RAVEN: A TOOL FOR PROBABILISTIC RISK ASSESSMENT

A. Alfonsi, C. Rabiti, D. Mandelli, J.J. Cogliati, R.A. Kinoshita

Idaho National Laboratory

{andrea.alfonsi, cristian.rabiti, diego.mandelli, joshua.cogliati, robert.kinoshita}@inl.gov

INTRODUCTION

RAVEN (**R**eactor **A**nalysis and **V**irtual control **EN**viroment) [1, 2] is a software framework that acts as the control logic driver for the Thermo-Hydraylic code RELAP-7, a newly developed software at Idaho National Laboratory. The aim of this paper is to provide an overview of the software structure and its utilization in conjunction with RELAP-7/MOOSE [3, 4]. RAVEN is a multi-purpose Probabilistic Risk Assement (PRA) code that allows dispatching different functionalities. It is designed to derive and actuate the control logic required to simulate the plant control system and operator actions (guided procedures) and to perform both Monte-Carlo sampling of random distributed events and dynamic event tree based analysis [5]. In order to assist the user in the input/output handling, a Graphical User Interface (GUI) and a post-processing data mining module, based on dimensionality and cardinality reduction [6], are available. This paper wants to point up the link between the software layout and the mathematical framework from which its structure is derived. In order to show some capabilities, a demo of a Station Black Out (SBO) analysis of a simplified Pressurized Water Reactor (PWR) model is reported.

MATHEMATICAL FRAMEWORK

Let be $\bar{\theta}(t)$ a vector describing the plant status in the phase space; the dynamic of both plant and control system can be summarized by the following equation:

$$\frac{\partial\bar{\theta}}{\partial t} = \bar{H}(\theta(t), t) \tag{1}$$

In the above equation it is assumed the time differentiability in the phase space. Performing an arbitrary decomposition of the phase space, it is obatined the following statement:

$$\bar{\theta} = \begin{pmatrix} \bar{x} \\ \bar{v} \end{pmatrix} \tag{2}$$

The decomposition is made in such a way that $\bar{x}$ represents the unknowns solved by RELAP-7, while $\bar{v}$ are the variables directly controlled by the control system (i.e., RAVEN). Equation 1 can now be rewritten as follows:

$$\begin{cases} \dfrac{\partial\bar{x}}{\partial t} = \bar{F}(\bar{x}, \bar{v}, t) \\ \dfrac{\partial\bar{v}}{\partial t} = \bar{V}(\bar{x}, \bar{v}, t) \end{cases} \tag{3}$$

It is possible to note that the function $\bar{V}(\bar{x}, \bar{v}, t)$ representing the control system, does not depend on the knowledge of the complete status of the system but on a restricted subset (i.e. control variables) $\bar{C}$:

$$\begin{cases} \dfrac{\partial\bar{x}}{\partial t} = \bar{F}(\bar{x}, \bar{v}, t) \\ \bar{C} = \bar{G}(\bar{x}, t) \\ \dfrac{\partial\bar{v}}{\partial t} = \bar{V}(\bar{x}, \bar{v}, t) \end{cases} \tag{4}$$

The system of equations in Eq. 4 is fully coupled and has always been solved with an operator splitting approach. The reasons for this choice are several:

- Control system reacts with an intrinsic delay
- The reaction of the control system might move the system between two different discrete states and therefore numerical errors will be always of first order unless the discontinuity is treated explicitly.

Thus, RAVEN is using this approach to solve Eq. 4 which becomes:

$$\begin{cases} \dfrac{\partial\bar{x}}{\partial t} = \bar{F}(\bar{x}, \bar{v}_{t_i-1}, t) \\ \bar{C} = \bar{G}(\bar{x}, t) \\ \dfrac{\partial\bar{v}}{\partial t} = \bar{V}(\bar{x}, \bar{v}_{t_i-1}, t) \end{cases} \tag{5}$$

Even if all information needed is contained in $\bar{x}$ and $\bar{v}$, it is not a practical and efficient way to implement the control system. Hence, a system of auxiliary variables has been introduced.

The auxiliary variables are those that in statistical analysis are artificially added to non-Markovian systems into the space phase to obtain a Markovian behavior back, so that only the information of the previous time step is needed to determine the future status of the system. These variables can be classified into two types:

- Global status auxiliary control variables (e.g., SCRAM status, etc.)
- Component status auxiliary variables (e.g., correct operating status, etc.)

Thus, the introduction of the auxiliary system into the mathematical framework leads to the following formulation

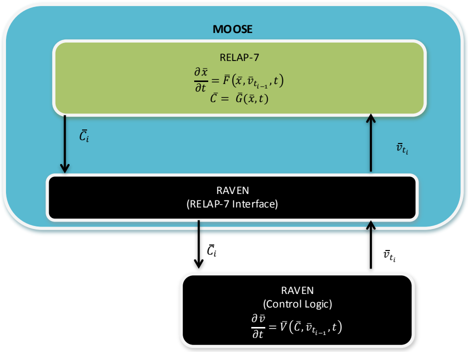
MOOSE
RELAP-7
$\frac{\partial \bar{x}}{\partial t} = \bar{F}(\bar{x}, \bar{v}_{t_{i-1}}, t)$
$\bar{C} = \bar{G}(\bar{x}, t)$
$\bar{C}_i$
$\bar{v}_{t_i}$
RAVEN
(RELAP-7 Interface)
$\bar{C}_i$
$\bar{v}_{t_i}$
RAVEN
(Control Logic)
$\frac{\partial \bar{v}}{\partial t} = \bar{V}(\bar{C}, \bar{v}_{t_{i-1}}, t)$

Fig. 1. Monte-Carlo sampling scheme.

of the Eq. 5:

$$\begin{cases} \dfrac{\partial \bar{x}}{\partial t} = \bar{F}(\bar{x}, \bar{v}_{t_i-1}, t) \\ \bar{C} = \bar{G}(\bar{x}, t) \\ \dfrac{\partial \bar{a}}{\partial t} = \bar{A}(\bar{x}, \bar{C}, \bar{a}, \bar{v}_{t_i-1}, t) \\ \dfrac{\partial \bar{v}}{\partial t} = \bar{V}(\bar{x}, \bar{v}_{t_i-1}, t) \end{cases} \tag{6}$$

SOFTWARE STRUCTURE

RAVEN, is plugged with the software enviroment MOOSE [3]. MOOSE is a computer simulation framework, developed at Idaho National Laboratory (INL), that simplifies the process for predicting the behavior of complex systems and developing non-linear, multi-physics simulation tools.Other than providing the algorithms for the solution of the differential equation, MOOSE also provides all the manipulation tools for the C++ classes containing the solution vector. This framework has been used to construct and develop the Thermo-Hydraulic code RELAP-7, giving an enormous flexibility in the coupling procedure with RAVEN.

RELAP-7 is the next generation nuclear reactor system safety analysis. It will become the main reactor systems simulation toolkit for RISMC (**R**isk **I**nformed **S**afety **M**argin **C**haracterization) [7] project and the next generation tool in the RELAP reactor safety/systems analysis application series. RAVEN has been developed in a high modular and pluggable way in order to enable easy integration of different programming languages (i.e., `C++`, `Python`) and coupling with other applications including the ones based on MOOSE. The code consists of four main modules:

- RAVEN/RELAP-7 interface
- `Python` Control Logic
- `Python` Calculation Driver
- Graphical User Interface

The RAVEN/RELAP-7 interface, coded in `C++`, is the container of all the tools needed to interact with RELAP-7/MOOSE. It has been designed in order to be general and pluggable with different solvers simultaneously in order to allow an easier and faster development of the control logic/PRA capabilities for multi-physics applications. The interface provides all the capabilities to control, monitor, and process the parameters/quantities in order to drive the RELAP-7/MOOSE calculation. In addition, it contains the tools to communicate to the MOOSE input parser whose information, i.e. input syntax, must be received as input in order to run a RAVEN calculation. The control logic module is used to drive a RAVEN/RELAP-7 calculation. Up to now it is implemented by the user via `Python` scripting. The reason of this choice is to try to preserve generality of the approach in the initial phases of the project so that further specialization is possible and inexpensive. The implementation of the control logic via `Python` is rater convenient and flexible. The user only needs to know few `Python` syntax rules in order to build an input. Although this extreme simplicity, it will be part of the GUI task to automatize the construction of the control logic scripting in order to minimize user effort.

The core of PRA analysis is contained in the module called "Raven Runner". It consists in a `Python` driver in which Monte-Carlo based algorithm has been implemented. It calls RAVEN multiple times, changes initial conditions and seeds the random generator for the distributions. The multiple calculations, required by the employment of these algorithms, can be run in parallel, using queues/sub-process/`Python` systems. The analysis of dynamic stochastic systems through Monte-Carlo algorithm can be summarized as follows:

1. Initial Sampling of:
   - (a) Static and dynamic uncertainty values of physical parameters
   - (b) Initial conditions
   - (c) Transition conditions, i.e. time instant in which transition events occur (e.g., time in which a reactor scram occurs, etc.)
2. Run the system simulator using the values previously sampled and eventually applying a random noise to some parameters at each time step
3. Stop the simulation when a transition condition occurs, and move from the actual status of the system to the new one
4. Run the simulation as performed in step 3 starting from the new coordinates and stop when a new transition condition occurs;
5. Repeat steps 3 and 4 until a stopping condition is reached
6. Repeat 1 through 4 for a large number of calculations (user input)

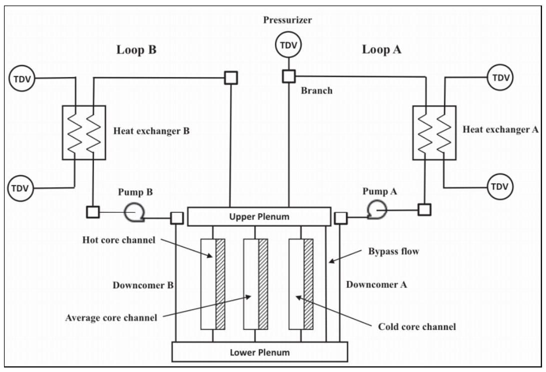

Fig. 2. PWR model scheme.

The "runner" basically performs a different seeding of the random number generator and interact, through RAVEN, with the Python control logic input in order to sample the variables specified by the user.

As previously mentioned, a Graphical User Interface (GUI) is not required to run RAVEN, but it represents an added value to the whole code. The GUI is compatible with all the capabilities actually present in RAVEN (control logic, Monte-Carlo, etc.). Its development is performed using QtPy, which is a Python interface for a C++ based library (C++) for GUI implementation. The GUI is based on a software named Peacock, which is a GUI interface for MOOSE based application and, in its base implementation, is only able to assist the user in the creation of the input. In order to make it fit all the RAVEN needs, the GUI has been specialized and is in continuous evolution.

DEMO FOR A PWR PRA ANALYSIS

In order to show the capabilities of RAVEN coupled with RELAP-7/MOOSE, a PRA analysis on a simplified PWR model (Fig. 2) has been employed.
Since RELAP-7 still has limitations for the component controllable parameters and models, it has been necessary to act on unconventional factors (i.e. inlet/outlet friction).

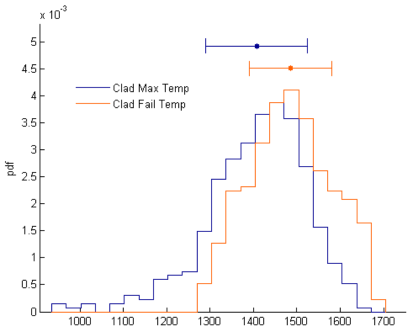

Fig. 3. Comparison between max reached clad temperature and clad failure temperature distributions: Probability distribution functions.

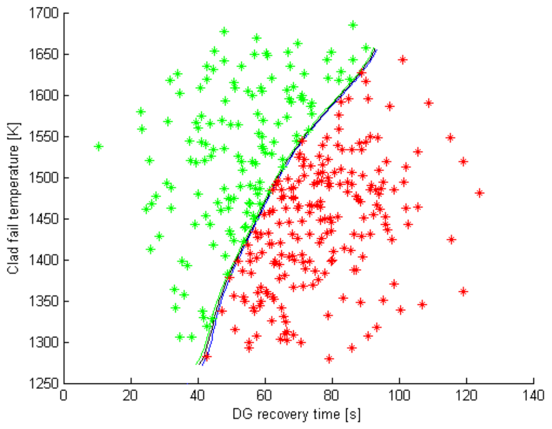

Fig. 4. Limit Surface for the SBO analysis of a simplified PWR model

The Probabilistic Risk Assessment analysis has been performed simulating a Station Black Out accident, running Monte-Carlo samplings (400 simulations) on the recovery time of the diesel generators t_1 (Normal distribution, mu = 120 s, sigma = 20 s) and the clad failure temperature TCf(Triangular distribution, xPeak = 1477.59 K, xMin = 1255.37 K, xMax = 1699.82 K). Since the scope of this demo is to show the functionalities contained in RAVEN and RELAP-7 capabilities are not optimized for long simulation times, the transient has been accelerated in order to simulate a maximum of 300 seconds. Figure 3 shows the distribution of the maximum temperature reached by the clad in the core channels (blue histogram) and compares it with the distribution of clad failure temperature (red histogram). As already mentioned, the transient has been accelerated, since the scope of the analysis was just to show RAVEN capabilities to perform stochastic analysis of relatively complex systems. That can explain the large overlapping of the two distributions, which indicates a high failure probability of the system considered.

Figure 4 shows the limit surface, i.e. the boundaries between system failure (red points) and system success (green points), obtained by the 400 Monte-Carlo simulations. Since only two uncertain parameters have been considered (i.e., DG recovery time and clad fail temperature), this boundary lies in a 2-dimensional space. The slope of the limit surface pictured in Fig. 4 also shows, in this particular demo, how the DG recovery time has a greater impact on the system dynamics then the clad failure temperature.

CONCLUSIONS

In this paper it has been presented RAVEN as a tool to perform dynamic PRA through Monte-Carlo sampling. In particular, the software structure and all the components that are involved in the computation have been presented, including system simulator (i.e., RELAP-7) and the control logic, characterized by monitor system dynamics and on-line control of selected parameters. An example of PRA analysis has been also presented for a SBO-like case for a simplified PWR loop. The description of the implementation for such case demonstrates how the flexibility of the software framework provides the basic tools to perform Dynamic PRA, uncertainty quantification and plant control. Next capabilities, to be implemented to RAVEN and that are currently under development, include dynamic event tree generation [5], adaptive sampling [8] and more advanced data mining algorithms [6].

REFERENCES

1. C. RABITI, A. ALFONSI, J. COGLIATI, D. MANDELLI, and R. KINOSHITA, "REACTOR ANALYSIS AND VIRTUAL CONTROL ENVIRONMENT (RAVEN) FY12 REPORT," Tech. Rep. INL/EXT-12-27351, Idaho National Laboratory (INL) (2012).
2. C. RABITI, A. ALFONSI, D. MANDELLI, J. COGLIATI, and R. MARTINEAU, "RAVEN as Control Logic and Probabilistic Risk Assessment Driver for RELAP-7," in "Proceeding of American Nuclear Society (ANS), San Diego (CA)," (2012), vol. 107, pp. 333–335.
3. D. GASTON, G. HANSEN, S. KADIOGLU, D. A. KNOLL, C. NEWMAN, H. PARK, C. PERMANN, and W. TAITANO, "Parallel multiphysics algorithms and software for computational nuclear engineering," *Journal of Physics: Conference Series*, **180**, 1, 012012 (2009).
4. D. ANDERS, R. BERRY, D. GASTON, R. MARTINEAU, J. PETERSON, H. ZHANG, H. ZHAO, and L. ZOU, "RELAP-7 Level 2 Milestone Report: Demonstration of a Steady State Single Phase PWR Simulation with RELAP-7," Tech. Rep. INL/EXT-12-25924, Idaho National Laboratory (INL) (2012).
5. A. HAKOBYAN, T. ALDEMIR, R. DENNING, S. DUNAGAN, D. KUNSMAN, B. RUTT, and U. CATALYUREK, "Dynamic generation of accident progression event trees," *Nuclear Engineering and Design*, **238**, 12, 3457 – 3467 (2008).
6. D. MANDELLI, A. YILMAZ, and T. ALDEMIR, "Scenario Analysis and PRA: Overview and Lessons Learned," in "Proceedings of European Safety and Reliability Conference (ESREL 2011), Troyes (France)," (2011).
7. D. MANDELLI and C. SMITH, "Integrating Safety Assessment Methods Using the Risk Informed Safety Margins Characterization (RISMC) Approach," in "Proceeding of American Nuclear Society (ANS), San Diego (CA)," (2012), vol. 107, pp. 883–885.
8. D. MANDELLI and C. SMITH, "Adaptive Sampling Using Support Vector Machines," in "Proceeding of American Nuclear Society (ANS), San Diego (CA)," (2012), vol. 107, pp. 736–738.