QUALIFICATION OF A COMPUTER PROGRAM TO ANALYZE SHUTDOWN SYSTEM FLUX DETECTOR RESPONSE IN POINT LEPREAU GENERATING STATION

Q. Alexander[1], V. N. P. Anghel[1], D. Comeau[2], M. Ducic[1], R. Hutanu[1], G. Jonkmans[1], J. McKay[1], B. Sur[1] and D. Taylor[2]

[1] Atomic Energy of Canada Limited, Chalk River, Ontario, Canada

[2] New Brunswick Power Nuclear, Point Lepreau, New Brunswick, Canada

Abstract

A study of more than ten years of reactor trip data, following the installation of Platinum-clad In-Core Flux Detectors at the Point Lepreau Generating Station (PLGS), revealed "anomalous" detector response times. In response to the finding, a fully automated and qualified computer program was developed to analyze detector response. The program analyzes delayed detector response to neutron flux as it deviates from design expectation. The following paper discusses the issues encountered and software development processes followed to ensure adherence to Canadian Standards Association (CSA) N286.7-99.

1. Introduction

The current In-Core Flux Detectors (ICFDs) at the Point Lepreau Generating Station (PLGS) are Straight Individually Replaceable (SIR) Platinum-clad Inconel self-powered detectors, housed in Helium Encapsulated SIR (HESIR) assemblies. The signal dynamics of the SIR detectors were measured via a manual reactor trip test in December 1992. The measured detector signals were then compared to design expectation from a full simulation of the reactor trip via a Reactor Fuelling Simulation Program (RFSP) [1]. At the time, the detector responses from the manual reactor test trip were confirmed to meet design specifications.

Years later, detector response data was collected from trips in 1995, 1996 and 1997 and analyzed without the benefit of a full trip simulation. Subsequent analyses (for reactor trips between 1997 and 2007) confirmed that some individual detector responses, as well as the Shutdown System (SDS) average response, deviated from design expectation [2] [3]. SDS1 "anomalous" detector response times were generally faster than the design expectation, hence, assessed not to be a problem in most cases. However, some SDS2 detectors were found to be slower than expected.

In response to the finding, AECL and PLGS have developed a fully automated and qualified computer program for detector response analysis [4]. The development of qualified software (VS program) provides a means to analyze the dynamic response characteristics for SDS1 and SDS2 flux detectors for CANDU ® reactors.

The VS program performs several tasks, such as:

- Extracting useful input from archived station data and displaying the information.
- Performing mathematical computations on "acceptable" data.
- Presenting the computed results to the user.

For input data, the program uses both fast (20 ms sampling interval) shutdown data (from a High-Speed Data Logging or HSDL System) and slower data from the Safety System Monitoring System (SSMS); this enables the automated response characterization of SDS1 and SDS2 detectors at the PLGS. The program performs a generalized least-square fit of the expected response to detector response data collected during shutdown. For platinum-clad Inconel detectors, a significant portion of the signals lag neutron flux, thus, the lagging part of the signal is assumed to be a linear combination of delayed fractions. Lifetimes of the delayed fractions are given in the safety analysis of the reactor, while their amplitudes are computed by fitting trip data to the theoretical response. The VS program computes the delayed response fraction amplitudes, enabling estimation of the detector dynamic compensation error [3] [4] [5].

The extent of qualification necessary for the VS program was dependent on the scope of the program and its safety implications for plant workers, public and the environment. The VS program will be used to monitor the performance of safety system ICFDs. More specifically, output of the program will be used to estimate the dynamic compensation error of ICFDs in the analysis of the Regional Overpower Protection (ROP) system. Consequent to the intended scope, the VS program, accompanied by the theoretical basis of the program, has to be compliant to the CSA N286.7-99 Standard.

This paper describes the VS program structure, discusses the development processes followed and discusses the results of the qualification process.

2. Program structure

2.1 Programming languages and portability requirements

The VS computer program is subdivided into two parts: the "front-end", where interaction with the user takes place, and the "back-end", where detailed numerical calculations are performed. Each end is coded using a different programming language so that functions are performed optimally:

a) The "front-end" program launches program runs, manipulates data files and interacts with the user. For this part, a standardized programming language was required to ensure compatibility of the VS program to various hardware and software environments. The "front-end" is written in Visual Basic.NET (VB.NET) 2008, a language that provides good interaction with the user, as required by the client.
b) The "back-end" program performs detailed numerical calculations. A standard programming language was required to ensure strong numerical processing capabilities. As VB.NET does not offer the needed capability, FORTRAN 90 is used for the "back-end" application.
c) To allow portability during the life of the software, as a rule, sufficient compatibility is needed to make updating the program a manageable task. Portability is achieved through the selection of well supported application programs and adherence to standardized programming practice.
d) The "back-end" communicates with the "front-end" through ASCII input-output files, and each "end" can be updated independently.

Figure 1 illustrates the structure of the VS computer program [2].

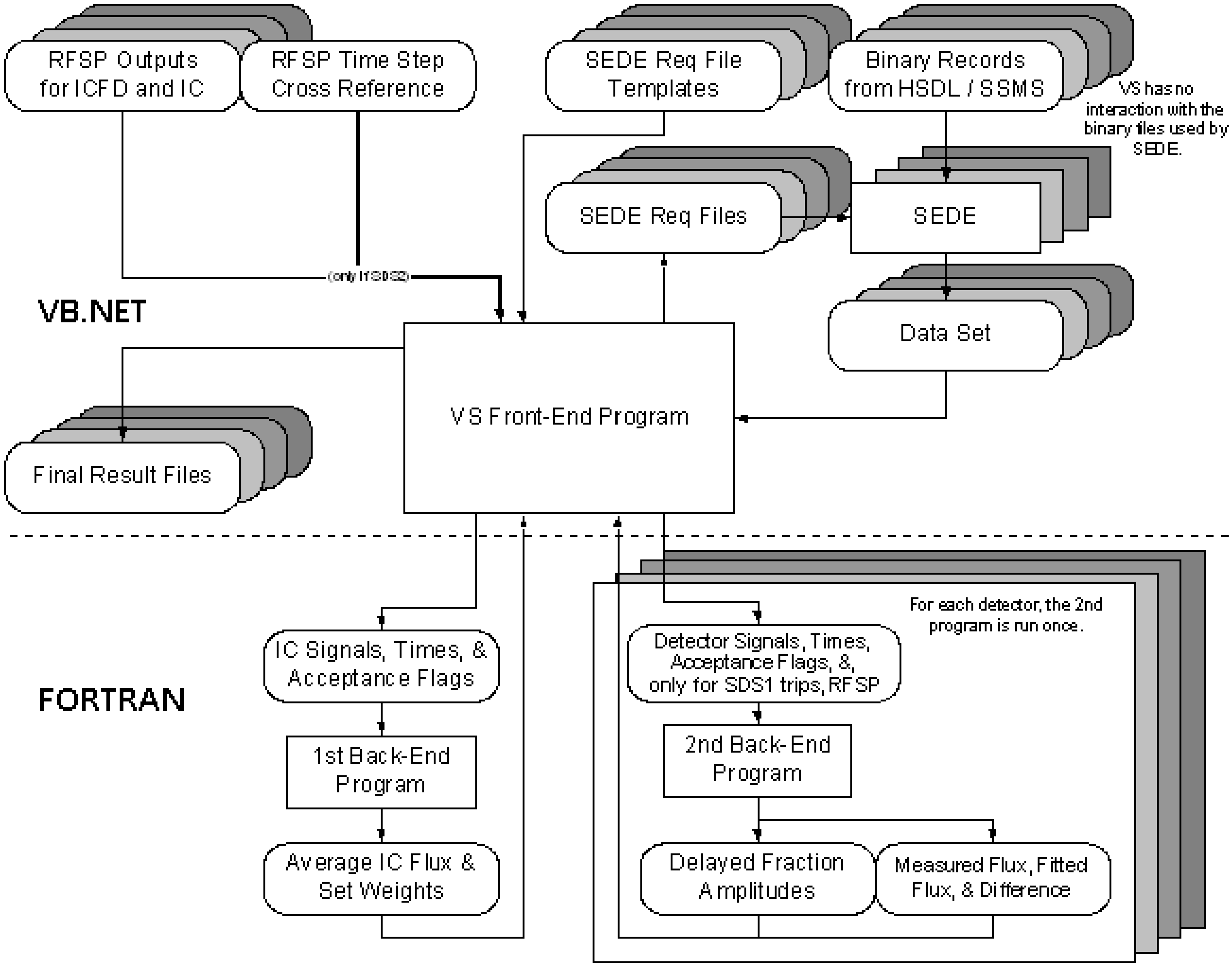

Figure 1   VS Computer program structure

2.2 Input Data

Detector and ion chamber output signal data was extracted from the data recorded by a High Speed Data Logger (HSDL) system. For the purpose of analysis, useful data had to be gathered to study the behaviour of each reactor trip. Consequently, the data from each reactor trip was stored in five separate data sets. Each data set contained detector and ion chamber signals that were represented by a specific sampling time and interval; the time interval was centred symmetrically around the trip, as shown in Table 1.

In some cases, the HSDL system does not record data over long time durations, i.e., the data corresponding to data sets 3, 4, and 5 in Table 1. In such cases, data from the Safety System Monitoring System (SSMS) are used instead. The SSMS samples data every 6 seconds.

Table 1
Data Set Sampling Time and Interval

Dataset	Sampling Time (s)	Total Duration
1	0.02	2 min
2	1	10 min
3	6	1 hour
4	60	10 hours
5	600	200 hours

The above table provides an example of the sampling time and time interval for each data set that was used towards the analysis.

3. General testing strategy

Based on the scope and input to plant safety, the design, development, maintenance and modification of the VS program must adhere to the CSA N286.7-99 standard. To ensure compliance, the VS program had to undergo verification and validation. Verification is defined as a process to determine whether or not the products of a given phase of the computer program development lifecycle fulfil the requirements from the previous lifecycle phase. Independent reviews, testing and walkthroughs are some of the means for accomplishing verification. On the other hand, validation is a process in which results of a computer program calculation are compared with measurements or known analytical or numerical solutions. With validation, the accuracy or uncertainty of an application-specific computer program calculation can be determined. Therefore, the purpose of validation is to demonstrate that a product fulfils its intended use when placed in its intended environment.

For the VS program, all verification and validation activities were performed by qualified persons that maintained independence of each phase of the computer program development lifecycle. Note: Validation was performed for the VS program as a whole.

3.1 Verification

The verification process entailed a systematic review of documents as required by the CSA N286.7-99 standard. Documentation verification included review of the 1) theoretical and mathematical foundation of the VS program to ensure that it is appropriate for the intended application, 2) the requirements specification for the VS program to ensure that it is complete and addresses the problem, and 3) the design to ensure that the VS program meets the requirements specification.

VS program coding was verified to ensure consistency against design. This was achieved through review of the source code, mathematical analysis of VS program functions and testing. As stated earlier, the VS program is subdivided into two parts, each “end” implemented through a different programming language. To address the challenges posed by the subdivided code, different criteria were used to perform the verification of each part. The FORTRAN 90 “back-end”, used for all numerical calculations, was verified for consistency with the Theory Manual and for good coding practice. The Visual Basic.NET “front-end”, used for launching program runs, manipulating data files and interacting with the user, was verified for consistency with the requirements specification, design description and good coding practice.

To ensure completeness, accuracy, clarity, consistency and traceability, the verification activities were performed through the use of checklists for the: 1) Theory Manual; 2) Requirements Specification; 3) Design Description; 4) Source code; 5) Acceptance Test Plan; 6) Verification Plan; and 7) Verification Report. Each checklist was composed of a series of questions that permitted the reviewer to answer with either "yes", "no" or "not applicable", as appropriate. Each question also corresponded with a "comments" field that gave the opportunity for concise feedback. Figure 2 illustrates one portion of the checklist used to review the Requirements Specification document.

VS Version 1.0 Verification

REQUIREMENT SPECIFICATION SECTION VERIFICATION CHECKLIST

Document Reference:
Date Verified:
Verified By:

	YES	NO	N/A	Comments
1. Is the Requirement Specification section complete?				
• Computer program name	☐	☐	☐	
• Programming language specification	☐	☐	☐	
• Functions of the computer program	☐	☐	☐	
• Interface requirement for hardware and software	☐	☐	☐	
• User of the computer program	☐	☐	☐	
• Operating System (OS) requirements	☐	☐	☐	
• Computational speed requirements	☐	☐	☐	
• Portability requirements	☐	☐	☐	
• File size and type requirements	☐	☐	☐	
• Input and output requirements	☐	☐	☐	
• Data structure and data flow requirements	☐	☐	☐	
• Fitting models and numerical algorithms	☐	☐	☐	
• Imposed physical or mathematical models requirements	☐	☐	☐	
• Error detection handling	☐	☐	☐	
• Accuracy targets	☐	☐	☐	
• Applications range	☐	☐	☐	
2. Are all requirements clear and organized (including tables and lists, where appropriate)?	☐	☐	☐	

Figure 2 One portion of the Requirements Specification verification checklist

The verification process was documented in an overall Verification Report, stating: 1) the requirements against which items were verified, 2) the methods and acceptance criteria, 3) the verification results, and 4) the disposition of anomalies. The Verification Report, alongside all other documents that accompany the VS program development lifecycle, has undergone documented review by qualified persons prior to being approved for use.

3.2 Validation

The VS software was validated for two main purposes:

- To ensure that the user is able to operate the software as described in the User Manual.
- To ensure the results obtained from the software are as expected by comparing the derived amplitudes obtained from the software against known amplitudes.

The VS program was validated as a whole, using artificial test cases. A MATLAB program was created to generate data for test purposes, using computational methods that differ from those employed by the VS program. Thus, a match of delayed fraction amplitudes obtained through either method would validate the VS computational approach.

To validate the results obtained from the VS program, three cases were studied:

3.2.1 First Case: "Accurate" artificial fluxes and detector responses

The "accurate" artificial model was designed with two purposes:

1. As MATLAB uses different algorithms than the VS program to generate the detector response data from the flux shape, the test case (indirectly) checks for the correctness of the mathematical approach behind the VS program.
2. The second purpose is to find an upper estimate of the computational error due to the VS program algorithm alone. When the VS program is used for a detector characteristics calculation, any error greater than the upper estimate can be attributed to an inaccuracy of the reactor flux rundown or detector response models.

The "accurate" artificial fluxes and detector responses model is generated through the following steps:

1. Artificial time-dependent flux is generated and looks similar in shape to the RFSP flux given in a "real" Shut-Off Rod (SOR) induced trip.
2. A set of randomly generated delayed detector amplitudes is used to construct the model for each detector.
3. MATLAB Simulink code is used to generate a time-dependent detector response from detector characteristics and the assumed flux.
4. Artificial fluxes and the corresponding detector responses are stored in VS input files, similar to the files generated by RFSP and SEDE[1], and are used for analysis of SOR-induced trips.
5. The sets of artificial amplitudes are stored for future comparison. The VS program is executed and the delayed and prompt amplitudes are computed for all artificial detector responses and fluxes. The VS-computed amplitudes are then compared to the artificially generated ones.

3.2.2 Second Case: "Mismatched" fluxes and detector responses

The process to execute the "mismatched" fluxes and detector responses model is the same as in the First Case, except that it differs by using a set of RFSP-computed fluxes as input for the VS program. The purpose of this case is to see which of the responses are most sensitive to inaccuracies in the flux model.

[1] To ensure feasible computation, the amount of data collected on each reactor trip had to be reduced. This reduction was achieved using the PLGS System Engineers Data Extraction (SEDE) application, where data was presented at increasing time steps for increased time intervals. The SEDE application can use HSDL data or SSMS data to build data sets.

3.2.3 Third Case: Artificial fluxes and noisy detector responses

The process to execute the case with added noise is the same as in the First Case, except that “random” noise is added to the computed detector response. The objective of this case is to investigate the effect of noise on the estimation of delayed detector amplitudes and on the maximum error of the fit algorithm.

In each case, the VS program-fitted results are compared against MATLAB-generated data. The following components were investigated for all three cases: 1) measured versus simulated detector signals; 2) maximum and minimum errors; 3) root-mean-square; 4) diagonals of the covariance matrix; 5) prompt fraction and delayed fraction; and 6) the quality of signal. This paper, however, only addresses analysis of the first two components for SOR-induced reactor trips.

4. Test Results and their interpretations

To assess the accuracy of results generated from the VS program, an estimation of error is necessary. The primary error estimate is found by subtracting the calculated signal from the actual signal at a given point in time. The range of error is found by taking the minimum and maximum of the primary error estimate for all points during the reactor trip. In the case of an optimal pointwise fit, the minimum and maximum errors should have equal absolute values and opposite signs.

4.1 First Case: “Accurate” artificial fluxes and detector responses

A MATLAB Simulink code used random detector characteristics (amplitudes) and the assumed run-down flux to generate a corresponding time-dependent detector response. The VS program was then executed and the delayed and prompt amplitudes were computed for all the artificial detector responses and fluxes. The VS-computed amplitudes were then compared to the artificially generated ones. Figure 3 presents the comparison between simulated and measured signals of the Prompt fraction (found on the top left) and the Delayed fractions for the First Case.

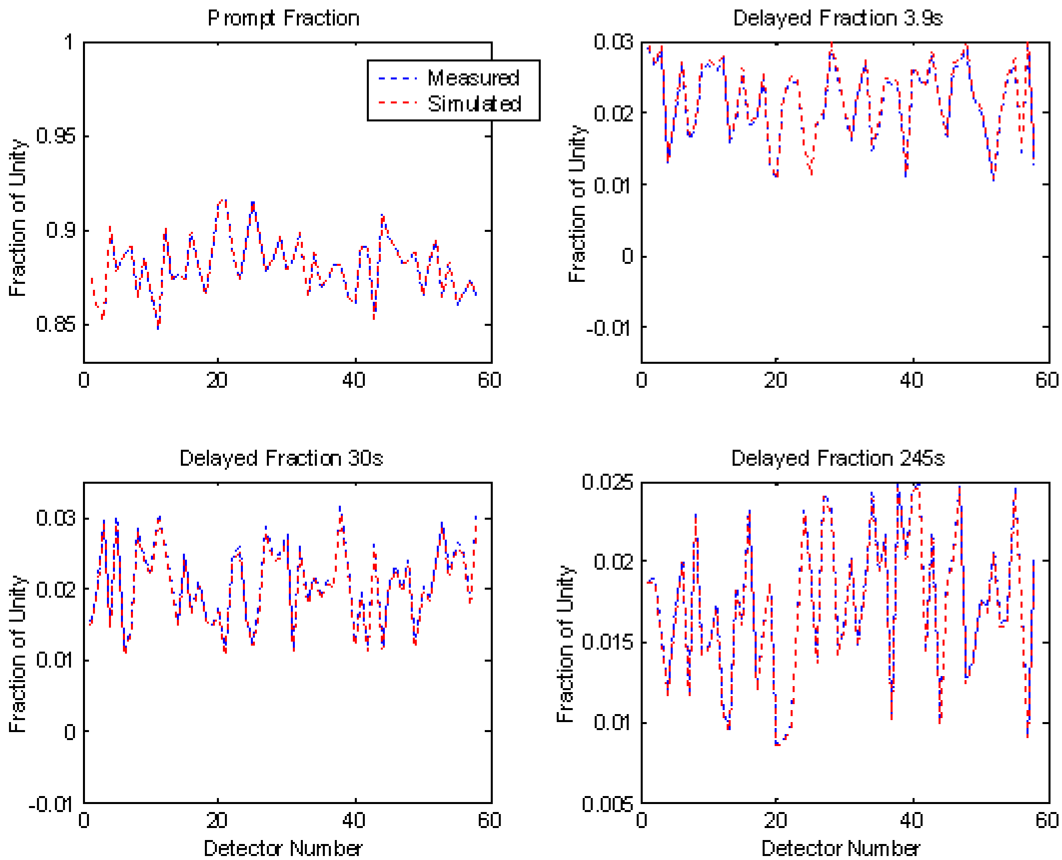

Figure 3 Prompt Fraction and Delayed Fractions – "accurate" artificial model

From the above figure, one can deduce that all the delayed signals match very closely. The Prompt and Delayed fractions show a difference of approximately 0.003 and 0.0005, respectively.

When analyzing a SOR-induced trip, the VS program takes input of five data sets containing the values of time-dependent neutron flux at each detector location, as well as the times at which these flux values were calculated. Each data set comprises of flux detector data recorded at a specific sampling frequency, centered on the reactor trip. The next illustration, Figure 4, presents the maximum and minimum errors of the simulated 58 detector signals for data five sets.

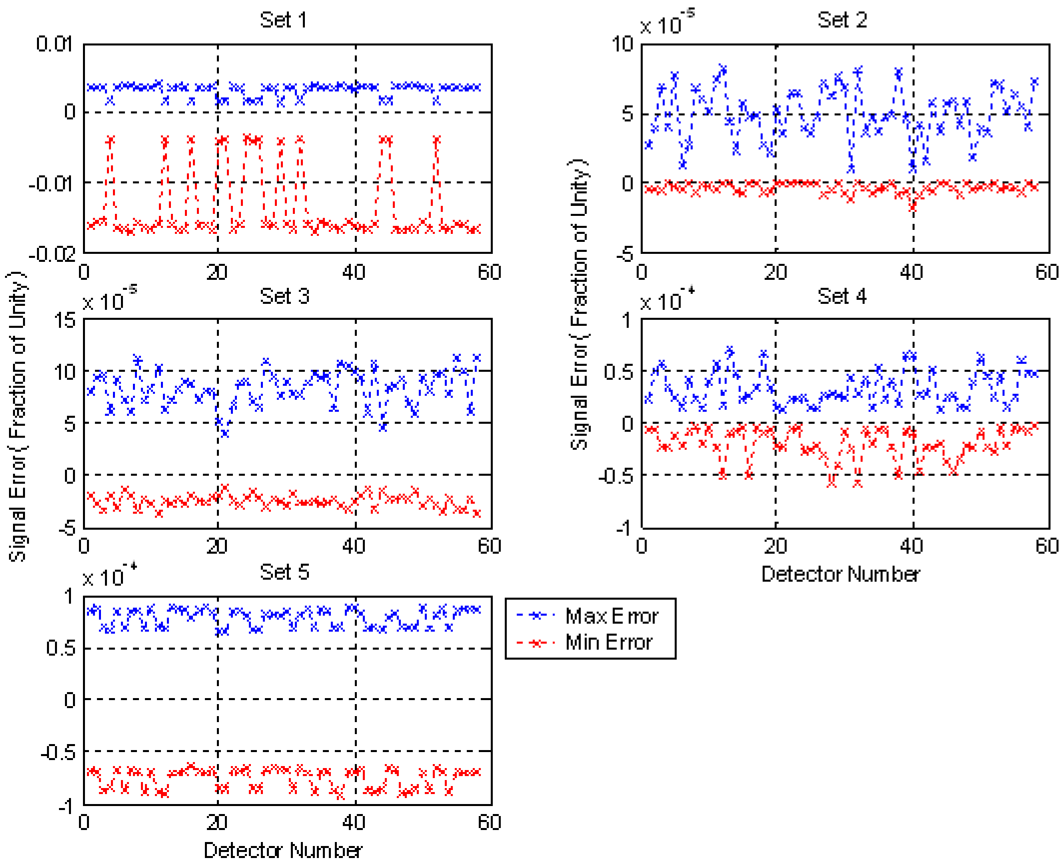

Figure 4 Maximum and minimum errors – "accurate" artificial model

The "accurate" artificial model produces very small errors. Set 1 shows a maximum and minimum error of approximately 0.005 and -0.015, respectively. Signal Set 1 is sampled at a sufficiently fast rate to capture the reactor trip, thus the set normally shows higher error due to the fast variation of flux during a trip. The remaining data sets are sampled at lower rates that only contain information to represent the flux significantly before and after a reactor trip. This is why Signal Set 1 normally gives higher maximum and minimum errors compared to Sets 2 through 5. The maximum and minimum errors give an estimate of the maximum contribution to error due to the algorithms used in the VS program alone.

4.2 Second Case: Mismatched fluxes and detector responses

The process to execute the Second Case is the same as in the First Case, except that it differs by using a set of RFSP-computed fluxes as input for the VS program. The purpose of this case is to see which of the responses are most sensitive to inaccuracies in the detector model.

The Second Case shows that the least square algorithm does not over-fit values in the Signal Sets when the RFSP-estimated flux is not correct. The most significant error was found using Signal Set 1 as the flux trip is mostly affected by the RFSP model assumptions.

As illustrated in Figure 5, signal Set 1 normally gives a higher maximum and minimum error compared to Sets 2 through 5.

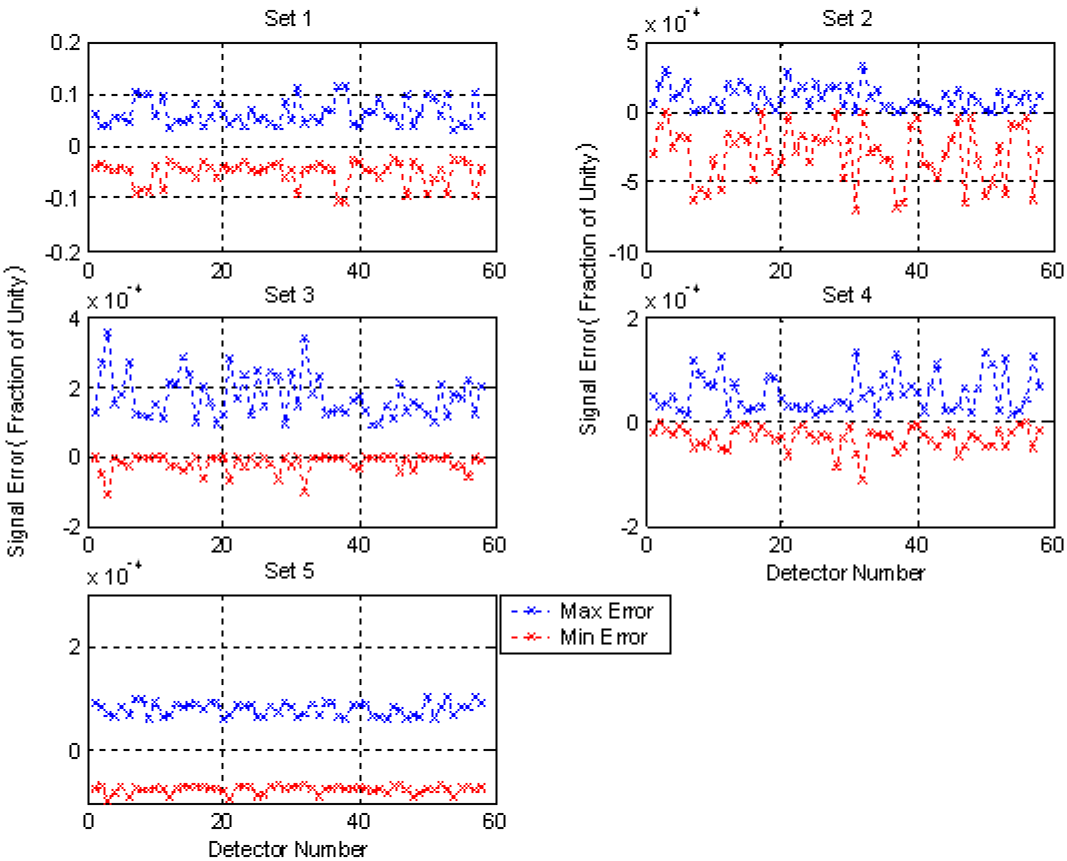

Figure 5   Maximum and minimum errors – "mismatched" model

From the above figure, Set 1 shows a maximum error of 0.05 and minimum error of -0.05.

4.3 Third Case: Artificial fluxes and noisy detector responses

The process to execute the case with added noise is the same as in the First Case, except that random noise is added to the computed detector response. The objective of this case is to investigate the effect of noise on the estimation of delayed detector amplitudes and on the maximum error of the fit algorithm.

In the Third Case, the absolute values of the maximum and minimum errors in Sets 2 through 5 are given by the maximum values of the random error. Therefore, random noise decreases the accuracy of the delayed detector amplitudes.

5. Conclusions

Validation testing of the VS program, intended to evaluate the dynamic compensation of safety-system Platinum-clad In-Core Flux Detectors, resulted in the following conclusions:

a) Tests involving the computed flux and detector response data show a high degree of accuracy (approximately 0.1 percent) for the method used to compute the delayed fractions. In addition to proving that the program works with a high degree of accuracy, the tests show that any error exceeding 0.1 percent can be attributed to factors such as noise or inaccuracy in either the detector model itself or in the flux evaluation. Unfortunately the program cannot distinguish between these factors nor can it indicate which factor is predominant.

b) The test with noise added to the "accurate" artificial model shows that the results of the fitting algorithm accurately depict the noise input.

c) Incorporating the strengths of both programming languages, FORTRAN 90 and VB.NET, in the VS program ensures the demands of detailed numerical calculations and compatibility with various hardware and software environments are not compromised.

In summary, the extent of qualification necessary for the VS program was dependent on the scope of the program and its safety implications on plant workers, public and the environment. The VS program may be used to monitor the performance of safety system ICFDs. More specifically, output of the program may be used to estimate the dynamic compensation error of ICFDs in the analysis of the ROP system. Consequent to the intended scope, a verification and validation program has been carried out to ensure the VS program is compliant to the CSA N286.7-99 Standard.

6. References

[1] B. Rouben, "RFSP-IST, The Industry Standard Tool Computer Program for CANDU Reactor Core Design and Analysis", in Proceedings of PBNC-2002 (12th Pacific Basin Nuclear Conference), Shenzhen, China, 2002 October.

[2] V.N.P. Anghel, et al., "Evaluation of Slow Shutdown System Flux Detectors in Point Lepreau Generating Station. I: Dynamic Response Characterization", Proceedings of the 30th Annual Conference of the Canadian Nuclear Society, Calgary, Alberta, Canada, 2009, May 31 – June 3.

[3] V.N.P. Anghel, B. Sur, D. Taylor, "Evaluation of Slow Shutdown System Flux Detectors in Point Lepreau Generating Station. II: Dynamic Compensation Error Analysis", Proceedings of the 30th Annual Conference of the Canadian Nuclear Society, Calgary, Alberta, Canada, 2009, May 31 – June 3.

[4] C. M. Bailey, R. D. Fournier, and F. A. R. Laratta, "Regional Overpower Protection in CANDU Power Reactors", International Meeting on Thermal Nuclear Reactor Safety, Chicago, Illinois, 1982, August 29 - September 2.

[5] C. M. Bailey, M. Nguyen, B. Sur, "A Proposed Method for Assessing In-Core Flux Detector Dynamic Compensation Adequacy Over a Range of Trip Times", Proceedings of the 22nd Annual Conference of the Canadian Nuclear Society, Toronto, Ontario, Canada, 2001.