Performance Test of Core Protection and Monitoring Algorithm with DLL for SMART Simulator Implementation

BonSeung Koo, DaeHyun Hwang and KeungKoo Kim
Korea Atomic Energy Research Institute, 1045, Daedeokdaero, Yuseong-gu, Daejeon, 305-353, Korea
bskoo@kaeri.re.kr

1. Introduction

The Korea Atomic Energy Research Institute (KAERI) has developed a system-integrated modular advanced reactor, SMART, with 330MW of thermal power. SCOPS and SCOMS were developed as an advanced real-time core protection and monitoring system of SMART[1,2].

A multi-purpose best-estimate simulator for SMART is being established, which is intended to be used as a tool to evaluate the impacts of design changes on the safety performance, and to improve and/or optimize the operating procedure of SMART. In keeping with these intentions, a real-time model of the digital core protection and monitoring systems was developed and the real-time performance of the models was verified for various simulation scenarios[3].

In this paper, a performance test of the core protection and monitoring algorithm with a DLL file for the SMART simulator implementation was performed. A DLL file of the simulator application code was made and several real-time evaluation tests were conducted for the steady-state and transient conditions with simulated system variables.

2. Methods and Results

2.1 Real-time Model Development

The SCOPS_SSIM and SCOMS_SSIM code are being developed as a protection and monitoring program for SMART simulator application, respectively. These codes will be loaded as a Dynamic Link Library (DLL) file in a simulator platform. Each code receives the computational status and system variables from the simulator, performs core protection or monitoring algorithm, and then returns calculated variables to the simulator platform. The codes were developed based on SCOPS and SCOMS.

Software design bases and requirements were set up for the simulator application as well as the software performance requirements. In addition, the input/output variables, how to execute the software, and how to connect the software with the simulator platform were defined. Fig. 1 shows the SCOPS_SSIM software connection method with the SMART simulator. The common (or shared) memory of the simulator is used as a data communication buffer. SCOPS_SSIM obtains system variables and then calculates pre-defined protection variables within the time requirement. The system code, neutronics code, thermo-hydraulics code, and all other related codes will be linked with the simulator platform. These codes perform their unique algorithm and provide calculated information to the common memory of the simulator. This inter-communication has to meet the performance requirements, i.e., the time interval, Backtrack/Snapshot option, and so on. SCOMS_SSIM is, also, connected with the simulator platform as like as SCOPS_SSIM.

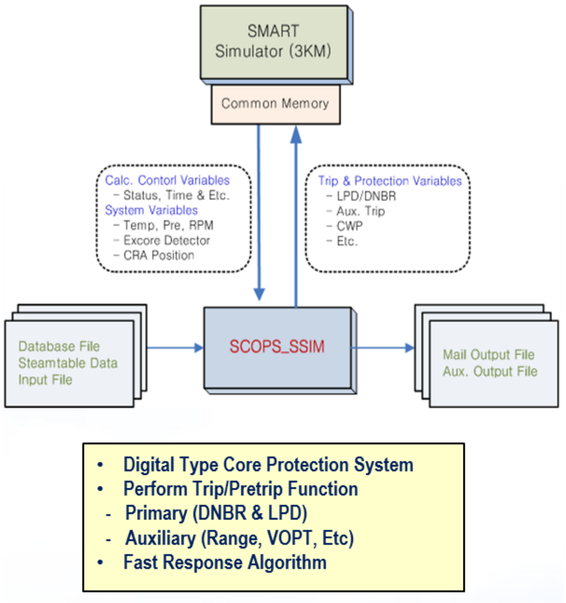
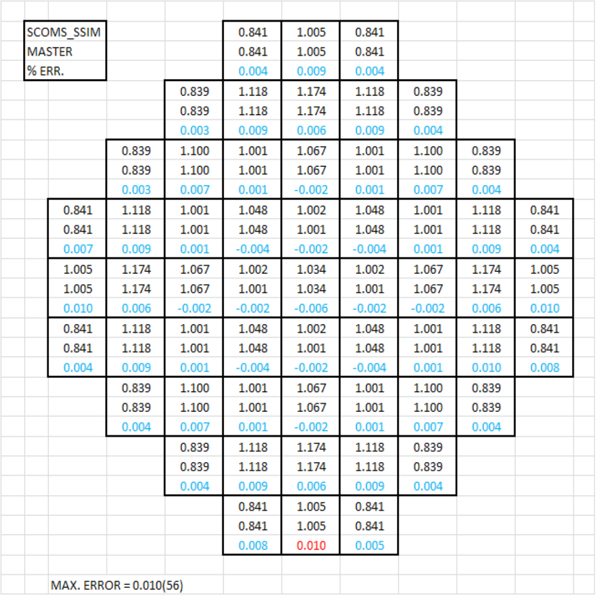


Fig. 1. SCOPS_SSIM code connection with simulator.

During the model development, protection and monitoring algorithms were improved and the real-time capability of SCOP/SCOMS_SSIM was enhanced to meet the performance requirement of the simulator. Fig. 2 shows the synthesized power distribution and its error of SCOMS_SSIM. This is a typical output of the SMART initial core at a BOC condition. As seen, the radial power distribution error (axially integrated 2-D power) shows a negligible value compared with the reference MASTER neutronics results.



SCOMS_SSIM MASTER % ERR.			0.841 0.841 0.004	1.005 1.005 0.009	0.841 0.841 0.004			
		0.839 0.839 0.003	1.118 1.118 0.009	1.174 1.174 0.006	1.118 1.118 0.009	0.839 0.839 0.004		
	0.839 0.839 0.003	1.100 1.100 0.007	1.001 1.001 0.001	1.067 1.067 -0.002	1.001 1.001 0.001	1.100 1.100 0.007	0.839 0.839 0.004	
0.841 0.841 0.007	1.118 1.118 0.009	1.001 1.001 0.001	1.048 1.048 -0.004	1.002 1.001 -0.002	1.048 1.048 -0.004	1.001 1.001 0.001	1.118 1.118 0.009	0.841 0.841 0.004
1.005 1.005 0.010	1.174 1.174 0.006	1.067 1.067 -0.002	1.002 1.001 -0.002	1.034 1.034 -0.006	1.002 1.001 -0.002	1.067 1.067 -0.002	1.174 1.174 0.006	1.005 1.005 0.010
0.841 0.841 0.004	1.118 1.118 0.009	1.001 1.001 0.001	1.048 1.048 -0.004	1.002 1.001 -0.002	1.048 1.048 -0.004	1.001 1.001 0.001	1.118 1.118 0.010	0.841 0.841 0.008
	0.839 0.839 0.004	1.100 1.100 0.007	1.001 1.001 0.001	1.067 1.067 -0.002	1.001 1.001 0.001	1.100 1.100 0.007	0.839 0.839 0.004	
		0.839 0.839 0.004	1.118 1.118 0.009	1.174 1.174 0.006	1.118 1.118 0.009	0.839 0.839 0.004		
			0.841 0.841 0.008	1.005 1.005 0.010	0.841 0.841 0.005			

MAX. ERROR = 0.010(56)

Fig. 2. Axially integrated power distribution error

2.2 DLL Test and Results

Real-time performance tests were conducted for the steady-state and transient conditions with a DLL file and simulated system variables. Test cases were selected to verify the input/output processing performance and justness of the real-time protection and monitoring algorithm. The following are the tested simulation cases:

- Trip (or Pretrip) variables generation
- LCO variables generation
- Power (1st, 2nd and Turbine) calculation
- DNBR/LPD calculation
- Power distribution synthesis
- POL (Power Operating Limit) calculation
- Real-time performance of most test cases

A DLL file of SCOPS/SCOMS_SSIM was made and mentioned simulation cases were tested. A processing program was generated to call a DLL file, to read system variables from an external file, and to process the calculational results. The program plays a role of simulator platform in this DLL test. Fig. 3 shows the procedure for the DLL test of SCOPS/SCOMS_SSIM software with the simulator platform. 3KEYMASTER will be used as a simulator platform. Each code determines the calculation mode by the received NMODE variable of the simulator platform. NMODE(1) is normal calculation mode and the others are used for the Snap/Backtrack function. NMODE(2 & 3) only reads a restart file for Snap/Backtrack calculation. NMODE(4) writes a restart file with a previously calculated output file. NMODE(5) is, also, normal calculation mode as well as NMODE(1), but generates a restart file right after finishing a calculation.

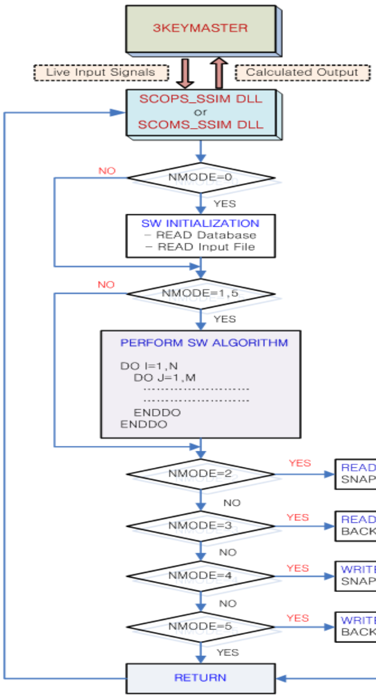


Fig. 3. Procedure for SCOPS/SCOMS_SSIM calculation with simulator platform.

Fig. 4 shows one of test results of the SCOPS_SSIM DLL test. As a flow-related event, the core flow sharply decreases and returns to the initial value as time goes on. The core flow rate changes proportionally to the rotational pump speed, and the DNBR, LPD and core power follow the change in flow rate. Fig. 5 is one the test results of the SCOMS_SSIM simulation. Two regulating groups (Reg.2 & Reg.3) are inserted and then withdrawn with a constant speed. In addition, group deviation occurs at the Reg.2 group with 50cm magnetite for 0.0 to 11.0 seconds.

Each DLL test results were compared with reference results calculated by a normal program, i.e., without using a DLL file. As shown in figures, DLL calculation results match the reference results well, and some features caused by the algorithm change are properly reflected in the DLL test results. SCOPS_SSIM DLL algorithm is performed with 0.1 second time step compared with pre-defined execution interval. So, this effect is shown in DNBR value of Fig. 4. In addition, it can be seen in Fig. 5 that DLL calculation is started from 12 seconds as a continuous simulation, i.e., running every second. In contrast, a reference calculation is started from 20 seconds because DNBR calculation module has a 10 second execution period.

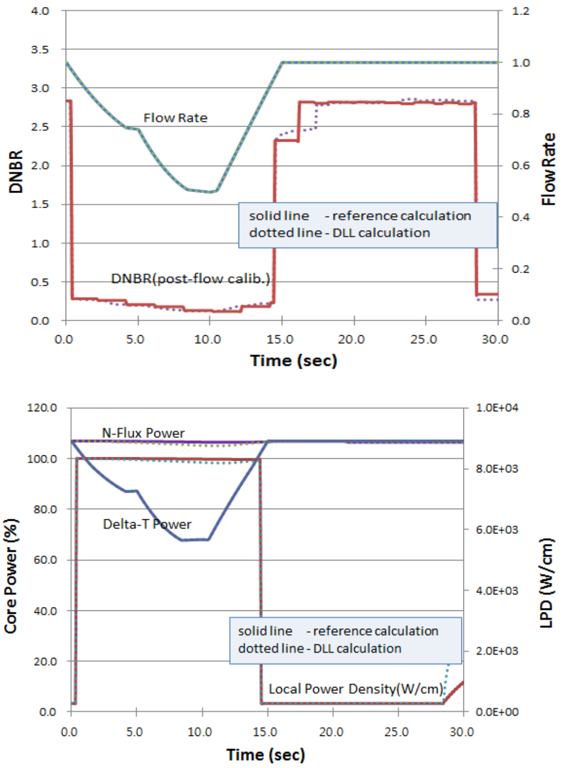


Fig. 4. SCOPS_SSIM DLL test results vs. reference results.

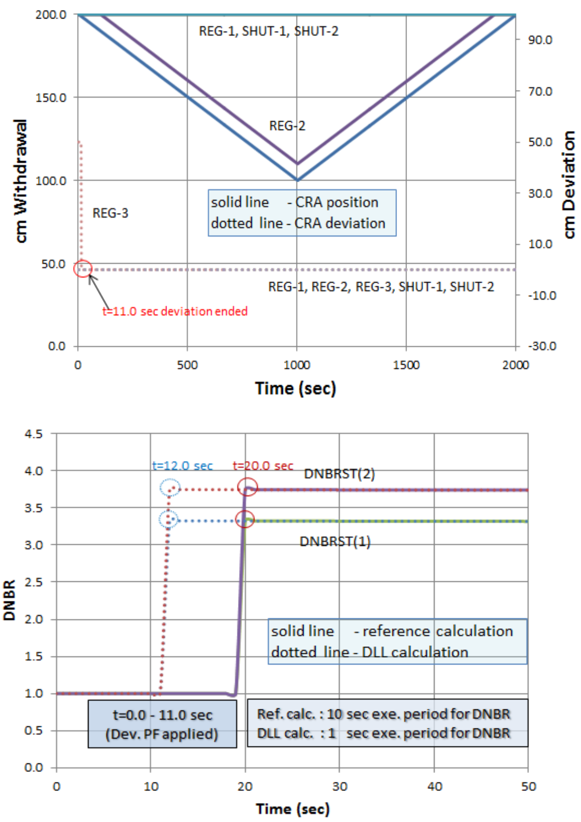
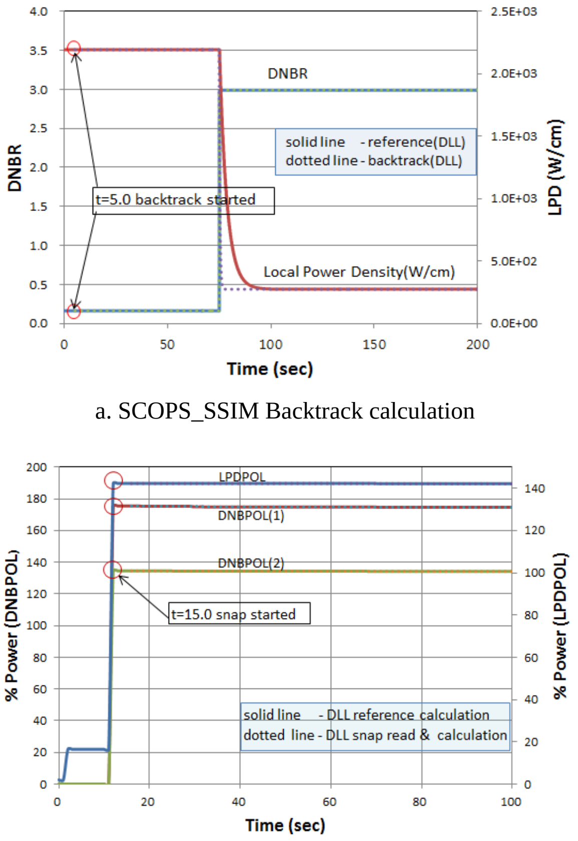


Fig. 5. SCOMS_SSIM DLL test results vs. reference results.

The capability of the Snap/Backtrack feature of the SCOPS/SCOMS_SSIM code was tested and its results were compared with the reference DLL results. As depicted in Fig. 3, the simulation software reads (or writes) a restart file for the Snap(or Backtrack) calculation. Therefore, the restart file must be ready for the Snap/Backtrack calculation. The restart file is composed of the initial condition data including live signals, as well as the final and temporary calculation variables. As shown in Fig. 6, the Snap/Backtrack calculation exactly reproduces the reference result from the designated time point. The SCOPS_SSIM Backtrack and SCOMS_SSIM Snap calculation were started at 5.0 and 15.0 seconds, respectively.



a. SCOPS_SSIM Backtrack calculation

b. SCOMS_SSIM Snap calculation

Fig. 6. DLL Snap/Backtrack option test vs. reference results.

3. Conclusions

A performance test of the core protection and monitoring algorithms for the SMART simulator was performed. A DLL file of the simulator version code was made and several real-time evaluation tests were conducted for various scenarios with a DLL file and simulated system variables.

The results of all test cases showed good agreement with the reference results and some features caused by algorithm change were properly reflected to the DLL results. Therefore, it was concluded that the SCOPS_SSIM and SCOMS_SSIM algorithms and calculational capabilities are appropriate for the core protection and monitoring program in the SMART simulator. Future works on the integrated tests with the simulator platform are under progress. The variety of the operating scenarios will be considered under an integrated platform situation, i.e., all of the software will be coupled with the simulator platform.

ACKNOWLEDGEMENTS

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2012M2A8A4026261).

REFERENCES

[1] B.S. KOO, et al., “SMART Core Protection System Design Report,” KAERI/TR-4590, KAERI (2012).
[2] B.S. KOO, et al., “SMART Core Monitoring System Design Report,” KAERI/TR-4589, KAERI (2012).
[3] B.S. Koo, et al., “Performance Evaluation of SMART Core Protection and Monitoring System for Simulator Coupling,” Transaction of ANS Winter Meeting, Vol.(109), Washington DC, Nov. 10-15 (2013) (CD-ROM).