Integration of CATHENA Thermal-hydraulic Model with CANDU 6 Analytical Simulator Controller

M. Borairi[1], H. Javidnia[1], D. Wallace[1], J. Tseng[1], V. Lau[1], F. Caracas[1]
[1] Candu Energy Inc., Ontario, Canada
(Majid.Borairi@candu.com)

Abstract

This paper introduces a powerful design and analysis tool named SIMCAT that is developed to support applications to license a CANDU nuclear reactor, refurbish projects, and support the existing CANDU stations. It consists of the Canadian Algorithm for THErmohydraulic Network Analysis (CATHENA), the control logics from CANDU6 Analytical Simulator (C6SIM), and a communication protocol, Parallel Virtual Machine (PVM). This is the first time that CATHENA has been successfully coupled directly with a program written in another language. The independence of CATHENA and the C6SIM controllers allows the development of both CATHENA and C6SIM controller to proceed independently.

1. Introduction

An application to license a CANDU reactor must include data or information to illustrate that the design provides adequate safety margins in the operation and maintenance of the facility. Much of this data is generated from computer codes developed to simulate the operation and performance of the nuclear reactor under various operating conditions.

Historically, CANDU design benefited from the evolution of the numerical computational tools and modeling of various complex systems. This section provides a brief description of a few important codes used in the design of CANDU 6 and Enhanced CANDU 6 (EC6) reactors.

a. CATHENA is a one-dimensional, two-fluid thermal-hydraulic simulation code designed to analyze two-phase flow and heat transfer in piping networks [1]. Its primary use has been the analysis and simulation of the consequences of postulated upset scenarios in CANDU and research reactors. CATHENA has evolved with the objective of providing a high degree of flexibility in modeling thermal-hydraulic systems. CATHENA is a qualified safety code and meets the quality assurance requirements of CSA-N286.7-99 ([2]) for a Tier level I Analytical, Scientific, and Design (ASD) code.

   CATHENA offers limited control system modelling capabilities by:

   - Including control system models in the CATHENA input file;
   - Linking CATHENA to an external executable code that provides the major portion of the plant control logic

However, there are some limitations in implementing the control logics using these methodologies; maintainability, flexibility, and ability to change modules independently.

b. Another code which was developed to study the transient performance of the nuclear heat transport system is the Simulation of Primary Heat Transport System (SOPHT). SOPHT was designed to differ from the simulation programs customarily used by the control system designers by emphasizing the transient properties of the thermal-hydraulic system as well as the response of the associated control system. The primary interest of the program is to predict the system transient during a relatively large, low frequency disturbance (upset conditions) as well as during normal controlled manoeuvring operation [3]. SOPHT has been retired from the Candu Energy/AECL code suite and does not comply with CSA N286.7 standard required for safety analysis and design.

c. C6SIM is developed by Candu Energy Inc., with the objective of providing accuracy and flexibility in modeling thermal-hydraulic, control/ instrumentation, and reactor core systems to predict the transient behaviour of the CANDU 6 reactor. The C6SIM thermal and hydraulics models are similar to SOPHT and its control is based on a typical CANDU 6 logics running on station computers. Unlike CATHENA and SOPHT which use a point kinetics model to represent the dynamics of the core, C6SIM uses a spatial model that calculates the flux (or power) at the center of each one of the 14 liquid zones of the CANDU reactor. C6SIM is developed on a modular basis so that subsystems such as the control logics could be easily separated and used independently.

   The C6SIM code can usually run faster than real-time and it is equipped with a powerful graphic user interface (GUI) developed in LabView environment. C6SIM is a qualified code and meets the quality assurance requirements of CSA-N286.7-99 for Tier level II ASD code.

d. SIMCAT, developed by Candu Energy Inc., with the intention of overcoming the shortcomings of the above codes by employing CATHENA and the C6SIM controller coupled together using a verified and reliable communication protocol, PVM.

   CATHENA and C6SIM are written in Fortran and MATLAB/Simulink platforms, respectively. This is the first time that CATHENA has been successfully coupled directly with a program written in another language. It is also the first time CATHENA has been coupled directly to a program that is not a true “child” process. The independence of CATHENA (the “parent” process) and the C6SIM controllers (the “child” process) allows the development of both CATHENA and C6SIM controller to proceed independently. The only requirement is that there is consistency in the communication interface exchanging data between the two processes.

   An efficient link between the two codes has been established to allow control programs to take advantage of a large existing library of CATHENA functions. The coupling of CATHENA with the C6SIM controllers uses a CATHENA “REMOTE PROCESS” option which provides the ability to couple CATHENA with external programs to perform integrated calculations. This is accomplished by allowing a user to initiate or spawn separate processes from a CATHENA simulation and to associate an executable, in this case such as the C6SIM controllers, with each remote process.

The SIMCAT is used to simulate the dynamic behaviour of the CANDU 6 and EC6 process and control systems for normal operation and some unplanned transient events such as loss of Pressure and Inventory Control (P&IC) system, loss of Class IV power, small Loss of Coolant Accident (LOCA) trip coverage analysis, steam generator tube rupture, and feedwater and steam line system failures.

This is accomplished by performing integrated CANDU6/EC6 thermal-hydraulic/process and control system transients of:

- The primary heat transport system (PHTS),
- The pressure and inventory control system (P&IC),
- The steam generator secondary side,

The range of applicability of the coupled CATHENA/C6SIM code is from zero power hot to 100% full power[1].

Validation of the thermal-hydraulic, fuel, and fuel channel thermal mechanical phenomena associated with these transients is provided through CATHENA validation documented in [4]. Twenty six validation exercises are identified covering thirty different phenomena to address the events listed above.

2. SIMCAT Components

2.1 Thermal-hydraulic Model

The CATHENA idealization of the PHT system is based on Wolsong/Qinshan models The idealization is a 4-channel, 2-loop core pass of a CANDU 6 reactor. The model was used in the development of a generic CANDU 6 controller.

Four Steam Generators (SGs) are modelled in the CATHENA idealization. The CATHENA model of the P&IC system is based on a SOPHT idealization that was used in the development of an earlier control model prototype. The thermal-hydraulic model includes three tank models (pressurizer, degasser-condenser, and D2O storage tank), feed, bleed and purification lines, degasser-condenser and spray lines, pressurizer lines and associated valves.

2.2 Control System Models

The models for the P&IC, and the SG level and pressure control systems are based on a typical CANDU 6 reactor program specifications.

The Simulink toolbox allows users to create blocks of systems/subsystems containing process control functions, such as integrators, transfer functions, etc. In order for the Simulink to run independently without MATLAB the system blocks are translated into C programming language,

[1] The full range of applicability is addressed through both the response of the C6SIM controllers and CATHENA input models that activate and provide inputs to the controllers.

and then compiled to a Linux executable using Simulink Coder (formerly known as Real-Time Workshop [5]).

2.3 CATHENA/C6SIM Controller Interface

The communication between the SIMCAT components is configured based on client and server communication methodology; CATHENA is the client and the C6SIM controller is the server.

An interface is required so that CATHENA and the C6SIM controllers can exchange data. The C6SIM controllers are translated into C code and use this interface to receive/send data from/to CATHENA. CATHENA has the PVM interface built into it and it is the normal mode of exchanging data with an external executable.

PVM is the suite of communication libraries used to interface with CATHENA variables through the CATHENA input file. It enables a collection of heterogeneous computer systems to be viewed as a single parallel virtual machine [6].

Any external application communicating with CATHENA must also support the PVM communication protocol. For each remote process, CATHENA sends its data and waits as the remote process receives this data, performs its calculations, and returns the data to CATHENA. A sampling time of 2 seconds is used with the C6SIM controllers. CATHENA continues its calculations based on the new input from the remote process updated every 2 seconds. This sequence of data exchange between CATHENA and the C6SIM controllers continues until CATHENA terminates the simulation.

The code generated by Simulink Coder for the controller normally relies upon user-input through the Simulink Interface. The user performs a run-time compilation before a simulation. Extraction and modification of the block of code created by the user through the Simulink interface is allowed but the source of the main C portion of the program is not.

Figure 1 shows the configuration of the CATHENA and C6SIM interfaces.

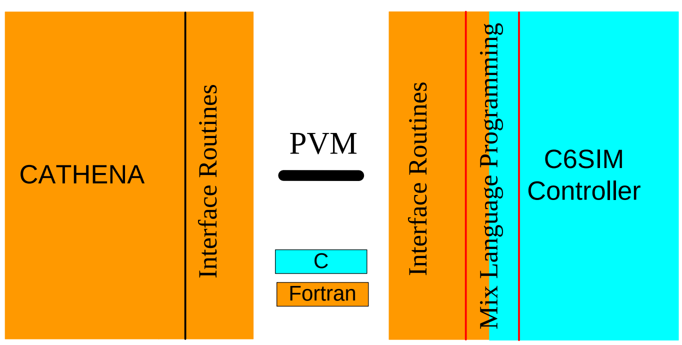

Figure 1: Configuration of CATHENA /C6SIM Interfaces

A CATHENA/C6SIM controller interface is embedded in a function written in C and driven by the code compiled by Simulink Coder. A modified makefile generates an executable of the controller that includes both the main program and the modified version of the C code. The CATHENA PVM remote process option will initiate a process created by the executable generated by the make-file. Data will be exchanged between CATHENA and the C6SIM model by communicating through the interface embedded in the C code.

The interface embedded in the C code and is *driven by* the remote process (the control program) initiated through PVM.

3. Program Descriptions

The PVM interface for SIMCAT is comprised of three Fortran subroutines that are invoked from the C code generated by the Simulink Coder. These subroutines are embedded inside the C code and this part of the code development is the mixed language programming that is highlighted in two colours in Figure 1. The Fortran subroutines do the following tasks:

1. Routine 1 is called only once in the beginning of a simulation, establishing the PVM connection, opening input/restart files and performing variable declarations and initializations as required.
2. Routine 2 reads/receives data from the parent process.
3. Routine 3 sends data to the parent process (CATHENA), decides if the simulation is to terminate and if so, writes data to restart files as required, closes files, and exits PVM.

3.1 Creating the C6SIM Controller Executable

An executable of the C6SIM controller is generated using the following steps:

1. Compile the Simulink model to create a “C” file.
2. Modify the C file to enable the communication protocol via PVM. This includes entering calls to the interface routines, and transferring data between C6SIM and CATHENA/PVM input/output variable names.
3. The “Makefile” generated by Simulink Coder for the controller is modified to include the interface routines, modules, and data structures and also to force the linker to build a static executable.

Figure 2 shows the steps to be followed for creating a C6SIM controller executable file.

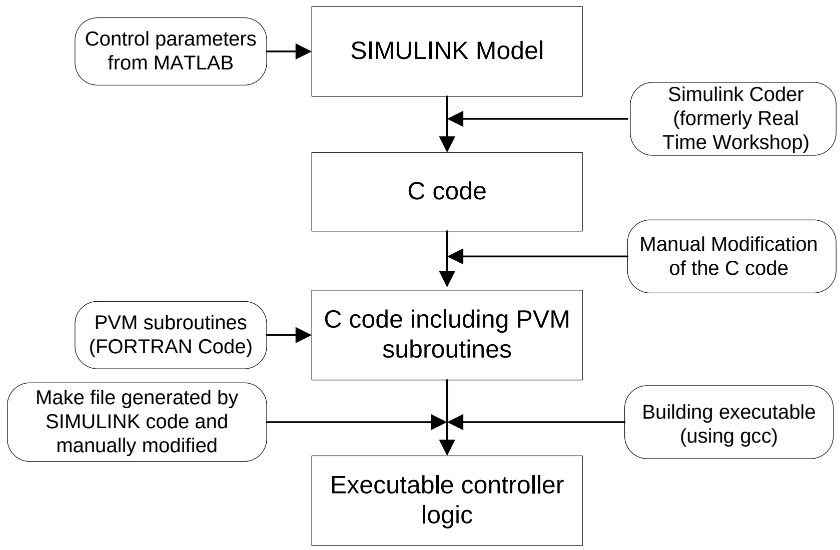

Figure 2: Steps for Creating C6SIM Controller Executable

4. Case Study

SIMCAT is used to assess the performance of the steam generator level control in a typical CANDU 6 reactor in response to a reactor power increase from 40%FP to 100%FP and a power decrease from 100% to 40%FP.

It is evident from Figures 3 that steam generator level and the control valves are well behaved and their performance is in agreement with the operational experience available for similar power transients.

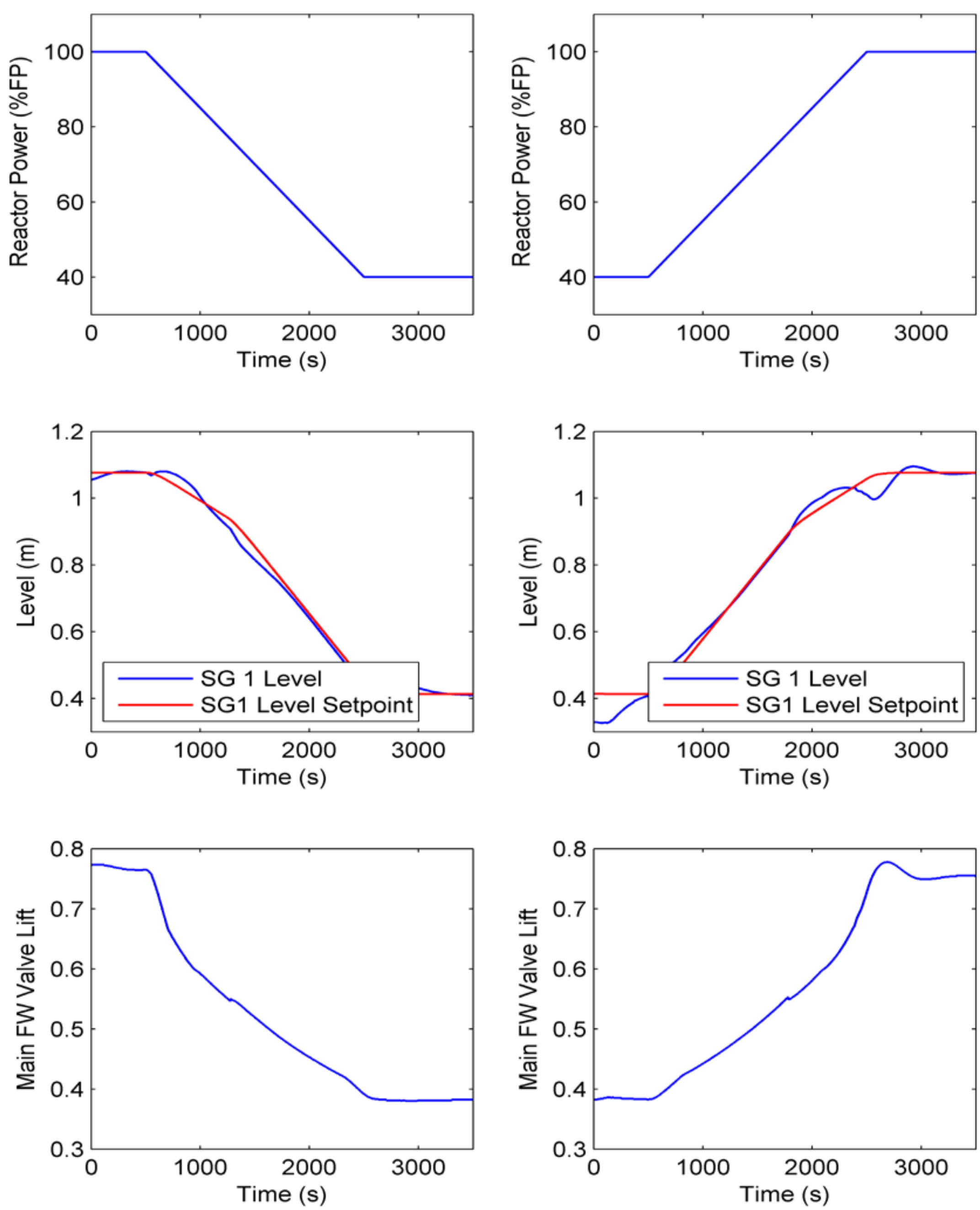

Figure 3: Responses of the Steam Generator Level and Feedwater Control Valves to Power Manoeuvring

5. Conclusion

The enhancement of CATHENA with the C6SIM controller through a reliable communication protocol has resulted in a unique and powerful analysis tool. SIMCAT is expected to improve the quality and accuracy of the safety, process, and control system design/analysis required for various projects. it allows for the first time to analyse important events such as loss of control system and generate the transient trajectories required for reliability testing of the CANDU shutdown system computers. Moreover, for the first time important events such as loss of control can be analysed and trajectories could be easily generated and be used in reliability testing of the CANDU shutdown system computers.

The independency of CATHENA and the controller allows efficiently design and examine the effectiveness of different control strategies for any CATHENA model.

The use of SIMCAT can now be contemplated to even build a desktop simulator purely in software.

6. REFERENCES

[1] Hanna, B.N., 1998, CATHENA: A thermalhydraulic code for CANDU analysis, Nuclear Engineering and Design (180) 113-131.

[2] Quality Assurance of Analytical, Scientific and Design Computer Programs for Nuclear Power Plants, CSA N286.7-99.

[3] A Themal-hydraulic System Simulation Model for the Reactor, Boiler, and Heat Transport System (SOPHT), Ontario Hydro CNS-00037-002, September 1997.

[4] Hanna, B., "CATHENA MOD-3.5d Validation Plan" AECL Report COG-01-204, March 2003.

[5] Getting Started with Simulink Coder, the MathWorks Inc., and September 2011.

[6] Al Geist, et. al.,"PVM: A User's Guide and Tutorial for Networked Parallel Computing" , MIT Press, 1994.

Acknowledgement

The authors would like to thank Thomas Beuthe and Alexsandar Vasic, and Bruce Hanna of AECL. This long work could not be successfully accomplished without their brilliant knowledge and assistance.