Introduction of Supervised Learning Capabilities of the RAVEN Code for Limit Surface Analysis

Cristian Rabiti*, Diego Mandelli, Andrea Alfonsi, Joshua Cogliati, Robert Kinoshita

Idaho National Laboratory, 2525 North Fremont Street, Idaho Falls (ID)
**Corresponding Author: cristian.rabiti@inl.gov*

INTRODUCTION

The development of the RAVEN code [1] [9] started in 2012 to provide the needed capabilities to the Risk Informed Safety Margins Characterization (RISMC) pathway [2] with the capability to forecast safety margins for nuclear power plant accident scenarios. The goal was achieved by a three way approach providing: 1) A graphical user interface for the RELAP-7 code, 2) A control logic/scenario generator framework for RELAP-7, and 3) A statistical analysis framework to assess safety margins.

This paper focuses on the latest addition to the statistical analysis framework, a library of supervised learning algorithms to determine the location of limit surfaces [10].

THE IMPORTANCE AND DEFINITION OF LIMIT SURFACES IN RISK ANALYSIS

The analysis of risk, in the engineering field, refers to the analysis of the probability that a certain event will take place multiplied by the consequences associated that event.

More specifically in a mathematical formulation given:

$\bar{x} \in X$: where X is phase space of the system S,

$p(\bar{x})$: probability distribution function of $\bar{x}$,

$c(\bar{x})$: consequences associated to the outcome $\bar{x}$.

Risk, or more specifically the differential contribution to risk, is defined by:

$$p(\bar{x})c(\bar{x})d\bar{x} = r(\bar{x})d\bar{x}.$$

As a consequence the total risk connected to a certain system S is given by:

$$R = \int_X p(\bar{x})c(\bar{x})d\bar{x}.$$

Currently the focus of the RAVEN project, with respect to Probabilistic Risk Assessment (PRA), is limited to accident scenarios where the nuclear fuel integrity has not yet been compromised. In our applications, a binary cost function is defined as:

$$c(\bar{x}) = \begin{cases} 1, & if\ fuel\ integrity\ is\ compromised \\ 0, & if\ fuel\ integrity\ is\ not\ compromised \end{cases}$$

Consequently the total risk evaluation could be reduced to:

$$R = \int_F p(\bar{x})d\bar{x},$$

where the failure region F is defined by:

$$F = \{\bar{x} | \bar{x} \in X, c(\bar{x}) = 1\}$$

The limit surface ∂F is defined as the boundaries of F. ∂F may be identified by a system of constraints that will be indicated by $\bar{f}(\bar{x}) = 0$.

The importance of the limit surface is clear when ∂F can be expressed in terms of the initial conditions of the system rather than its location in the phase space $\bar{x}$. The possibility of defining the constraints representing the limit surface location in terms of the input space of the system, is related to the stochastic characterization of the system S and its phase space definition. Since S is usually a dynamic stochastic system, it is not possible to establish, for a given set of initial condition $\bar{x}_0$, a unique trajectory $\bar{x}(t, \bar{x}_0)$. This situation might lead to the impossibility to define the location of the limit surface in terms only of the initial condition of the system.

Reference [3] describe more in detail what are the characteristics, of a dynamic stochastic system, under which the phase space and the corresponding input set, could be expanded so that the behavior of the system is fully determined once the initial conditions are set.

When it is possible to uniquely determine the system trajectory for a given set of initial conditions the limit surface in the initial condition space (to be intended as the extended initial condition set as described in [3]) is determined by:

$$\partial \bar{x}_{0,LS} = \left\{\bar{x}_0 | \exists\ t \in [0, t_{end}], \bar{f}\left(\bar{x}(t, \bar{x}_0)\right) = 0\right\},$$

where t_{end} is the maximum time during which the system is monitored. Similarly the input region that is mapped into F by the function representing the system evolution $\bar{x}(t, \bar{x}_0)$ is:

$$\bar{x}_{0,LS} = \left\{\bar{x}_0 | \exists\ t \in [0, t_{end}], c\left(\bar{x}(t, \bar{x}_0)\right) = 1\right\}$$

Knowing $\bar{x}_{0,LS}$ in the input space has a clear value since it allows to establishing which region of the input space will lead to a an evolution of the system toward a specific outcome.

The risk integral with respect the in the input space is then given by:

$$R = \int_0^{t_{end}} dt \int_{\bar{x}_{0,LS}} p\left(\bar{x}(t, \bar{x}_0)\right) c\left(\bar{x}(t, \bar{x}_0)\right) d\bar{x}\,(t, \bar{x}_0)$$

For the specific binary cost function defined earlier:

$$R = \int_F p(\bar{x})d\bar{x} = \int_{\bar{x}_{0,LS}} p(\bar{x}_0)d\bar{x}_0$$

IDENTIFICATION OF THE LIMIT SURFACE

The problem of the location of the limit surface is to determine the non-convex hull surrounding the failure region. This type of problem are known for having a large

computational complexity which reduction is very difficult.

Going back to our simple cost function where $R = \int_F p(\bar{x})d\bar{x} = \int_{\bar{x}_{0,LS}} p(\bar{x}_0)d\bar{x}_0$, we can build a Cartesian grid in the input space such that $\Delta x_0^{i,j}$ and $x_0^{i,j}$ satisfies:

$$\int_{x_0^{i,j}}^{x_0^{i,j}+\Delta x_0^{i,j+1}} p(x_0^i)dx_0^i = \frac{1}{n_i}$$
$$x_0^{i,0} = min\{x_0^i\}$$
$$x_0^{i,n_i+1} = max\{x_0^i\}$$
$$i = 1, \dots, \mathrm{m} = \dim[\bar{x}_0]$$
$$j_i = 0, \dots, n_i$$
$$x_0^{i,j_i+1} = x_0^{i,j_i} + \Delta x_0^{i,j_i+1},$$

so that $\Delta V_0^l = \prod_{i=1}^n \Delta x_0^{i,j_i}$ is the control volume (l is the global index) and the barycenter (probability weighted) is given by:

$$\bar{x}_0^l = \left(x_0^{1,j_1+1/2}, \dots, x_0^{i,j_i+1/2}, \dots, x_0^{m,j_m+1/2}\right),$$

where:

$$\int_{x_0^{i,j_i}}^{x_0^{i,j_i+1/2}} p(x_0^i)dx_0^i = \int_{x_0^{i,j_i+1/2}}^{x_0^{i,j_i+1}} p(x_0^i)dx_0^i.$$

Note that the grid built is a *regular* Cartesian grid under the transformation $p(x_0^i)dx_0^i = dx_0^{i'}$, being therefore an equal probable Cartesian partition of the input space.

The approximations of $\bar{x}_{0,LS}$ and $\partial\bar{x}_{0,LS}$ can be written as:

$$\bar{x}_{0,LS} \approx \bigcup_j \left\{\Delta V_0^l | \exists\, t \in [0, t_{end}], c\left(\bar{x}(t, \bar{x}_0^l)\right) = 1\right\}$$

$$\partial\bar{x}_{0,LS} \approx \bigcup_j \left\{\bar{x}_0^l \middle| \bar{x}_0^l \in \bar{x}_{0,LS}, \exists i \middle| \bar{x}_0^l - \left(0, \dots, -x_0^{i,j_i+\frac{1}{2}} + x_0^{i,j\mp1\mp\frac{1}{2}}, \dots, 0\right) not\ \in \bar{x}_{0,LS}\right\}$$

Considering that:

- the number of points on the grid, to achieve a reasonable accuracy, can be rather large
- the assessment of $c\left(\bar{x}\left(t, \bar{x}_0^{j+1/2}\right)\right) = \begin{cases}1\\0\end{cases}$ requires the evaluation of the system response, which in our case is a quite expensive RELAP-7 simulation

in most of the cases a brute force approach leads to unreasonable computational costs.

For this reason several acceleration schemes have been proposed to overcome this challenge. RAVEN takes advantage of the fact that most of these schemes follow the same functional pattern so it is possible to implement a software layout that allows a quick integration of any new algorithm.

GENERAL ACCELERATION SCHEME FOR THE DETERMINATION OF THE LIMIT SURFACE

The determination of the limit surface, relying on the evaluation of the system response on a Cartesian grid in the input space, is too expensive in most PRA cases. One of the classes of acceleration schemes to speed up the process is based on Surrogate Models (SMs). For the moment we can think of a SM as a mathematical model capable of approximating the system response $\bar{x}(t, \bar{x}_0)$ at a fraction of the computational cost of the evaluation of the original model.

Generally, the error related to this approximation depends on the variation of the system response with respect to the input space ($\frac{\partial\bar{x}(t,\bar{x}_0)}{\partial x_0^i}$), from the choice of the SM and from the number of constraints used to determine the values of the parameters characterizing the SM.

While SMs are used in a large range of mathematical end engineering fields, here the naming conventions familiar in the artificial intelligence research field are used. According to this choice the process of determining the parameters of the SM is named as the *fit step*, the process of predicting the system response as the *predict step*, and the overall process as a *supervised learning process*.

In particular, RAVEN implements a scheme that is more specifically called *active supervised learning process*. Active processes differ in that they allow recursive training of the SM. The overall scheme is represented in Figure 1.

To test the convergence of the iterative process, we are considering several possible options :

- Test the convergences of the parameters describing the SM
- Test the convergence of $\partial\bar{x}_{0,LS}$

Either of these tests can be performed using different types of norm such as L_1, L_2, L_∞, etc.

The general policy for choosing the next point where the full system model is evaluated is to maximize the information gain concerning the variation of the system response with respect the input parameters. Several options are possible.

LIMIT SURFACE EXAMPLE

To clarify the limit surface concept, it may be helpful to show an example that has been obtained from post processing of large Monte Carlo samplings generated in the past by RAVEN and RELAP-7 [4, 5].

The type of system and accident scenario examined is explained in mode detail in [5]: a Pressurized Water Reactor (PWR) station blackout scenario. For this

scenario, the considered input parameters are the recovery time of the auxiliary diesel generators (DGs) t_{DG}, the auxiliary power grid connection t_{RSST}, and the main power grid connection t_{T138}. Core damage is reached unless AC power is recovered by either external (the two power grid connections) or internal (the DGs) means.

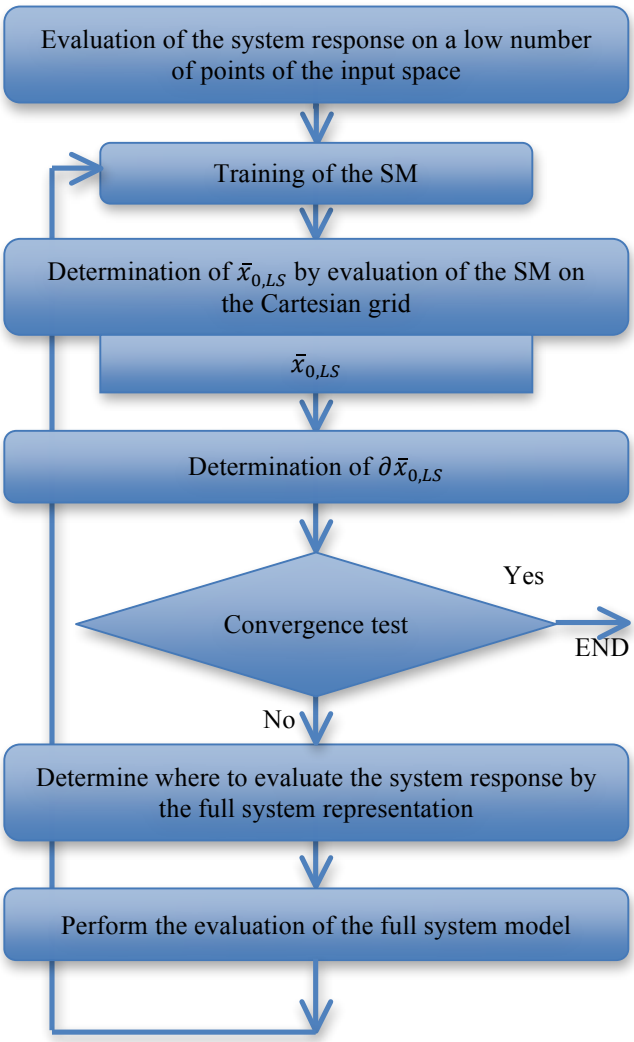

Figure 1: Software scheme of the active super-vised learning acceleration to determine the location of the LS.

In other terms, outcome (core damage) is determined by comparing the time to reach fuel failure temperature and the AC recovery time $t_{AC\_rec} = min(t_{DG}, t_{RSST}, t_{T138})$. For such case, the LS in this 3-dimensional space $(t_{DG}, t_{RSST}, t_{T138})$ is shown in Figure 2.

RAVEN IMPLEMENTATION AND TESTING

RAVEN implements a scheme very similar to the one described in figure 1. More specifically:

1. The input points used to generate the initial training set are either externally generated or chosen using a Monte Carlo approach
2. The SMs currently available to be trained is the support vector machine based classifier available through the scikit-learn software library [6], and the closest neighbor based classification.
3. The location of the $\partial\bar{x}_{0,LS}$ is determined on a Cartesian grid generated from user input.
4. The convergence test requires that none of the barycenter points of the grid will change its classification between two iterations, and this should be true for a certain number of times (user input).
5. The next point to be tested using the full system representation is chosen as the barycenter point of the grid laying on $\partial\bar{x}_{0,LS}$ which is the farthest point from any other already sampled in the input space.
6. The software tools representing physical systems available in this scheme are: RELAP-7, RELAP5-3D [7], any MOOSE-based code [8]. A general Application Programming Interface (API) is also provided to allow independent implementation of a physical system representation.

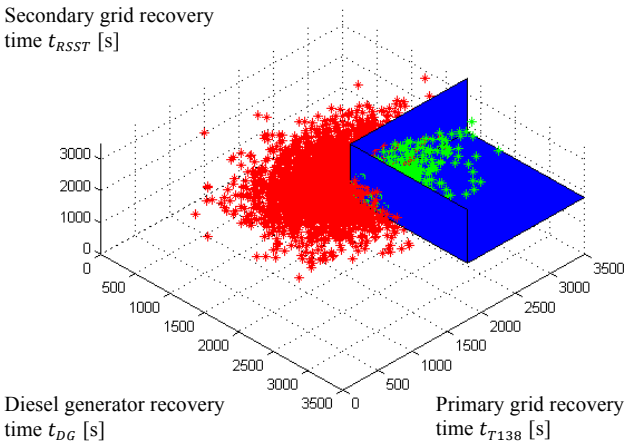

Figure 2: LS for PWR Station Black Out

Currently the whole implementation is in an advanced stage of development, Figure 3 shows how, after few iterations, the training point (blue) are distributed along the LS given defined by $x^2 + y^2 - 25 = 0$.

Real case applications are currently ongoing based on RELAP-7 RAVEN simulations and more results will be added soon.

CONCLUSIONS

A general scheme for limit surface searching has been implemented in RAVEN, characterized by very

general and flexible APIs. These allow the use of established external software to augment most steps of the active supervised learning process: generation of the initial training set, creating the supervised algorithm, and producing full system representation code. The generation of the initial training set may also be done with any of the samplers already implemented in RAVEN (Monte Carlo, Latin Hypercube, and custom made grids).

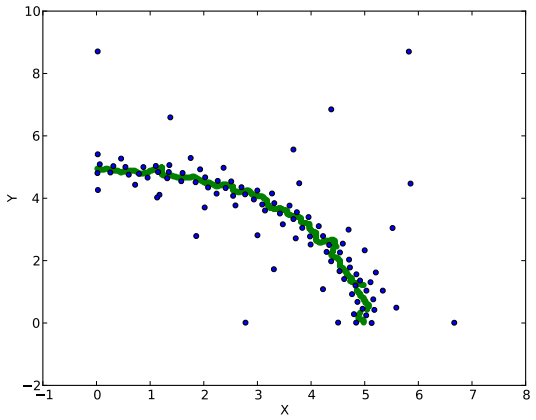

Figure 3: Example of LS (green line) and adaptive sampling (blue points chosen along the LS) in a 2-dimensional space

REFERENCES

1. C. RABITI, A. ALFONSI, J. COGLIATI, D. MANDELLI, R. KINOSHITA, "Reactor Analysis and Virtual Control Environment (RAVEN) FY12 Report," Idaho National Laboratory report: INL/EXT-12-27351 (2012).

2. R. W. YOUNGBLOOD, V. A. MOUSSEAU, D. L. KELLY, and T.N. DINH, "Risk-Informed Safety Margin Characterization (RISMC): Integrated Treatment of Aleatory and Epistemic Uncertainty in Safety Analysis," *The 8th International Topical Meeting on Nuclear Thermal-Hydraulics, Operation and Safety (NUTHOS-8)* Shanghai, China, October 10-14, 2010.

3. C. RABITI, D. MANDELLI, A. ALFONSI, J. COGLIATI, R. KINOSHITA "Mathematical Framework for the Analysis of Dynamic Stochastic Systems," *International Conference on Mathematics and Computational Methods Applied to Nuclear Science & Engineering M&C2013* (2013).

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," Idaho National Laboratory report: INL/EXT-12-25924 (2012).

5. C. RABITI, A. ALFONSI, D. MANDELLI, J. COGLIATI, R. MARTINEAU, C. SMITH, "Deployment and Overview of RAVEN Capabilities for a Probabilistic Risk Assessment Demo for a PWR Station Blackout," Idaho National Laboratory report: INL/EXT-13-29510 (2013).

6. F. PEDREGOSA, A. GRAMFORT, B. THIRION, O. GRISEL, M. BLONDEL, P. PRETTENHOFER, R. WEISS, V. DUBOURG, J. VANDERPLAS, A. PASSOS, D. COURNAPEAU "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, 12 2825-2830 (2011).

7. The RELAP5 Development Team, "RELAP5/ MOD3 Code Manual, Volumes 1 and 2", NUREG/CR-5535, INEL-95/0174, Idaho National Engineering Laboratory, (1995).

8. D. GASTON, C. NEWMAN, G. HANSEN, D. LEBRUN-GRANDIE', "MOOSE: A parallel computational framework for coupled systems of nonlinear equations", *Nuclear Engineering and Design*, **239**, Issue 10, pp. 1768- 1778 (2009).

9. A. ALFONSI, C. RABITI, D. MANDELLI, J. COGLIATI, R. KINOSHITA, "Raven as a tool for dynamic probabilistic risk assessment: Software overview," *International Conference on Mathematics and Computational Methods Applied to Nuclear Science & Engineering M&C2013* (2013).

10. D. MANDELLI and C. SMITH, "Adaptive sampling using support vector machines," in *Proceeding of American Nuclear Society (ANS),* vol. 107, pp. 736-738, (2012).