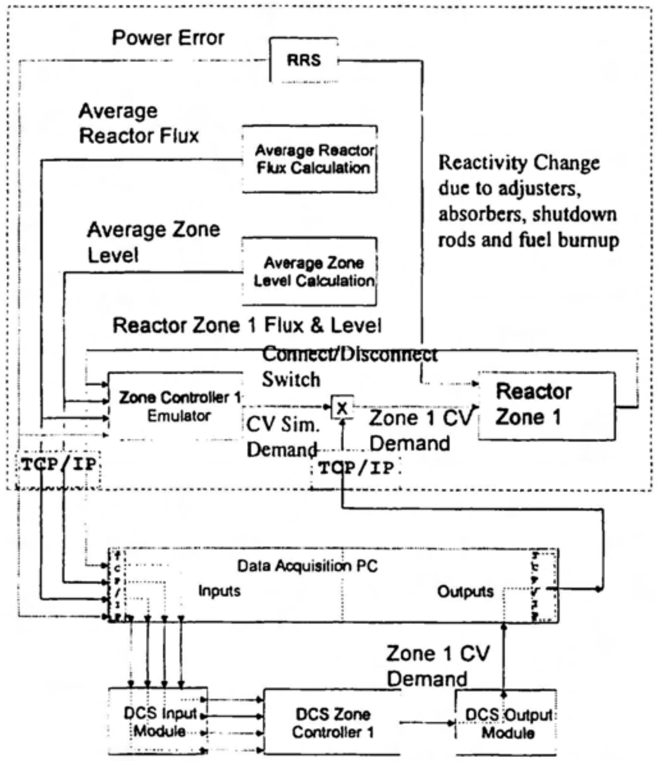
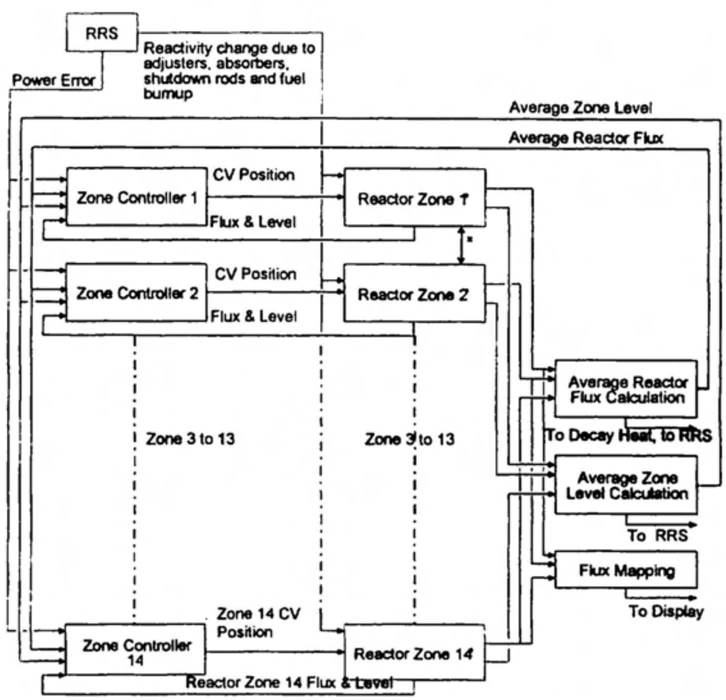


Figure 3. Liquid Zone Control System Simulation and Interface to DCS



- reactivity changes due to temperature change, xenon poisoning and voiding are modelled within each reactor zone
- coupling is modelled between each neighbouring zone

Figure 4. Reactor Model

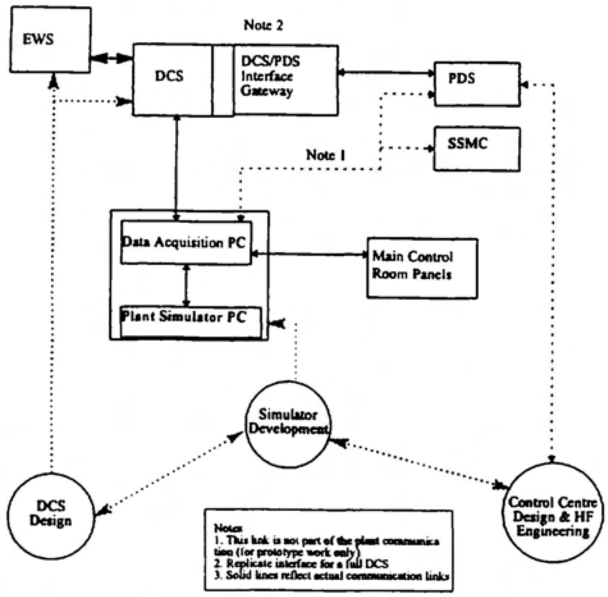


Figure 5. Schematic of the Mockup Design with the Simulator

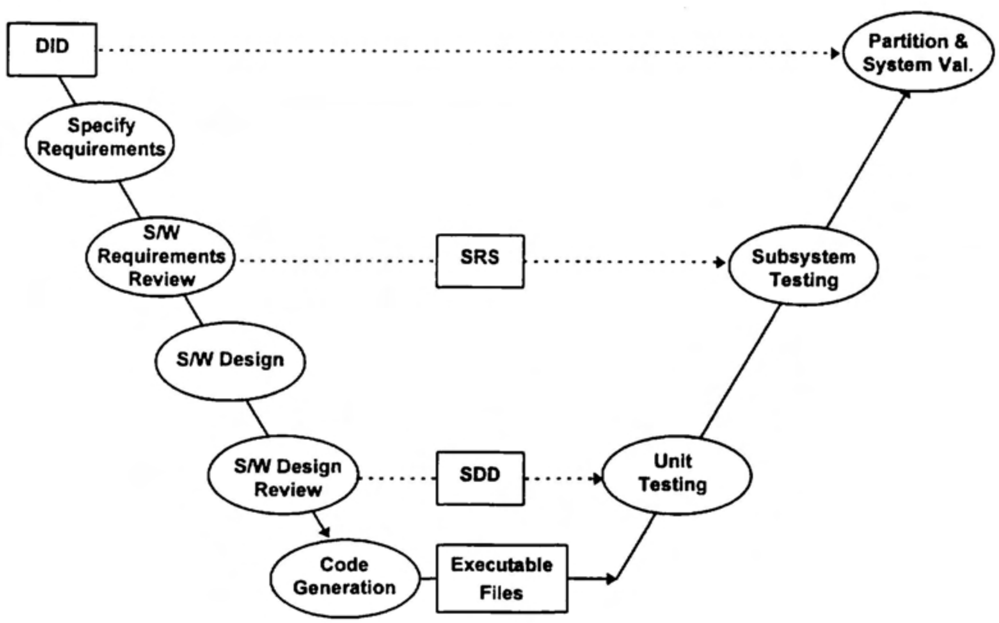


Figure 6. DCS Software Development Process