Contents lists available at SciVerse ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

Scenario clustering and dynamic probabilistic risk assessment

Diego Mandelli [a,*], Alper Yilmaz [b], Tunc Aldemir [a], Kyle Metzroth [a], Richard Denning [a]

[a] *Nuclear Engineering Program, The Ohio State University, Columbus, OH, United States*

[b] *Photogrammetric Computer Vision Lab., The Ohio State University, Columbus, OH, United States*

ARTICLE INFO

Article history:
Received 2 February 2012
Received in revised form
3 January 2013
Accepted 11 February 2013
Available online 28 February 2013

Keywords:
Transient analysis
Scenario clustering
Dynamic PRA

ABSTRACT

A challenging aspect of dynamic methodologies for probabilistic risk assessment (PRA), such as the Dynamic Event Tree (DET) methodology, is the large number of scenarios generated for a single initiating event. Such large amounts of information can be difficult to organize for extracting useful information. Furthermore, it is not often sufficient to merely calculate a quantitative value for the risk and its associated uncertainties. The development of risk insights that can increase system safety and improve system performance requires the interpretation of scenario evolutions and the principal characteristics of the events that contribute to the risk. For a given scenario dataset, it can be useful to identify the scenarios that have similar behaviors (i.e., identify the most evident classes), and decide for each event sequence, to which class it belongs (i.e., classification). It is shown how it is possible to accomplish these two objectives using the Mean-Shift Methodology (MSM). The MSM is a kernel-based, non-parametric density estimation technique that is used to find the modes of an unknown data distribution. The algorithm developed finds the modes of the data distribution in the state space corresponding to regions with highest data density as well as grouping the scenarios generated into clusters based on scenario temporal similarities. The MSM is illustrated using the data generated by a DET algorithm for the analysis of a simple level/temperature controller and reactor vessel auxiliary cooling system.

Published by Elsevier Ltd.

1. Introduction

Dynamic methodologies for probabilistic risk assessment (PRA) are those that account for possible coupling between triggered or stochastic events through explicit consideration of the time element in the system evolution. They are usually needed when the system has more than one failure mode, control loops, and/or hardware/process/software/human interaction [1]. Dynamic methodologies are also capable of modeling the impacts of both the epistemic[1] and aleatory[2] uncertainties on the system figure-of-merit within a phenomenologically consistent framework.

Dynamic PRA methods include Dynamic Logical Analytical Methodology (DYLAM) [2], Dynamic Event Tree Analysis Method (DETAM) [3], ADS [4], ADAPT [5], Sequence Diagrams (ESDs) [6], Petri Nets [7], Dynamic Flowgraph Methodology (DFM) [8], Discrete Dynamic Event Trees (DDETs) [9], Markov/Cell-to-Cell Mapping Technique [10] and Monte Carlo Dynamic Event Tree (MCDET) [11]. The list is not exhaustive and only provides some samples of dynamic PRA methods. A more comprehensive discussion of dynamic methods is given in [1].

The DYLAM, DETAM, ADS and ADAPT are among methodologies that use dynamic event trees (DETs) to account for aleatory uncertainties. ADAPT can also account for epistemic uncertainties within the DET framework. A DET is an expansion on traditional static event trees (ETs), and seeks to incorporate timing and process relationships into the stochastic system model. Static ETs have a fixed and predetermined event sequence defined by the analyst, determined after a series of review processes and thermal hydraulic calculations. Fig. 1 shows a simplified ET for a large break loss of cooling accident (LOCA). In order to reach a safe state of the plant, the reactor protection system trips the reactor and performs the cooling of the reactor through the emergency cooling system (ECCS). A failure in any of these two systems will cause core damage (CD).

The DETs are generated by the direct coupling between the dynamic model of the plant and the stochastic behavior of system components (including software/firmware) and human actions. The branching conditions in a DET are generated by user specified rules, such as activation/non-activation upon demand of components, correct/faulty crew action depending on specific plant conditions or when state variables reach predefined setpoints during the simulation. Process and modeling uncertainties (which can affect the ordering of the events [5]) are also taken into account in terms of branching conditions. The dynamic model of the plant is built using system analysis codes (e.g., RELAP [12] or MELCOR [13])

* Corresponding author. Tel.: +1 614 579 5653.
E-mail addresses: diego.mandelli@inl.gov, mandelli.1@osu.edu (D. Mandelli), yilmaz.15@osu.edu (A. Yilmaz), aldemir.1@osu.edu (T. Aldemir), metzroth.1@osu.edu (K. Metzroth), denning.8@osu.edu (R. Denning).

[1] Uncertainties associated with lack of knowledge such as modeling uncertainties.

[2] Uncertainties due to stochastic variability of input and/or model parameters.

0951-8320/$ - see front matter Published by Elsevier Ltd.
http://dx.doi.org/10.1016/j.ress.2013.02.013

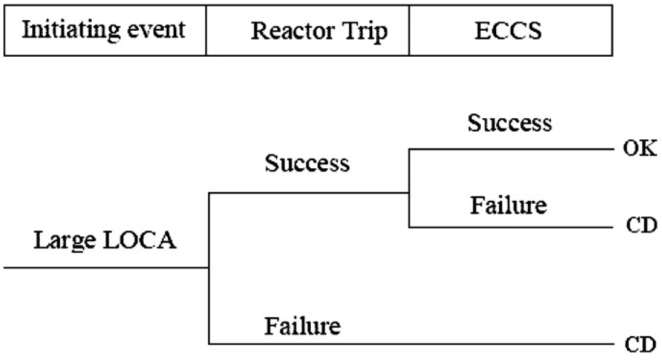

Fig. 1. Example of an event tree.

which evaluate its temporal behavior and determine the timing and nature of each branch. Subsequently, the use of DETs allows a more systematic and mechanistic search of the uncertainty space and also allows consideration of both the epistemic and aleatory uncertainties in a consistent phenomenological framework.

The major challenges in using DETs (as well as other dynamic methodologies) are the heavier computational and memory requirements compared to the classical ET analysis. This is due to the fact that each branch generated can contain time evolutions of a large number of variables (about 50,000 variables (or data channels) are present in MELCOR) and thus a much larger number of scenarios can be generated (on the order of several thousands) compared to the traditional ET/fault-tree (FT) approach. Such large amounts of information are very difficult to organize and interpret in regard to the main trends in scenario evolutions and the main risk contributors for each initiating event [14]. A solution to this problem is to partition the set of scenarios into groups, called clusters, and analyze each group individually rather than all the scenarios simultaneously. The partition is performed by identifying similarities among scenarios and grouping them according to a predefined similarity criteria. Once the partition is obtained, the user can analyze each group and identify differences among groups.

When dealing with nuclear transients, it is possible to analyze the set of scenarios in two modes:

- *End State Analysis* classifies the scenarios into clusters based on the end state of the scenarios.
- *Transient Analysis* classifies the scenarios into clusters based on time evolution of the scenarios.

While the first mode has been widely used in the classic fault tree/event tree analysis [15], the second one is starting to be considered in the recent years [16]. From a safety point of view, for example, it might be useful to group scenarios based on their temporal behavior and identify how sequence and timing of events affect the overall system dynamics other than focusing only on the end result of the simulation.

This paper presents a scenario clustering algorithm which can simplify the organization and analysis of the large dataset generated by a DET. By scenario clustering we mean two actions:

1. Identify the scenarios that have a similar behavior (i.e., identify the most evident clusters).
2. Associate each scenario with a unique cluster.

When clusters are determined, the user can then identify similarities among the scenarios in each cluster (e.g., timing and sequence of events) and compare them among different clusters. For example, clustering applied to dynamic PRA helps the user to understand how small changes in sequence/timing of events impact the overall system dynamics (see Section 4.3). A metric of success is, thus, the ability to determine a set of clusters that can help the user to identify such effects.

In Sections 2 and 3, the notions of clustering and classification are introduced along with the need and the approach to pre-processing of the raw data. In Section 4, we introduce the MSM using the dataset generated by DET for the level controller presented in [17] as an example (Section 4.2). In Section 4.3, we apply the Mean Shift Methodology (MSM) to the reactor vessel auxiliary cooling system (RVACS) of a conceptual design for a sodium-cooled fast reactor. Section 5 presents the conclusions of the study.

2. Clustering: an overview

Clustering is the process of organizing objects into groups whose members are in some way similar. A cluster is therefore a collection of objects which are similar to each other and are dissimilar to the objects belonging to the other clusters [18].

Fig. 2 shows an elementary example of partitional clustering [19] applied to a two-dimensional data. Here, we easily identify the 3 clusters into which the data can be divided. The similarity criterion is the distance measure: two or more objects belong to the same cluster if they are "close" according to a specified distance measure. The approach of using distance metrics to clustering is called distance-based clustering and will be used in this work by employing the Euclidean distance as a measure of the similarity between two D-dimensional data points $\vec{x_i}$ and $\vec{x_j}$

$$d(\vec{x_i},\vec{x_j}) = \left(\sum_{k=1}^{D} |x_{ik}-x_{jk}|^2\right)^{1/2}. \tag{1}$$

where data point $\vec{x_i}$ represents a scenario in the D-dimensional space of the data channels or variables of interest.

More formally, the concept of clustering [18] that we aim is to find a partition $\mathbf{C}=\{C_1,\ldots,C_l,\ldots,C_L\}$ of the set of I scenarios $\mathbf{X}=\{\vec{x_1},\ldots,\vec{x_i},\ldots,\vec{x_I}\}$ Each C_l $(l=1,\ldots,L)$ is called a cluster. The partition $\mathbf{C}$ of $\mathbf{X}$ is given as follows:

$$\begin{cases} C_l \neq \emptyset, l=1,\ldots,L \\ \bigcup_{l=1}^{L} C_l = \mathbf{X} \end{cases} \tag{2}$$

Clustering algorithms can be divided into two classes [18]:

- Hierarchical algorithms
- Partitional algorithms

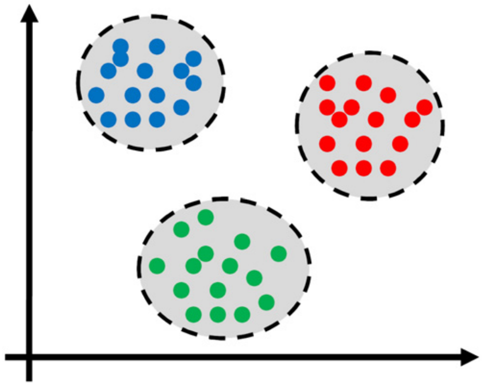

Fig. 2. Example of clustering process.

Given a set of data points $\overrightarrow{x_i}$, hierarchical algorithms build a hierarchical tree from the individual point (leaf) by progressively merging them into clusters until all points are inside a single cluster (root). Partitional clustering, on the other hand, seeks a single partition of the data sets instead of a nested sequence of partitions obtained by hierarchical methodologies. Under this category it is possible to classify methodologies under five main sub-categories: Squared Error (e.g., K-Means [20]), Fuzzy clustering (e.g., Fuzzy C-Means [21]), Mode Seeking (e.g., Mean-Shift [22]), Graph Theoretical [23] and Neural Network based [24].

We initially investigated hierarchical algorithms as well as partitional algorithms including [25]:

- Squared Error
- Fuzzy C-Means
- Mode-Seeking

Hierarchical algorithms [26] organize data into a hierarchical structure accordingly to a proximity matrix in which an entry (α,β) is some measure of the similarity (or distance) between the items to which row α and column β corresponds. Usually, the final result of these algorithms is a binary tree, also called dendrogram, in which the root of the tree represents the whole dataset and each leaf is a data point. Squared Error [26] algorithms assign each point to a cluster whose center (also called centroid) is nearest. The cluster center is the average of all the points in the cluster, that is, its coordinates are the arithmetic mean of each dimension independently over all the points in the same cluster. The most famous and used methodology is the K-Means algorithm [20], where the user specifies a priori the number of clusters to be determined. Fuzzy C-Means [21] clustering is very similar to K-Means but it assigns a degree of belonging coefficient for each point to each cluster, as in fuzzy logic, rather than assigning it to a single cluster. Mode-Seeking [27] is based on the assumption that the distribution of the points in the state space can be described through an unknown probability density function (pdf). The goal is to find the modes with highest probability, i.e., the regions in the state space with higher data densities. In this case, the number of clusters obtained is dependent on the local density functions that are used.

A particular mode seeking approach the MSM [22], which has been used for a number of applications in different fields. For our purpose, we identified the MSM as the most promising approach for the following reasons:

- Hierarchical, K-Means and Fuzzy C-Means algorithms are able to identify clusters of points having only spherical or ellipsoidal shape while MSM can identify clusters having any arbitrary geometry.
- Hierarchical, K-Means and Fuzzy C-Means algorithms have difficulty identifying outliers, i.e., clusters having a very small number of points in it. On the other hard, the MSM can easily identify scenarios that are considerably distant from the others.
- The level of discrimination among the clusters can be specified using the bandwidth parameter of the MSM. In that way, the appropriate number of clusters is determined by the algorithm itself instead of specifying the number of clusters that are going to be determined as required by K-Means and Fuzzy C-Means algorithms.

Once the clustering algorithm is chosen, the clustering process consists of the following steps [26]:

1. *Variable selection*: This first step is the identification of the variables that are considered useful for characterization of the data. For a transient in a nuclear plant, these variables may be both process variables such as temperature, pressure or level at specific points of the system and hardware/software/firmware states. The choice of the variables of interest therefore specifies the dimension of the state space.
2. *Transient representation format*: The chosen variables may be different in both nature (e.g., temperature and pressure) and scale (i.e., the range of these variables might differ in terms of order of magnitude). Hence, it may be necessary to perform a scaling operation on the chosen variables. In this work, we use the Principal Component Analysis (PCA) [28] as a tool to process the data before clustering is performed. The PCA also allows reducing the dimensionality of the state space for the clustering process.
3. *Clustering algorithm design*: In this phase, the distance measure (metric) along with other parameters of the algorithm is chosen.
4. *Clustering*: The clustering algorithm is applied to the dataset and clusters along with cluster centers are determined.
5. *Post-processing*: The cluster centers are converted back into the original format of the data.
6. *Interpretation of the results*: Given the cluster centers obtained in the previous step, the goal is to provide the user with meaningful insight of the original dataset. Further classification analysis may be required in order to gain such a meaningful insight (see Section 4.3). For example, clustering provides the user with a representative scenario in the cluster but understanding how small changes in sequence/timing of events impact the overall system dynamics may require further analysis as indicated in Section 1 (also see Section 4). Clustering analysis also allows identification of set of scenarios that have temporal profiles considerable different from the majority of the rest of scenarios (i.e., outlier identification).

3. Data representation

Since the temporal evolution of each scenario is typically described by the time evolution of all system state variables (e.g., pressure and temperature at a computational node), we chose to represent each scenario $\overrightarrow{x_i}$ $(i=1,\ldots,I)$ by M state variables x_{im} $(m=1,\ldots,M)$ plus time t (ranging from 0 to T) as the state vector:

$$\overrightarrow{x_i}=[x_{i1}(t_1),\ldots,x_{iM}(t_1),\ldots,x_{i1}(t_K),\ldots,x_{iM}(t_K)], \tag{3}$$

where $x_{im}(t_k)$ corresponds to the value of the variable x_m (e.g., temperature, pressure at a computational node) sampled at time t_k (e.g., $t_1=0$ and $t_K=T$) for scenario i. Note that the dimensionality of each scenario is $M\cdot K$ and can be extremely high for complex systems (i.e., large number of state variables and large number of samples). When dealing with transient codes such as RELAP [12] or MELCOR [13], the state space would theoretically include all the variables of all the discretization nodes (on the order of tens of thousands). Thus, a smaller set of variables of interests need to be chosen (e.g., by expert judgment).

At this point it should be noted that the dynamic behavior of variables that change rapidly in time can be captured by increasing the number of time points t_k or the sampling frequency. However, this may considerably increase the dimensionality of $\overrightarrow{x_i}$ and thus negatively affect the computational time in the clustering process.

The expression of $d(\overrightarrow{x_i},\overrightarrow{x_j})$ in (1) assumes that each data point (or scenario) $\overrightarrow{x_i}$ (see 3) is lying in a multidimensional space characterized by a set of orthogonal axes. However, due to the possible correlation of the chosen variables x_{im}, the assumption of orthogonality is not often justified. We addressed this problem by transforming the dataset using PCA [28]. PCA aims to transform the original set of possible oblique coordinate axes into a new set of orthogonal axes allowing the use of any type of metrics. This new set of orthogonal axes can be obtained by finding the eigenvectors of the covariance matrix of the dataset and the data points can be projected into the new coordinate system.

Another issue that arises when dealing with nuclear transients is the fact that the variables of interest may differ in nature (e.g., pressure, level or temperature) and, thus, have different ranges. Hence, each variable needs to be normalized before the clustering process. A common procedure [26] is to obtain the normalized variable $\overline{x_{im}}$ by scaling x_{im} into the [0,1] interval as follows:

$$\overline{x_{im}} = \frac{x_{im} - \min(x_{im})}{\max(x_{im}) - \min(x_{im})}. \tag{4}$$

4. Mean-Shift Methodology

The MSM [22] is a non-parametric iterative procedure that assigns each data point to one cluster center through a set of local averaging operations. The local averaging operations provide the empirical cluster centers in the locality and define the vector which denotes the direction of increase for the underlying unknown density function.

The basic idea is to treat each data point, or scenario, $\vec{x_i}$ $(i = 1,\ldots,I)$ of the dataset as an empirical probability distribution function, or kernel $K(\vec{x}) : \mathbb{R}^{M \cdot K} \to \mathbb{R}$ (see Fig. 3 where each of the 7 data points represented by circles is associated with a probability density function). This kernel density resides in a multidimensional space where regions with high data density (i.e., modes) correspond to local maxima of the density estimate $f_I(\vec{x})$ [29] (represented by the trimodal upper curve in Fig. 3) defined by

$$f_I(\vec{x}) = \frac{1}{Ih^d} \sum_{i=1}^{I} K\left(\frac{\vec{x} - \vec{x_i}}{h}\right), \tag{5}$$

where $\vec{x} \in \mathbb{R}^{M \cdot K}$ and h is often referred as the bandwidth of the kernel.

The kernel serves as a weighting function [29] associated with each data point and can be expressed as

$$K(\vec{x}) = c_k k(\| \vec{x}^2 \|), \tag{6}$$

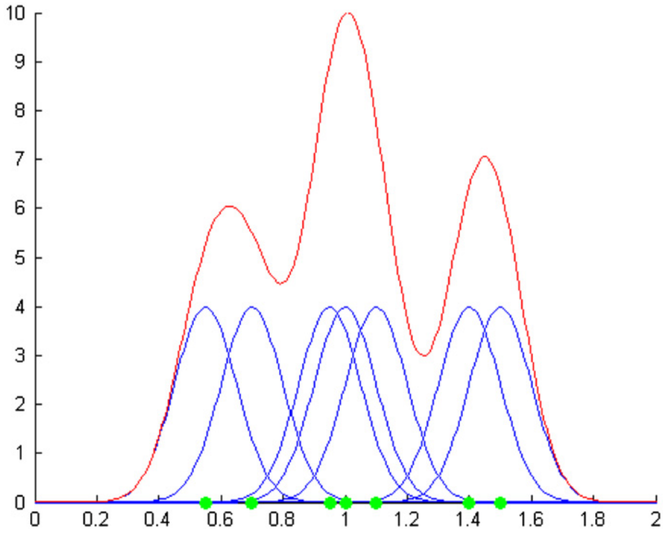

Fig. 3. An example density function.

where $k(x) : [0,\infty] \to \mathbb{R}$ is referred as the *profile* and c_k is a normalization constant. The profile satisfies the following properties:

1. $k(x)$ is non-negative,
2. $k(x)$ is non-increasing (i.e., $k(a) \geq k(b)$ if $a < b$),
3. $k(x)$ is piecewise continuous and $\int_0^\infty k(x)\,dx < \infty$.

In order to estimate the data points with highest probability from an initial estimate (i.e., the modes of $f_I(\vec{x})$), consider the situation $\nabla_x f_I(\vec{x}) = 0$ [22], where

$$\nabla_x f_I(\vec{x}) = \frac{2c_k}{Ih^{d+2}} \sum_{i=1}^{I} (\vec{x} - \vec{x_i}) k'\left(\left\| \frac{\vec{x} - \vec{x_i}}{h} \right\|^2 \right)$$

$$= \frac{2c_k}{Ih^{d+2}} \underbrace{\left(\sum_{i=1}^{I} g\left(\left\| \frac{\vec{x} - \vec{x_i}}{h} \right\|^2 \right) \right)}_{A} \underbrace{\left(\frac{\sum_{i=1}^{I} \vec{x} g\left(\left\| \frac{\vec{x} - \vec{x_i}}{h} \right\|^2 \right)}{\sum_{i=1}^{I} g\left(\left\| \frac{\vec{x} - \vec{x_i}}{h} \right\|^2 \right)} - \vec{x} \right)}_{B} \tag{7}$$

points in the direction of the increase in kernel density estimate. The kernel $K(\vec{x})$ is also referred to as the shadow of $G(\vec{x}) = c_g g(\| \vec{x} \|^2)$ [30] where c_g, similar to c_k, is a normalization constant and $g(x)$ is the derivative of $k(x)$ with respect to x, i.e., $g(x) = k'(x)$. In (7) the first term denoted as A is a scalar proportional to the density estimate computed with the kernel $G(\vec{x})$ and does not provide information regarding the mode position. Unlike A, the vector quantity B, which is the second term in (7), is the difference between the weighted mean:

$$m(\vec{x}) = \frac{\sum_{i=1}^{I} \vec{x} g\left(\left\| \frac{\vec{x} - \vec{x_i}}{h} \right\|^2 \right)}{\sum_{i=1}^{I} g\left(\left\| \frac{\vec{x} - \vec{x_i}}{h} \right\|^2 \right)}, \tag{8}$$

and the initial estimate $\vec{x}$. This term points in the direction of local increase in density using kernel $G(\vec{x})$, hence provides a means to find the mode of the density. In other words, each scenario or data point $\vec{x_i}$ is "shifted" toward the mean defined by Eq. (8) which is the reason why the methodology is called "mean-shift". Note that all data points used to compute a particular mode are considered to reside in the same cluster.

Since each data point $\vec{x_i}$ (or scenario) is considered as an empirical probability distribution function, the probability associated with each scenario as obtained from the DET can be accounted for in the clustering analysis.

The choice of the value of bandwidth greatly affects the number of cluster that can be obtained. For the 7 data points shown in Fig. 3, a small value of bandwidth would generate a $f_I(\vec{x})$ with 7 local maxima as shown in Fig. 3 (i.e., a number of local maxima equal to the number of data points). On the other hand, a large value of bandwidth would generate a $f_I(\vec{x})$ with only one local maxima (i.e., one cluster).

Section 4.1 presents the algorithm that we have developed to implement the MSM. Sections 4.2 and 4.3 show, respectively, an application of the MSM applied to the dataset generated by a DET for a simple level controller [10] and to the dataset generated for a more complex system such as the behavior of conceptual design for a sodium-cooled fast reactor during an aircraft crash scenario.

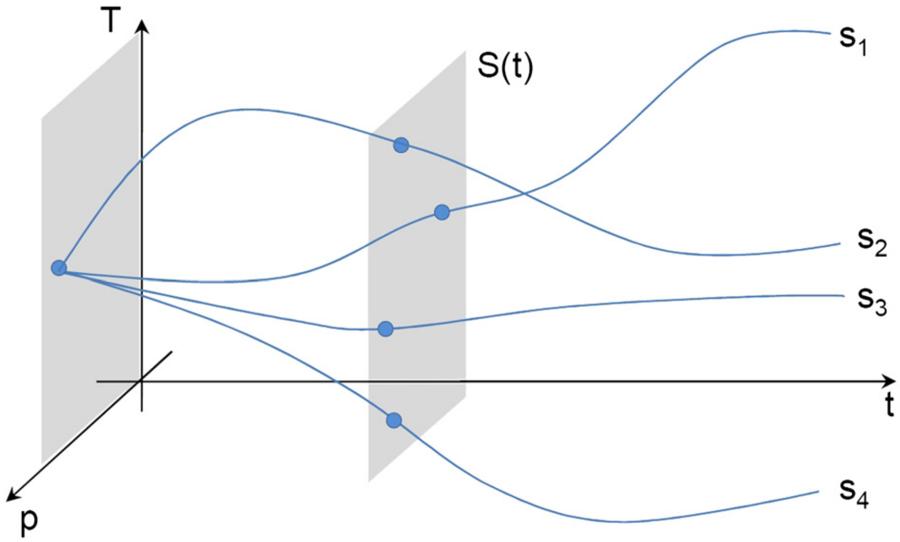

Fig. 4. Representation of example scenarios generated by a DET in a three-dimensional space.

4.1. The algorithm

In order to illustrate how the algorithm works, let us consider a system whose dynamics can be described by 3 variables consisting of time t, pressure p and temperature T as an example. Let us assume that a DET analysis quantifying the impact of the uncertainties in the system yields the trajectories shown in Fig. 4. The $S(t)$ plane in Fig. 4 represents the state of these scenarios at time t.

Note that the new dataset on the $S(t)$ plane consists of data points distributed in a two-dimensional space $\mathbb{R}^2$. Starting from an arbitrary data point (e.g., point S_A in Fig. 5), the algorithm associates a circle (or a ball, in general, depending on the number of dimensions of the state space) centered at that data point.

The radius of this circle (ball) is equal to the bandwidth h of the chosen kernel. The objective is to consider the weighted average of the data points residing in the circle (ball) which corresponds to their center of mass (point $m(S_A)$ in Fig. 5) where

$$m(S_A) = \frac{\sum_{i=1}^{I} \vec{x_i}\, g\left(\left\|\frac{S_A - \vec{x_i}}{h}\right\|^2\right)}{\sum_{i=1}^{I} g\left(\left\|\frac{S_A - \vec{x_i}}{h}\right\|^2\right)}. \tag{9}$$

The weighted average or the center of mass $m(s_A)$ is used as the new position for S_A for the next iteration, such that the density in the new center of mass is always higher than its previous position. Convergence is reached when the distance between the new center of mass and the old one is below a fixed threshold[3] (point S_C in Fig. 5). Upon reaching the stopping condition, point S_C is considered the center of a cluster and the original point S_A is uniquely associated to the cluster centered by point S_C.

This procedure is performed for each data point $\vec{x_i}$. When the data set is characterized by a single cluster (see Fig. 5), the cluster center S_C is associated with all the data points. However, when the data set is characterized by multiple regions having high data densities, the MSM determines multiple cluster centers as shown in Fig. 6. Each data point is associated with a specific cluster according to its Mean-Shift path.

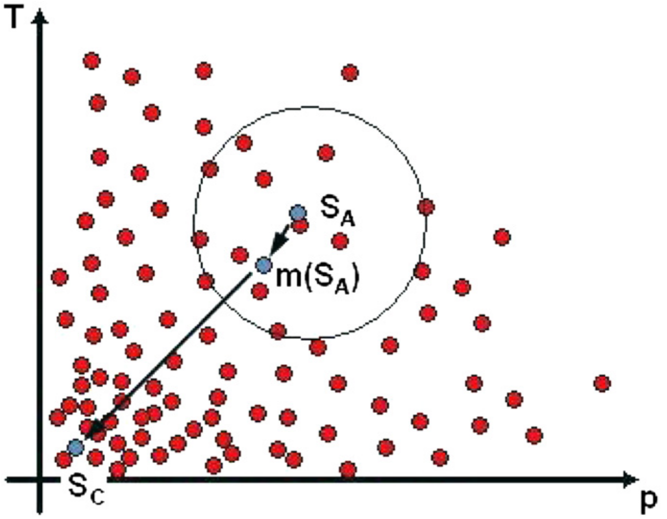

Fig. 5. Determination of a cluster center in a two-dimensional space using the Mean-Shift algorithm on the $S(t)$ plane.

During this process, several derivative kernels $G(\vec{x})$ can be used as indicated in [27]. Given data with unknown distribution, Epanechnikov and Gaussian kernels provide the optimal mean integrated squared error. Among the two, coding of the Gaussian kernel eliminates the requirement of choosing a user-defined discriminatory threshold. Hence, we chose the Gaussian kernel in our implementation (see Fig. 7):

$$G(\vec{x}) = e^{-\|\vec{x}\|^2/h^2}. \tag{10}$$

When this iterative procedure is repeated for all the points in the dataset, it is possible to obtain the center of all the clusters and the list of all the points that belong to that specific cluster, as well as, the cluster to which each point belongs (as mentioned earlier, each point belongs to one cluster only).

Fig. 8 shows the flow diagram of the proposed approach. Starting from a dataset generated by the DET tool, the user selects variables of interest (Feature Selection in Fig. 8) based on their physical relevance to the process under consideration or possibly as a result of dimensionality reduction process (e.g., based on PCA). Following

[3] Usually the threshold is a fixed value chosen to be a small fraction of h (typically $h/100$ or $h/1000$)

the choice of bandwidth, type of kernel, and the distance metric (possibly random initially), the clustering is iteratively performed (Classification in Fig. 8). Sensitivity of the grouping of scenarios can be then examined for different types of kernels, values of bandwidth and metrics (Post-Processing in Fig. 8).

Besides the choice of the distance, the geometry of the clusters plays a vital role in the point-to-cluster decision process. Fig. 9 illustrates the results of MSM when it is applied to a set of points

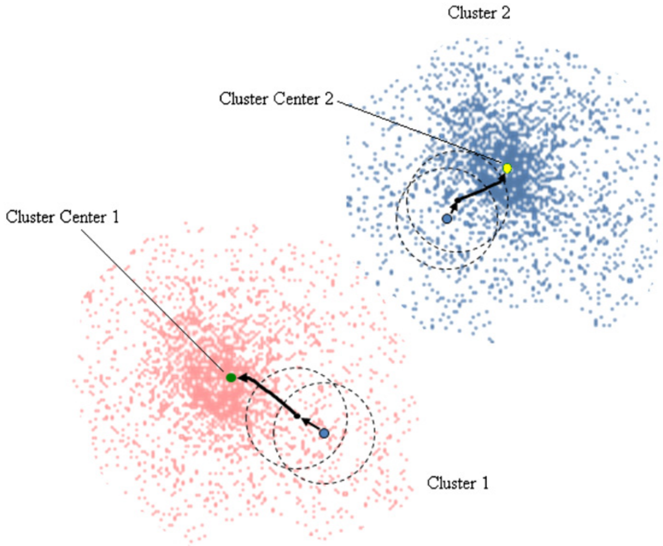

Fig. 6. Determination of multiple cluster centers in a two-dimensional space using the Mean-Shift algorithm.

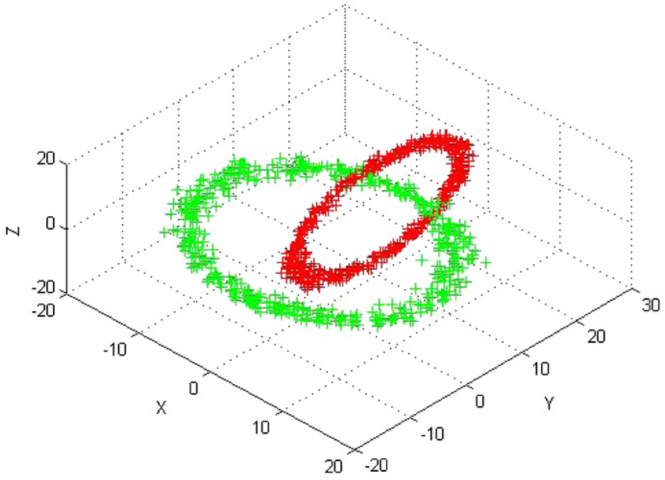

Fig. 9. Mean-Shift analysis applied to a set of data distributed over two rings. Different colors denote different clusters.

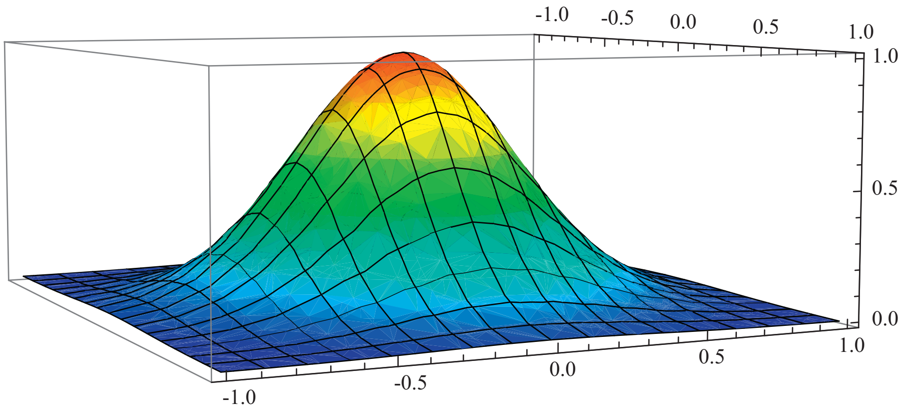

Fig. 7. Graphical representation of the two-dimensional Gaussian kernel.

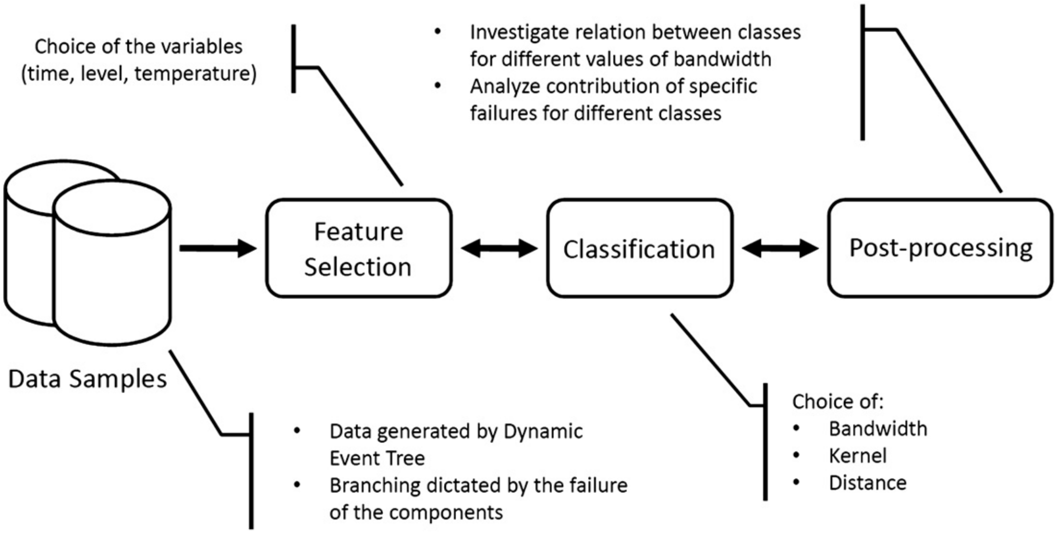
Choice of the variables
(time, level, temperature)
• Investigate relation between classes for different values of bandwidth
• Analyze contribution of specific failures for different classes
Data Samples
Feature Selection
Classification
Post-processing
• Data generated by Dynamic Event Tree
• Branching dictated by the failure of the components
Choice of:
• Bandwidth
• Kernel
• Distance

Fig. 8. General flow of the Mean-Shift Methodology algorithm.

distributed normally along two rings. In the region where the two rings are very close, points that belong to different clusters have different memberships despite being close to each other compared to the other points belonging to the same cluster. As shown in Fig. 9, the data lying on two rings have been clustered using the MSM algorithm described above into two clusters denoted by red and green colors even some points lying on different clusters are very close to each other. Hence, methodologies such as K-Means [20] or Fuzzy C-Means [21] that consider distance as the only criteria to establish cluster memberships are not able to cluster the datasets pictured in Fig. 9 correctly.

4.2. Analysis of the water level controller

As a simple system to illustrate the deployment of the algorithm shown in Fig. 8, we use the heated tank example presented in [31] and shown in Fig. 10. In this system, the liquid level L is actively controlled through the actuation of three components: two inlet pumps and one outlet valve, hereafter called Units 1, 2 and 3, respectively. Each unit is a multi-state component operating either correctly ON or OFF, stuck ON or stuck OFF. At time $t=0$, the system is assumed to be in its nominal state (ON,OFF,ON), with nominal values of $T=30.93\ ^\circ\mathrm{C}$ of the liquid temperature and $L=7$ m. The temperature T is assumed to directly affect the failure rates λ of the components as given by [31]

$$\lambda(T)=\frac{b_1 e^{-b_c(T-20)}+b_2 e^{-b_d(T-20)}}{b_1+b_2}\overline{\lambda(i)},(\mathrm{i}-1,2,3) \tag{11}$$

where $b_1=3.0295$, $b_2=0.7578$, $b_c=0.05756$, $b_d=\ 0.2301$. Furthermore: $\overline{\lambda(1)}=2.2831\times10^{-3}\ \mathrm{h}^{-1}$, $\overline{\lambda(2)}=2.857\times\ 10^{-3}\ \mathrm{h}^{-1}$, $\overline{\lambda(3)}=1.5625\times10^{-3}\ \mathrm{h}^{-1}$. A power source heats up the fluid to keep it at the nominal temperature. The liquid level is kept between 6 and 8 m through the control laws reported in [10] (also see Table 1). Two possible Top Events that need to be considered are Low Level ($L<4$ m) and High Level ($L>10$ m).

For testing the algorithm described in Section 4.1, the variables that describe the evolution of each scenario were chosen to be the temperature T, level L and time t. Thus the state space is three-dimensional.

In the DET analysis, the branching is dictated by the failure of the three active components (i.e., Units 1, 2 and 3 in Fig. 10). We applied the MSM to the set of transients generated in [32]. Each transient contains information about the time evolution of temperature and level and the status of the three controllers as well. We converted each transient into a vector $\overrightarrow{x_i}$ as shown in (3) with

- $M=2$ (i.e., 2 state variables: temperature and level),
- time is ranging from 0 to 5 h (i.e., $T=5$), and,
- level, temperature sampled every hour (i.e., K=5): $t_1=0$, $t_2=1$, $t_3=2$, $t_4=3$, $t_5=4$ and $t_6=5$.

Thus, each scenario is characterized by a vector having dimensionality equal to 10 ($M\cdot K=10$).

We performed the clustering for the scenarios leading to Low Level and High Level separately, with 105 scenarios leading to High Level and 23 leading to Low Level. Figs. 11–13 show the cluster centers (i.e., the representative scenarios) for the data generated by the DET for these Top Events. As can be seen from the figures, the clustering process becomes more refined by decreasing the value of the bandwidth and the number of clusters obtained increases. Asymptotically, the number of clusters equals the number of scenarios with decreasing bandwidth. Also note the different behavior in terms of cluster centers obtained for the two Top Events: the number of clusters obtained for High Level increases as h increases while, for Low Level, same clusters have been obtained for $h=9$ and $h=11$. This indicates that different values of h may be needed when clustering is performed for scenarios leading to different Top Events. For this data set, however, comparison of Figs. 11–13 indicates that while going

Table 1
Water level controller control laws.

Case	Controller 1	Controller 2	Controller 3
$6\ \mathrm{m}\le L\le 8\ \mathrm{m}$	ON	OFF	ON
$L\le 6$ m	ON	ON	OFF
$L\ge 8$ m	OFF	OFF	ON

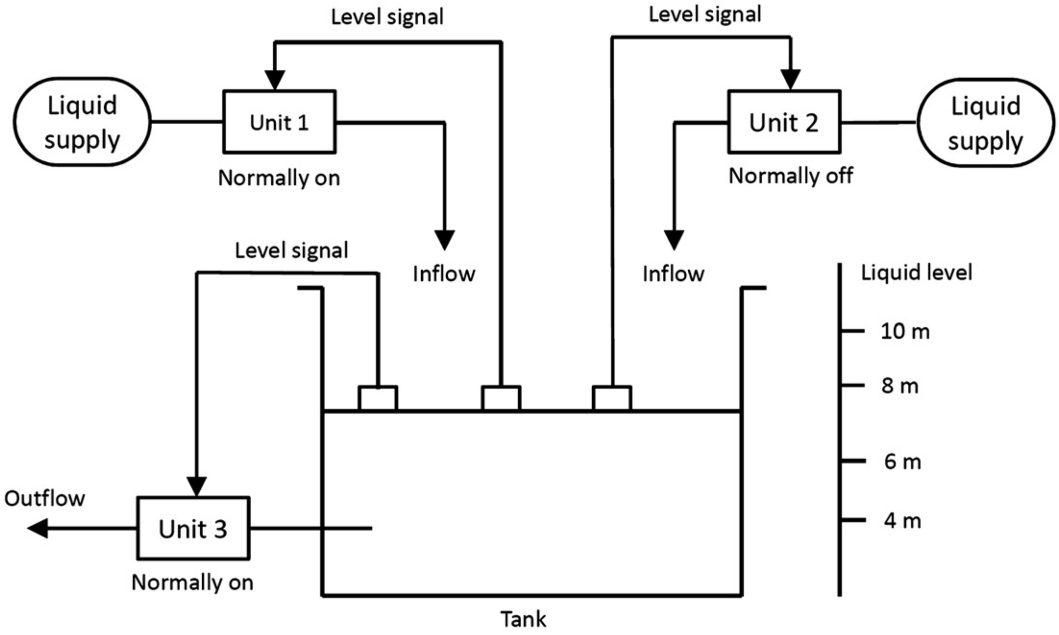

Fig. 10. Schema of the water heated tank.

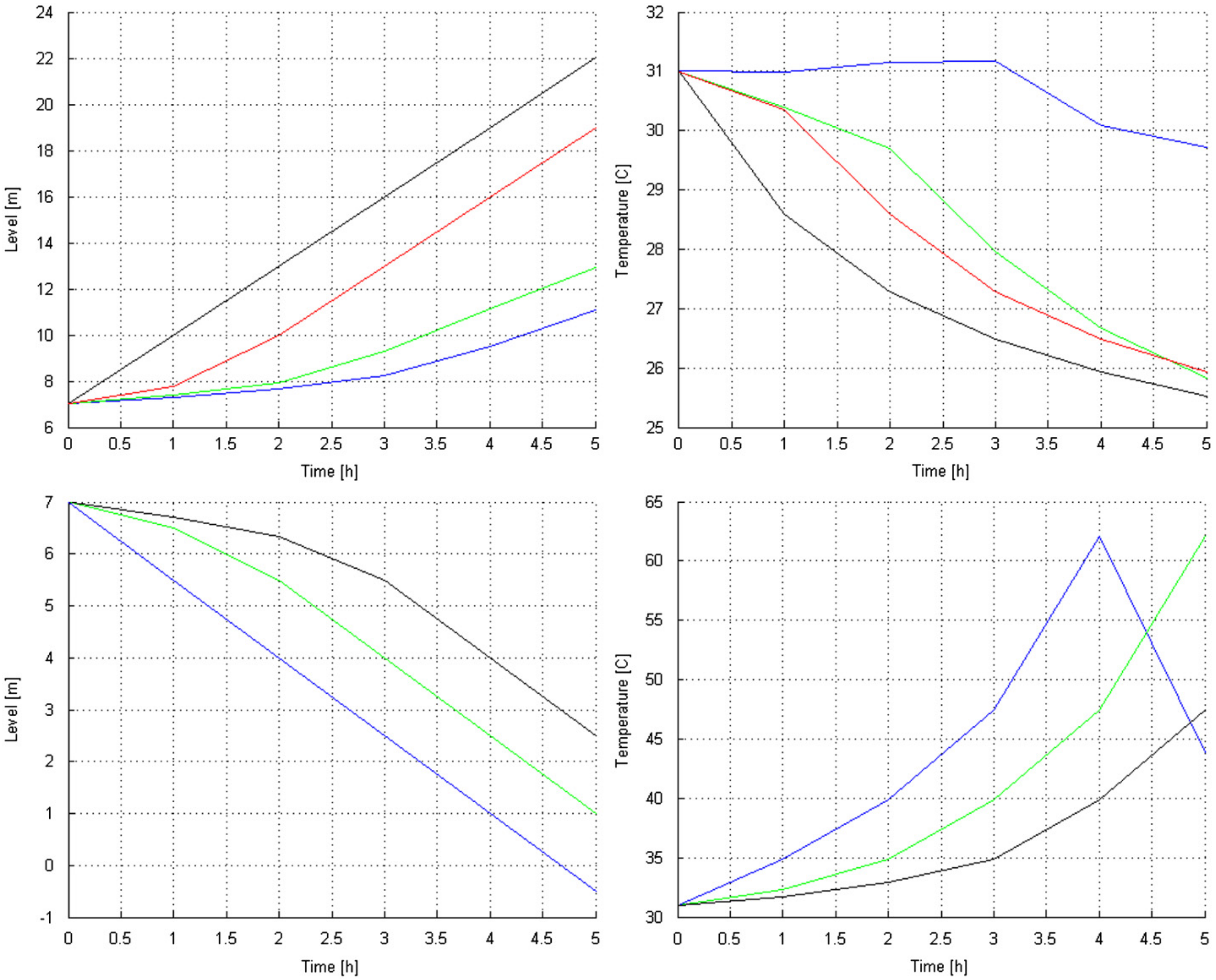

Fig. 11. Cluster centers for high level (top) and low level (bottom): $h=11$.

from $h=11$ to $h=9$ introduces a new pattern in the temperature profile for High Level (compare Figs. 11 and 12), there are no fundamental differences in the profiles in going from $h=9$ to $h=7$ (compare Figs. 12 and 13). In that respect, $h=9$ provides the best bandwidth for the date set under consideration. In should be also indicated that there is always some subjectivity on the choice of the bandwidth, depending on the objective of the clustering process and the granularity desired.

4.3. Analysis of an aircraft crash scenario

A more complex system used for the illustration of the proposed clustering methodology is the analysis of recovery from an aircraft crash into an RVACS of a conceptual design for a sodium-cooled fast reactor as schematically shown in Fig. 14. The RVACS is a passive decay-heat removal system that removes heat by natural circulation of air in the gap between the vessel and a duct surrounding the vessel. RVACS has been considered for use in some conceptual design studies of advanced reactors. With this system, the reactor decay heat is released to the atmosphere through four towers or stacks.

The Analysis of Dynamic Accident Progression Trees (ADAPT) tool [5] was used in this study as the DET generator tool while the system dynamics was modeled using RELAP5 [12]. The RVACS RELAP5 model is shown in Fig. 15 [33]. The analysis was performed on a three-node cluster. Run time was 3.19 days and the output data size was 40 GB.

At time zero with the plant operating at 100% power, an aircraft crashes into the plant. Three of the four towers are assumed to be destroyed, producing debris that blocks the air passages (hence, impeding the possibility to remove the decay heat). The reactor trips, off-site power is lost, the pump trips and coasts down. A recovery crew and heavy equipment are used to remove the debris. Fig. 16 illustrates the strategy that is followed by the crew in reestablishing the capability of the RVACS to remove the decay heat. Crew arrival and tower recovery times are stochastic variables and can have any value within the specified bands. Several assumptions have been made for the purpose of the analysis:

- A tower is assumed to have no heat removal capacity until the rubble has been removed. At that point it is assumed to regain full capacity;
- There is a 1 h dead period (see the red bars in Fig. 16) following the crash in which a fire is being extinguished;

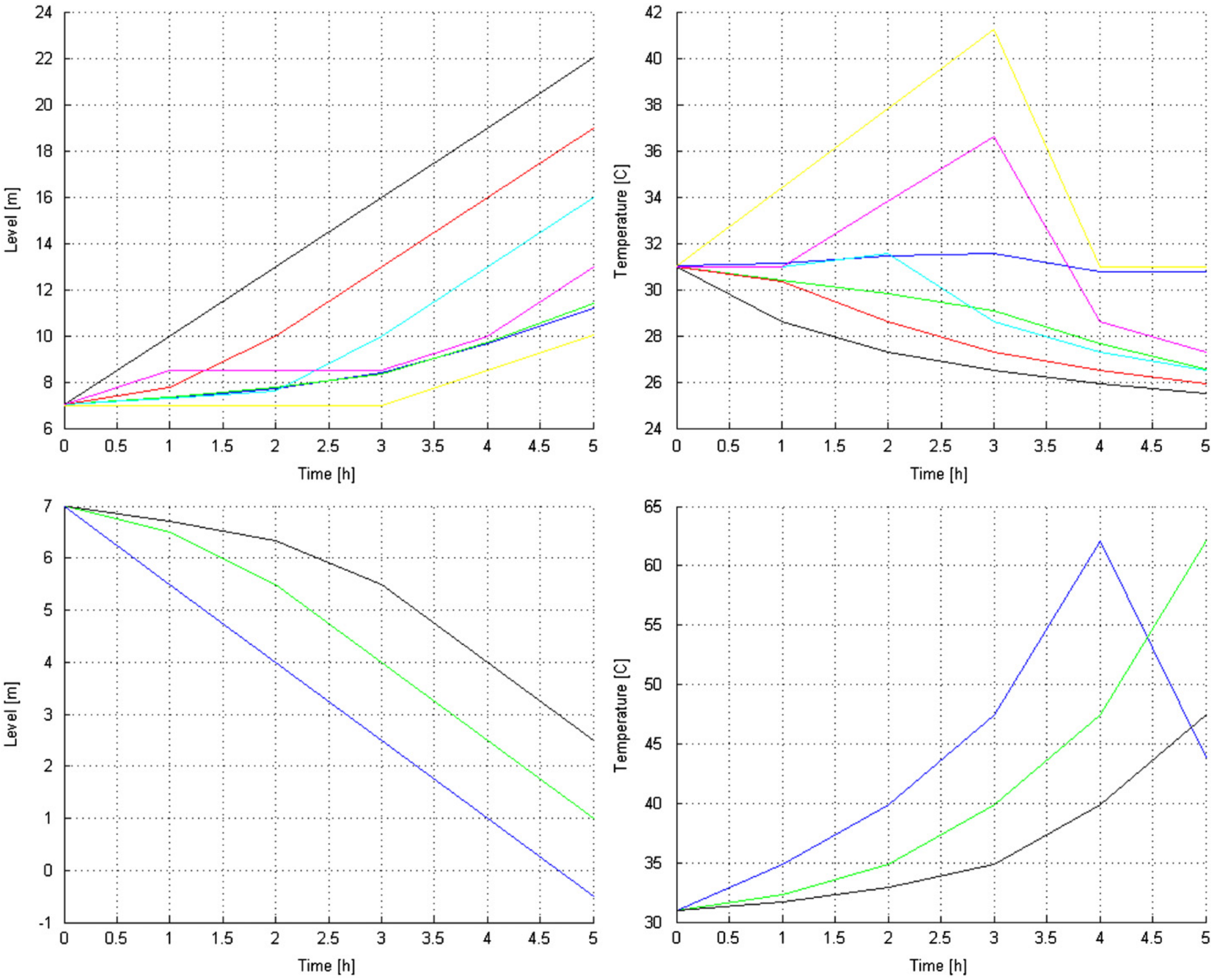

Fig. 12. Cluster centers for high level (top) and low level (bottom): $h=9$.

- There is a uniform probability of work being initiated between one and 9 h after the crash;
- The workers remove debris from one tower at a time;
- After work begins on a tower there is a minimum time of 2 h to recover the tower;
- There is a uniform distribution of recovery between two and 10 h. The team then moves on to the next tower;
- The recovery time of each tower is assumed to be independent of the other towers.

While the numerical values chosen do not reflect actual data, effort has been made to inject realism into the analysis by the use of dead time and minimum and maximum times.

As indicated above, the uncertainties in crew arrival time and tower recovery have been modeled by assigning to each one a uniform probability distribution function [33]. Example branching times were chosen for ADAPT to correspond to the probabilities 0.001, 0.25, 0.5, 0.75, and 1.0 on the corresponding cumulative distribution function. When one of these branching times is reached in the RELAP simulation, ADAPT generates a bifurcation in the system evolution occurs in which one branch represents the realization of the specific event (e.g., crew arrival) and the other branch represents non-realization of the specific event. The latter branch then continues until the time corresponding to the next branching point is reached on the cumulative distribution function. This type of branching scheme used in ADAPT is explained in more detail in [5].

Only one Top Event has been considered: temperature T of the core reaches the limit of 1000 K, associated with clad failure by eutectic formation. Fig. 17 shows the temporal behavior of the temperature of the core for all the 610 scenarios generated by ADAPT. Mission time for this system analysis has been fixed to 2×10^5 s.[4]

Each transient includes information about:

- Time profile of core temperature,
- Crew arrival time,
- Recovery time of tower i ($i=1, 2, 3$).

We performed the clustering analysis of the dataset pictured in Fig. 17 using the values of the core temperature sampled at

[4] In the original dataset, scenarios that reach a core temperature of 1000 K are stopped even though they did not reach 2×10^5 s. Consequently, the time length of scenarios may change depending if they have reached 1000 K or not. For those scenarios that reach 1000 K before 2×10^5 s, it was decided to extend in time these scenario up to 2×10^5 s with the last value simulated.

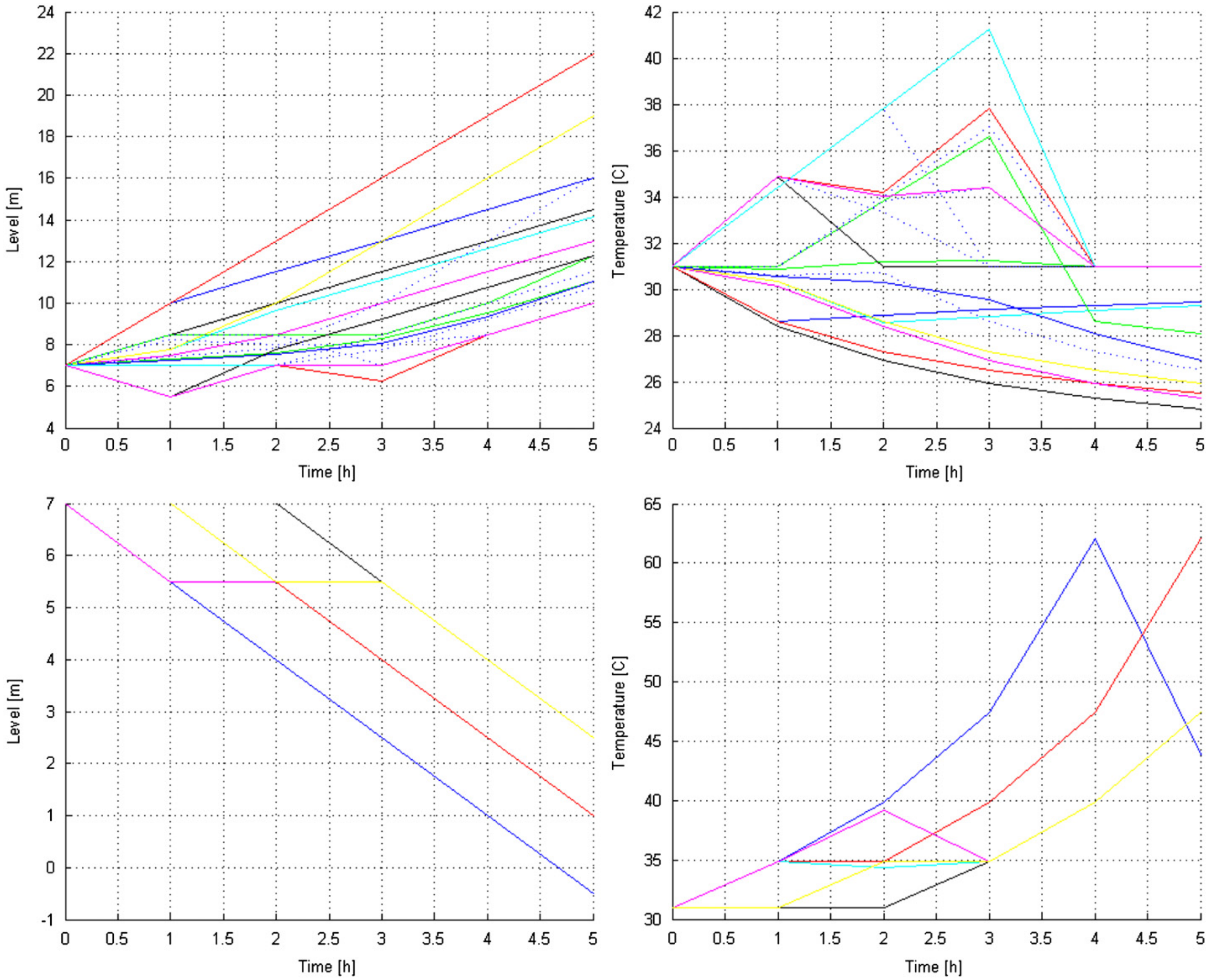

Fig. 13. Cluster centers for high level (top) and low level (bottom): $h = 7$.

specific time instants. Each scenario was represented by $K = 68$ time points sampled uniformly over a time horizon of $T = 2 \times 10^5$ s.

Table 2 shows the number of clusters obtained for different values of bandwidth h. For very large values of h (e.g., $h = 4$) the algorithm determines a single cluster which contains all the 610 scenarios. On the other hand, for very small values of h (e.g., $h = 0.001$) the algorithm determines 610 clusters; each of them contains a single scenario. For both of these cases no additional information can be extracted from the clusters since the clusters reflect exactly the original dataset.

As indicated earlier, the choice of the optimal value of bandwidth h is case dependent. For this data set, h is chosen such that each cluster can identify specific safety insights such as correlations between timing of events and temporal profiles of the state variables. For the RVACS data set, these correlations emerged for a value of bandwidth $h = 1.5$ as described below.

Fig. 18 shows the cluster centers obtained for $h = 1.5$. Fig. 18 also shows, for each of the 8 cluster centers, the cluster probability and the fraction of scenarios that belong to that cluster. Cluster probability is determined by summing all the probabilities of the scenarios contained in that cluster. Fig. 19 shows the cluster centers (i.e., the representative scenarios) in black lines and the scenarios belonging to that cluster in red lines.

At this point, it is possible to analyze the properties of the clusters individually instead of the full dataset. In this respect, a second analysis is performed for each of the eight obtained clusters by evaluating the distribution of the crew arrival time towers 1-3 as function of time for the scenarios belonging to each cluster. Fig. 20 shows the distribution of the crew arrival time and tower recovery time for all the scenarios belonging to each of the 8 clusters. Note that the vertical scales are not similar in Fig. 19 due to the difference of duration of the scenarios in different clusters.

Figs. 19 and 20 indicate the following:

- Scenarios contained in Clusters 1 and 4 are characterized by very early crew arrival (bars in Fig. 20 located at 2.5×10^4 s) and a rapid sequence of towers recovery which allow to cool the core rapidly (all towers are recovered before 8.5×10^4 s for all scenarios contained). As shown in Fig. 19, the scenarios included in both clusters lead to adequate core cooling (i.e., the maximum core temperature do not reach 1000 K) and, as expected, a rapid recovery of the towers is sufficient condition for the safety of the plant.

- Scenarios contained in Cluster 2 are characterized as well by an early crew arrival (bar in Fig. 20 located at 2.5 × 10^4 s) and a rapid recovery of the first tower (blue bars in Fig. 20 located between 3.5 × 10^4 s and 4.5 × 10^4 s). However, the recovery of the remaining two towers is not as rapid (green and purple bars in Fig. 20). Since this cluster contains scenarios that lead to both adequate core cooling and core damage (see Fig. 19 where some scenarios reach the limit of 1000 K and others are below it), it is possible to conclude that early crew arrival and recovery of the first tower is not sufficient condition to reach the safe state of the plant.
- Scenarios contained in Clusters 3 and 5 lead to both sufficient core cooling and core damage (see Fig. 19 where some scenarios reach the limit of 1000 K and so are below it). This is due to the fact that crew arrives on the field considerably late (bars in Fig. 20 located between 3.0 × 10^4 s and 4.5 × 10^4 s) followed by a rapid recovery of the three towers. However, the rapid recovery is not sufficient to avoid core damage.
- Cluster 6 is composed exclusively of scenarios that lead to core damage (see Fig. 19 where all scenarios reach the limit of 1000 K). Moreover, it also contains all the scenarios characterized

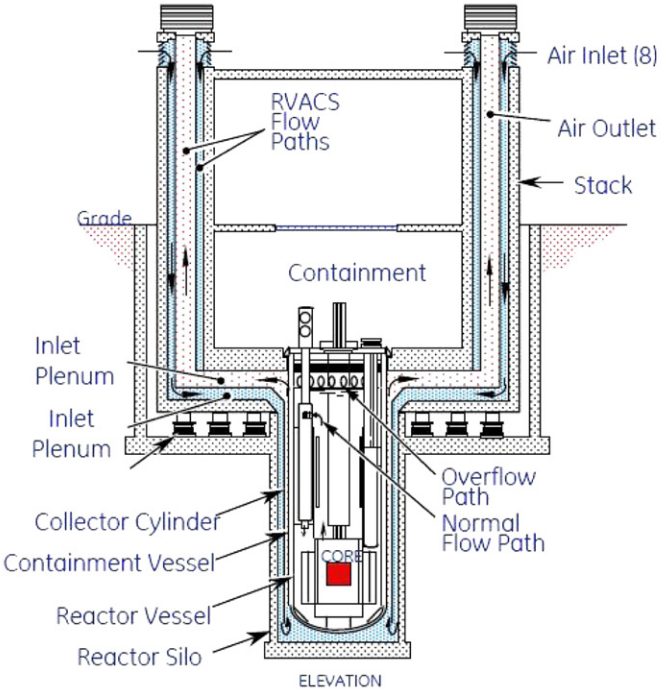

Fig. 14. RVACS system.

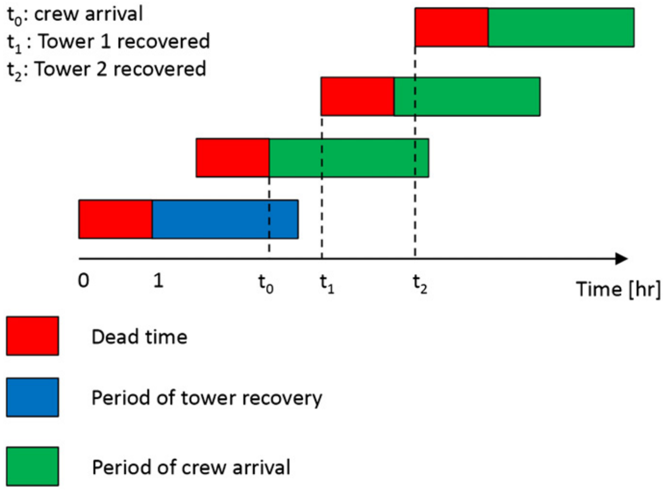

Fig. 16. Crew recovery strategy for the aircraft crash scenario.

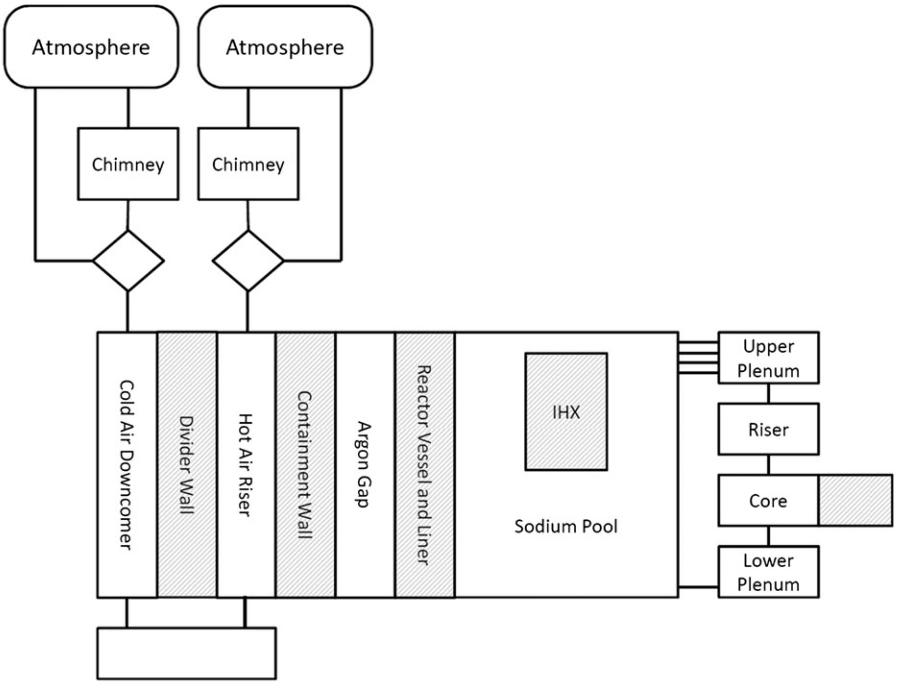

Fig. 15. RELAP RVACS model including the primary system of the example sodium-cooled fast reactor.

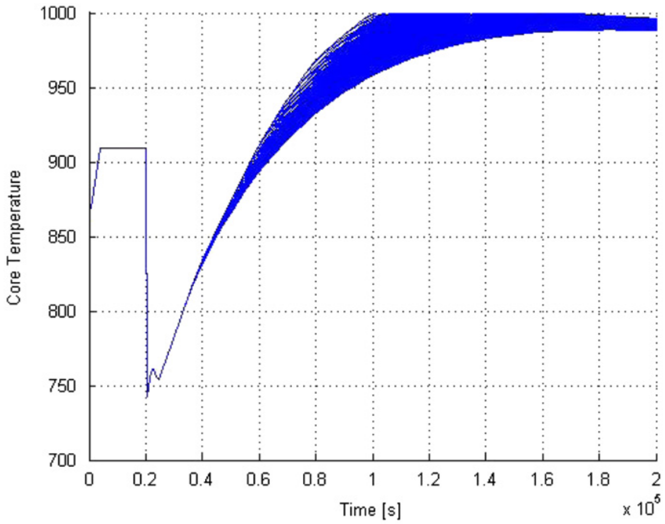

Fig. 17. Graphical representation of the scenarios generated by ADAPT for the aircraft crash scenario.

Table 2
Number of clusters obtained for different values of bandwidth *h*.

h	Number of clusters
4.0	1
3.0	2
2.0	4
1.5	8
1.0	22
0.5	96
0.1	300
0.01	308
0.001	610

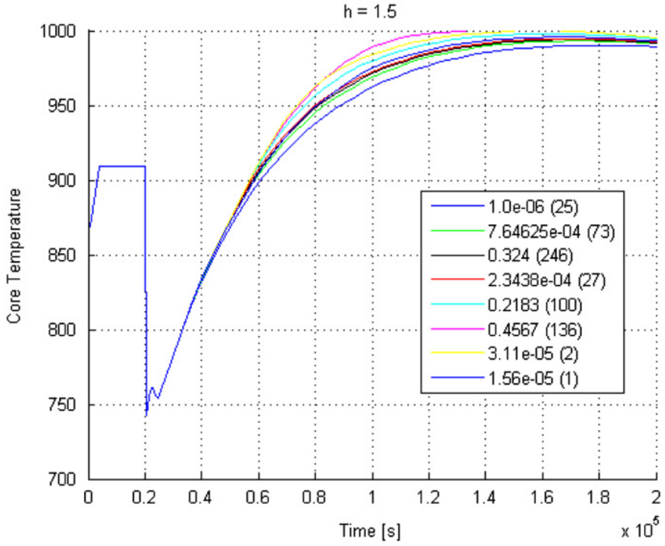

Fig. 18. Cluster centers for the RVACS system for $h=1.5$. The numbers in the legend indicate the cluster probability and, in parenthesis, the number of scenarios that fall in each cluster.

by the non-recovery of the third tower (bar located at 0 s in Fig. 20). As observed for Clusters 3 and 5, crew arrives very late on site (bars in Fig. 20 located between 3.0×10^4 s and 4.5×10^4 s) and hence the tower recovery strategy is not sufficient for adequate core cooling.

- Clusters 7 and 8 contain a very small number of scenarios (i.e., 2 and 1 scenario respectively). Scenarios included in Cluster 7 are characterized by a late crew arrival (see crew arrival time bars in Fig. 20), a late recovery of the first tower and a fast recovery of the following towers. This action allows to sufficiently cool the core but temperature profile for these scenarios is very close to the limit temperature (i.e., max temperature of the core for the scenarios in Fig. 19 is 999 K). However, the scenario included in Cluster 8 is characterized by an early crew arrival and recovery of the first tower (see crew arrival time and tower 1 bars in Fig. 20). The recovery of the remaining towers is not particularly fast but it avoids exceedance of the limit temperature.

The mixed nature of Clusters 2, 3 and 5 (i.e., some scenarios leading to core damage and some to adequate core cooling) imply the following which would have been difficult to observe from the full data set:

- Early crew arrival and early recovery of the first tower is not sufficient condition to adequate core cooling; a late recovery of the remaining two towers leads to core damage.
- Late crew arrival time does not necessarily lead to core damage. Fast recovery of the 3 towers can be sufficient to provide adequate core cooling.

No model uncertainties were considered in this analysis. It should be indicated that scenario evolutions might have been different if model uncertainties were considered [5] and might have affected the consequences of some of the near miss scenarios. It should be also indicated that performing a similar analysis using the traditional ET/FT would have been extremely challenging since some of the crew actions depend on the event history (e.g., repair of the second tower cannot start before the repair of the first tower is completed). Capability to model such history dependencies was one of the needs that led to the development of dynamic PRA methodologies.

5. Conclusion

In this paper, we proposed a methodology based on PCA and MSM for clustering of the data that may be generated by a DET analysis. The methodology was illustrated using a dataset generated by ADAPT for a simple level controller and an aircraft crash recovery of a conceptual design for a sodium-cooled fast reactor. ADAPT has been linked to RELAP5 to simulate the temporal behavior of the accident transient for the aircraft crash recovery scenarios.

The results show the methodology presented is able to simplify the analysis of large sets of transients generated by dynamic PRA methods. It is shown that grouping scenarios into clusters can be helpful to identify trends and evaluate their characteristics. Such a grouping will be also valuable to identify differences between DET datasets generated for different system configurations.

As a final remark, the application for methodology presented in this paper is not only relevant for the post-processing of dataset generated by dynamic methodologies. In fact, any time a large dataset of multi-dimensional functions (e.g., flux profiles) need to be analyzed, the user may find in the MSM a valuable tool for the analysis.

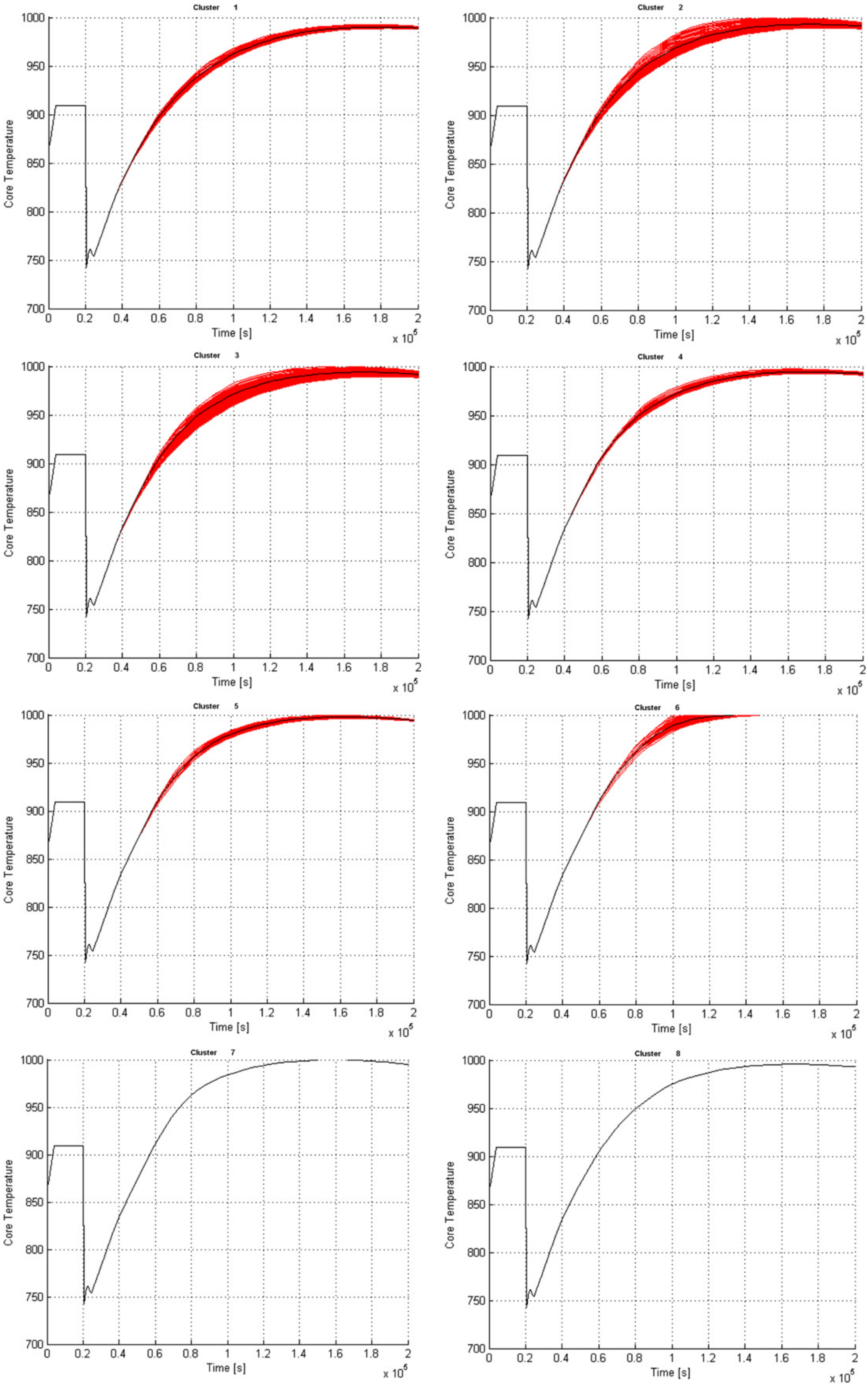

Fig. 19. Cluster centers (black lines) and associated scenarios (red lines) for each cluster. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

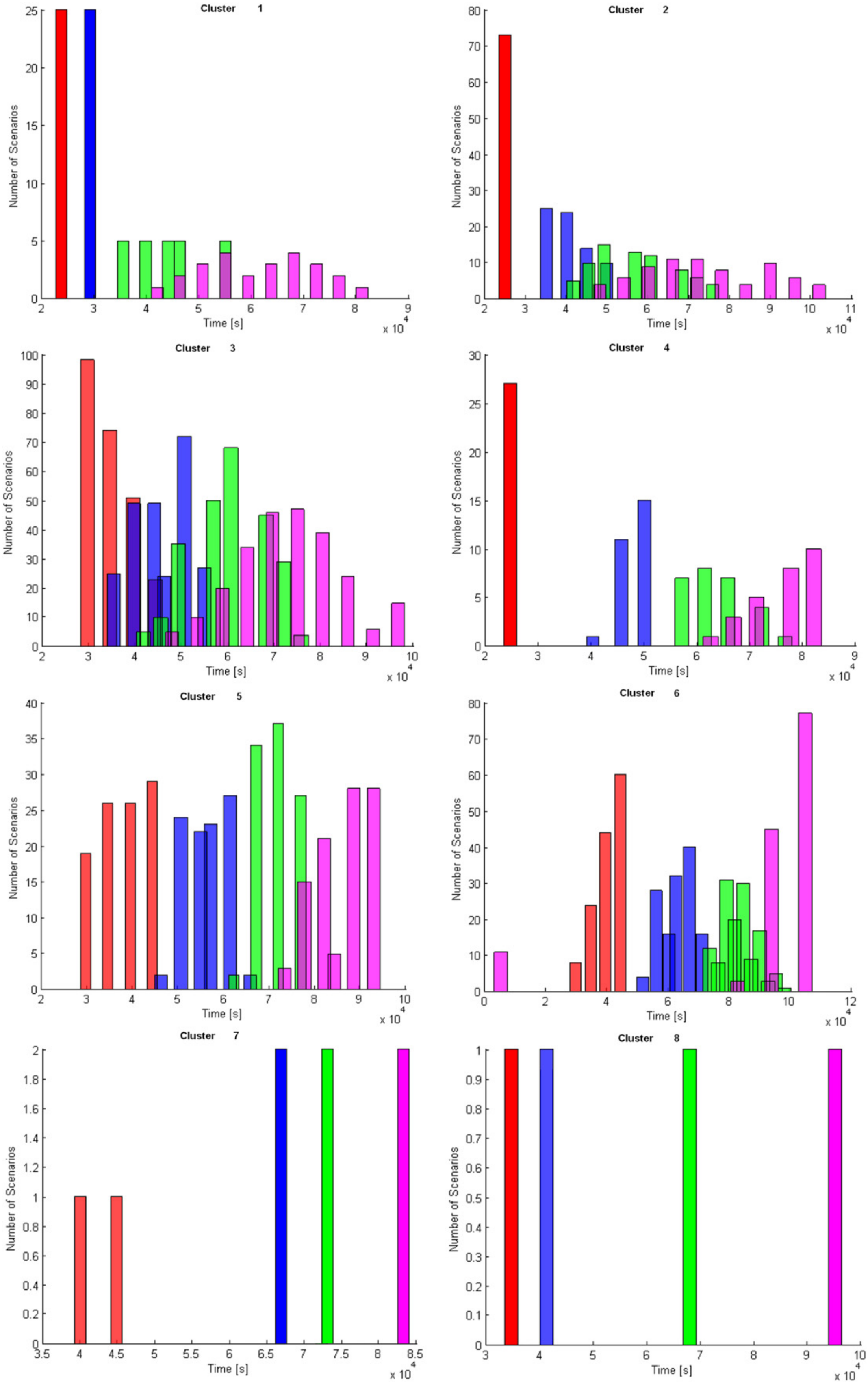

Fig. 20. Distribution of crew arrival time (red) and the recovery time of tower 1 (blue), 2 (green) and 3 (magenta). (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

References

[1] Aldemir T, Miller D, Stovsky M, Kirschenbaum J, Bucci P, Fentiman A, et al. NUREG/CR 6901: current state of reliability modeling methodologies for digital systems and their acceptance criteria for nuclear power plant assessments. Washington, DC: Division of Fuel, Engineering and Radiological Research, Office of Nuclear Regulatory Research, U.S. Nuclear Regulatory Commission; 2006.

[2] Amendola A, Reina G. Dylam-1, a software package for event sequence and consequence spectrum methodology. In: EUR-924, CEC-JRC. ISPRA: Commission of the European Communities; 1984.

[3] Acosta C, Siu N. Dynamic event trees in accident sequence analysis: application to steam generator tube rupture. Reliability Engineering and System Safety 1993;41:135–54.

[4] Hsueh KS, Mosleh A. The development and application of the accident dynamic simulator for dynamic probabilistic risk assessment of nuclear power plants. Reliability Engineering and System Safety 1996;52:297–314.

[5] Hakobyan A, Aldemir T, Denning R, Dunagan S, Kunsman D, Rutt B, et al. Dynamic generation of accident progression event trees. Nuclear Engineering and Design 2008;238(12):3457–67.

[6] Swaminathan S, Smidts C. The mathematical formulation of the event sequence diagram framework. Reliability Engineering and System Safety 1999;65(65):103–18.

[7] Peterson JL. Petri nets. Computing Surveys 1977;9(3):223–52.

[8] Guarro S, Yau M, Motamed M. NUREG/CR-6465: development of tools for safety analysis of control software in advanced reactors. Washington, DC: Office of nuclear regulatory research, U.S. Nuclear Regulatory Commission; 1996.

[9] Marchand S, Tombuyses B, Labeau P. DDET and Monte Carlo simulation to solve some dynamic reliability problems. In: Proceeding on probabilistic safety assessment and management (PSAM4), no. 4; 1998. p. 2055–60.

[10] Aldemir T. Utilization of the cell-to-cell mapping technique to construct Markov failure models for process control systems. In: Proceedings of probabilistic safety assessment and management: PSAM1. New York: Elsevier; 1991. p. 1431–36.

[11] Hofer E, Kloos M, Krzykacz-Hausmann B, Peschke J, Woltereck M. An approximate epistemic uncertainty analysis approach in the presence of epistemic and aleatory uncertainties. Reliability Engineering and System Safety 2002;77:229–38.

[12] RELAP5-3D Code Development Team, RELAP5-3D Code Manual; 2005.

[13] Gauntt RO. MELCOR Computer Code Manual, Version 1.8.5, Vol. 2, Rev. 2. Sandia National Laboratories, NUREG/CR-6119.

[14] Zio E. Reliability engineering: old problems and new challenges. Reliability Engineering and System Safety 2009;94(2):125–41.

[15] U.S.NRC. NUREG 1150 – severe accident risks: an assessment for five U.S. nuclear power plants. Washington, DC: Division of Systems Research, Office of Nuclear Regulatory Research. U.S. Nuclear Regulatory Commission; 1990.

[16] Zio E, Maio FD. Processing dynamic scenarios from a reliability analysis of a nuclear power plant digital instrumentation and control system. Annals of Nuclear Energy 2009;36:1386–99.

[17] Aldemir T. Computer-assisted Markov failure modeling of process control systems. IEEE Transactions on Reliability 1987;36:133–44.

[18] Rui X, Ii. Survey of clustering algorithms. IEEE Transactions on Neural Networks 2005;16(May):645–78.

[19] Jain AK, Murty MN, Flynn PJ. Data clustering: a review. ACM Computing Surveys 1999;31:264–323.

[20] MacQueen JB. Some methods for classification and analysis of multivariate. In: Cam LML, Neyman J, editors. Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, vol. 1. University of California Press; 1967. p. 281–97.

[21] Bezdek JC. Pattern recognition with fuzzy objective function algorithms. Norwell, MA, USA: Kluwer Academic Publishers; 1981.

[22] Fukunaga K, Hostetler L. The estimation of the gradient of a density function, with applications in pattern recognition. IEEE Transactions on Information Theory 1975;21(1):32–40.

[23] van Groenewoud H. A cluster analysis based on graph theory. Plant Ecology 1974;29(2):115–20.

[24] Wei H, Su G, Qiu S, Ni W, Yang X. Applications of genetic neural network for prediction of critical heat flux. International Journal of Thermal Sciences 2010;49(1):143–52.

[25] Mandelli D, Yilmaz A, Aldemir T. Data processing methodologies applied to dynamic PRA: an overview. In: Proceeding of PSA 2011 international topical meeting on probabilistic safety assessment and analysis, CD-ROM. LaGrange Park, IL: American Nuclear Society; 2011.

[26] Duda R, Hart P, Stork D. Pattern classification. Wiley-Interscience Publication; 2000.

[27] Cheng Y. Mean shift, mode seeking, and clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence 1995;17(8):790–9.

[28] Jolliffe IT. Principal component analysis, 2nd ed. Springer; 2002.

[29] Cacoullos T. Estimation of a multivariate density. Annals of the Institute of Statistical Mathematics 1966;18(1):179–89.

[30] Sheikh YA, Khan E, Kanade T. Mode-seeking by medoidshifts. In: Eleventh IEEE international conference on computer vision (ICCV 2007), no. 1; October 2007.

[31] Tombuyses B, Aldemir T. Continuous cell-to-cell mapping. Journal of Sound and Vibration 1997;202(3):395–415.

[32] Bucci P, Kirschenbaum J, Mangan LA, Aldemir T, Smith C, Wood T. Construction of event-tree/fault-tree models from a Markov approach to dynamic system reliability. Reliability Engineering & System Safety 2008;93(11): 1616–27.

[33] Winningham R, Metzroth K, Aldemir T, Denning R. Passive heat removal system recovery following an aircraft crash using dynamic event tree analysis. In: Proceeding of American Nuclear Society (ANS). vol. 100; 2009. p. 461–2.