ANS NPIC HMIT 2009 Topical Meeting - Nuclear Plant Instrumentation, Controls, and Human Machine Interface Technology
Knoxville, Tennessee, April 5-9, 2009, on CD-ROM, American Nuclear Society, LaGrange Park, IL (2008)

IDENTIFICATION OF FAULTS IN A LEVEL CONTROL DYNAMIC SYSTEM

Francesco Di Maio, Marco Stasi and Enrico Zio
Energy Department, Polytechnic of Milan
Via Ponzio 34/3, 20133 Milano, Italy
enrico.zio@polimi.it

Diego Mandelli and Tunc Aldemir
Mechanical Engineering Department
Ohio State Univerity
201 W. 19. Ave. Columbus, OH 43210, USA

ABSTRACT

Identification of faults is an important task in system safety analysis. The large number of system scenarios considered in a realistic safety assessment need to be post-processed to identify the system critical states. The task can be quite complex when the dynamic aspects of the system behavior and its control play a relevant role in the analysis. For example, a fault event may lead to different scenario evolutions depending on the time of failure occurrence and the corresponding state of the controlled process variables. In this work, a Fuzzy C-Means clustering algorithm is applied for the classification of the fault scenarios of a level control system. The classification is based on information from both the stochastic events sequence and the patterns of the process variables evolution.

Key Words: dynamic reliability, fault scenarios classification, fuzzy clustering, Fuzzy C-Means, level control system.

1 INTRODUCTION

The accident behavior of control dynamic systems cannot be completely captured by the classical Probabilistic Risk Assessment (PRA) modeling tools like Event-Trees (ETs) and Fault-Trees (FTs), because they typically do not take into account neither the timing nor the sequencing of failures [1]. However, the sequential order of the failure events and the timing of their occurrence along a stochastic accident scenario may affect its evolution [2-3].

Dynamic reliability approaches have been developed to explicitly take into account the interactions among the physical parameters of the process (temperature, pressure, speed, etc.), the human operators actions and the times and sequencing of the failures of the components [4-6] as well as the presence of software [7-10]. These methods include for example the Events Sequence Diagrams (ESDs) [11], Petri Nets [12], Dynamic Flowgraph Methodology (DFM) [13-14], Discrete Dynamic Event Trees (DDET) [15], DYnamic Logical Analytical Methodology (DYLAM) [16-17] and the Dynamic Event Tree Analysis Method (DETAM) [18]. The main challenge with these techniques is their computational complexity: the number of dynamic scenario branches increases by a power law with the number of occurring events (branch points) [5]. A consequence of such an increase is that the analysis of the results becomes difficult without assisting software tools.

In this work, a supervised evolutionary procedure for the optimization of the Fuzzy C-Means clustering algorithm [19-20] is applied for identifying the classes of scenario evolution corresponding to different end states of a Level Control Dynamic System (LCDS) of a liquid holdup tank [21]. The clustering is based on the timing and magnitude of the components failure events and leads to substantial reduction in the number of dynamic event trees to be analyzed.

The structure of the paper is as follows. In Section 2, the dynamic model of the LCDS is described and the inadequacy of its reliability computation from a purely static viewpoint is highlighted. In Section 3, the scenario classification by fuzzy-clustering is illustrated. The results of the application of the approach to the scenarios of the LCDS are given in Section 4. Conclusions and remarks are given in Section 5.

2 RELIABILITY ANALYSIS OF THE LCDS MODEL

2.1 The Model

The system under analysis is the level controlled tank described in detail in [21]. The liquid level is actively controlled through the actuation of three components (Figure 1): two inlet pumps and one outlet valve, hereafter called Unit 1, 2 and 3, respectively.

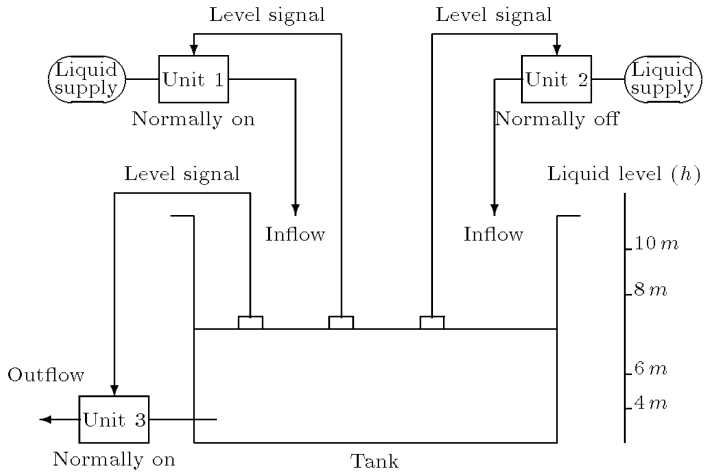

Figure 1. The heated holdup tank [21]

Each unit is a multi-state component operating either correctly ON or OFF (0), stuck ON (1) or stuck OFF (2). At t=0, the system is assumed to be in its nominal state (ON,OFF,ON), with equilibrium values of 30.93 °C of the liquid temperature and 7 [m] of the level. The temperature of the liquid is assumed to directly affect the failure rates of the components [21]. A thermal power source heats up the fluid to keep it almost equal to the nominal temperature, in spite of the level fluctuations. The control laws reported in Table I act upon the state of the components to keep the liquid level h between 4 and 10 [m], the lower and upper safety thresholds, respectively: thus, two possible Top Events need to be considered, i.e. dryout (level $\leq$ 4 [m]) and overflow (level $\geq$ 10 [m]).

Table I. The control laws [21]

	Control laws
1	If the liquid level *h* drops under 6 [m], the units 1, 2, 3 are put respectively in state ON, ON and OFF (if they are not stuck ON or OFF)
2	If the liquid level *h* rises above 8 [m], the units 1, 2, 3 are put respectively in state OFF, OFF and ON (if they are not stuck ON or OFF)

2.2 The system simulation

The simulation of the dynamic system scenarios has been performed using the Markov/Cell-to-Cell Mapping Technique (CCMT) methodology [21], which describes probabilistically the dynamics of the level and temperature process variables and of the process controller. Each one of the three multi-state system components (the two inlet pumps and the outlet valve) is modeled by a Markov process; the CCMT describes the dynamics of the system in discrete time, in terms of transitions between computational cells in the discretized system state space [22].

The evolution of the dynamic system depends on:

- the deterministic process equations governing the evolution of the level and temperature variables;
- the laws of the control system (i.e., Table I);
- the stochastic process governing the transport (i.e. configuration change) of the components among their different reachable states.

The dynamics is modeled as transitions between cells V_j (j = 1,…, J) that partition the complete state space describing both the values of the process variables and the configuration of the system; the transitions occur with probabilities $g(j|j',n',k)$ for the controlled variables and $h(n|n',j'\rightarrow j)$ for the system configurations.

The cell-to-cell transition probabilities $g(j|j',n',k)$ are conditional probabilities that the controlled variables are in the cell V_j at time $t=(k+1)\Delta t$ given that:

- the controlled variables are in the cell $V_{j'}$ at time $t=k\Delta t$ and,
- the system configuration is $n(k)=n'$ at time t.

These probabilities are determined from the simulation of system behavior during Δt under the assumption that the components do not change states within Δt as described in [21].

The stochastic transport of the system state is described by $h(n|n',j'\rightarrow j)$, which is the probability that the system configuration at time $t=(k+1)\Delta t$ is n, given that:

- $n(k)=n'$ at $t=k\Delta t$ and
- the controlled variables transit from cell $V_{j'}$ to cell V_j during $k\Delta t\leq t<(k+1)\Delta t$.

Starting from given initial conditions, the Markov/CCMT algorithm leads to the computation of the probability $p_{n,j}(k)$ that the system is in cell j with configuration n at time $t=k\Delta t$ from

$$p_{n,j}(k+1) = \sum_{n'=1}^{N} \sum_{j'=1}^{J} g(j \mid n', j', k) h(n \mid n', j' \rightarrow j, k) p_{n',j'}(k)$$

The $p_{n,j}(k)$ can then be converted into dynamic event trees for specified initial and end states [23].

2.3 The reliability analysis

The three possible states of each of the three components (Unit 1, 2 and 3) combine into a total of 27 possible system configurations (Table II). To identify the corresponding system end states, a large number of simulations of the Markov/CCMT has been run for each of the 27 configurations, with different orderings in the sequence of events for the multiple failure configurations. In the simulations, the faults are conservatively considered to occur at the beginning of the scenarios and those system configurations that lead into overflow or dryout failure mode are identified as minimal cut sets (MCS) for the respective system end state.

Table II. System configurations: 0 = safe component, 1 = stuck ON component, 2 = stuck OFF component

System state	*Unit 1 state*	*Unit 2 state*	*Unit 3 state*	*Failure mode*	
				Overflow	*Dryout*
0	0	0	0	*no*	*no*
1	0	0	1	*no*	*no*
2	0	0	2	*no*	*no*
3	0	1	0	*no*	*no*
4	0	1	1	*no*	*no*
5	0	1	2	*yes*	*no*
6	0	2	0	*no*	*no*
7	0	2	1	*no*	*no*
8	0	2	2	*no*	*no*
9	1	0	0	*no*	*no*
10	1	0	1	*no*	*no*
11	1	0	2	*yes*	*no*
12	1	1	0	*yes*	*no*
13	1	1	1	*yes*	*No*
14	1	1	2	*yes*	*No*
15	1	2	0	*no*	*No*
16	1	2	1	*no*	*no*
17	1	2	2	*yes*	*no*
18	2	0	0	*no*	*no*
19	2	0	1	*no*	*no*
20	2	0	2	*no*	*no*
21	2	1	0	*no*	*no*
22	2	1	1	*no*	*no*
23	2	1	2	*yes*	*no*
24	2	2	0	*no*	*no*
25	2	2	1	*no*	*yes*
26	2	2	2	*no*	*no*

Three second-order MCS are found for the overflow end state (Figure 2). Figures 3-5 show the liquid level evolution resulting from the MCS 1, 2, 3 occurrence at time t=0 (i.e., system configurations 5, 11, 12 in Table II). In all three cases, the overflow is shown to occur within 5 hours.

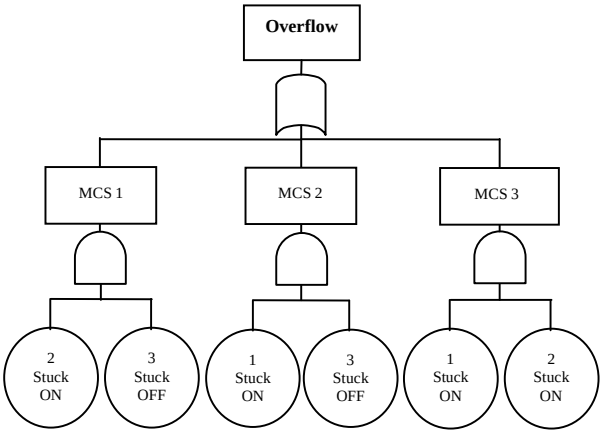

Figure 2. FT for the overflow failure mode

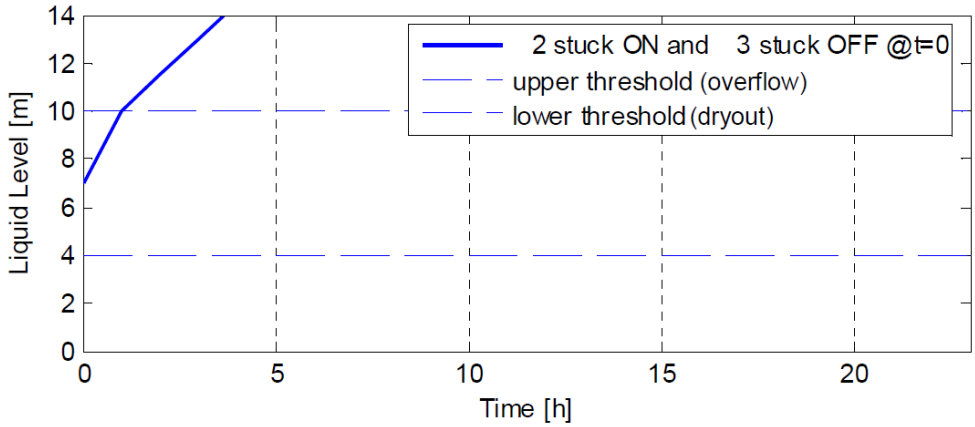

Figure 3. Overflow failure mode: liquid level evolution resulting from the MCS 1 occurrence at time *t*=0

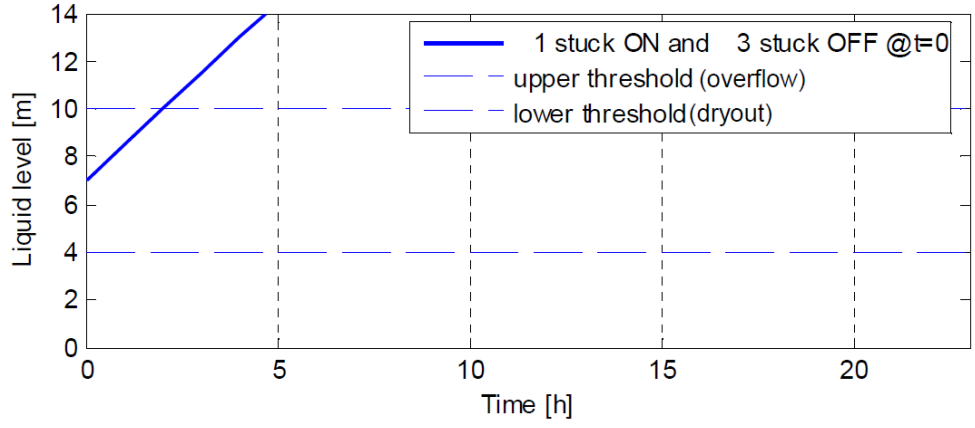

Figure 4. Overflow failure mode: liquid level evolution resulting from the MCS 2 occurrence at time *t*=0

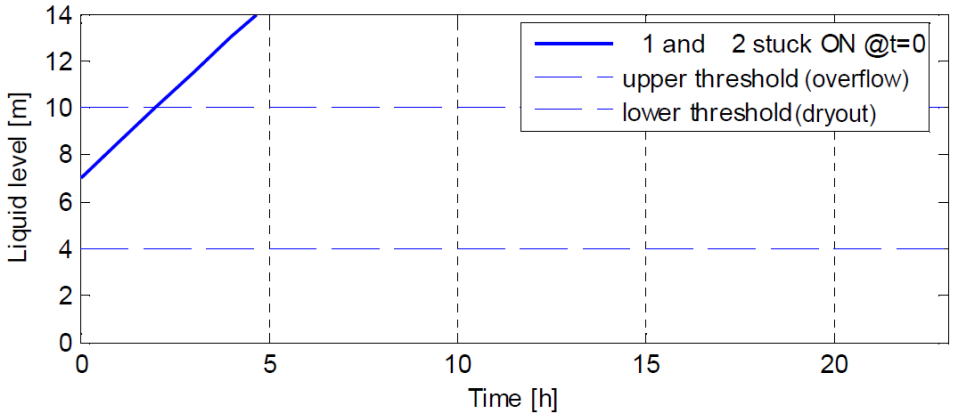

Figure 5. Overflow failure mode: liquid level evolution resulting from MCS 3 occurrence at time *t*=0

For the dryout failure mode of the system, only one third-order MCS is identified, corresponding to system configuration 25 in Table II (Figure 6).

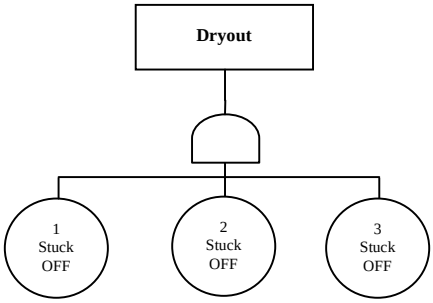

Figure 6. FT for the dryout failure mode

Figure 7 shows the liquid level evolution arising from the occurrence of the MCS of the dryout at time *t*=0, i.e. the units 1 and 2 stuck OFF and unit 3 stuck ON, at time *t*=0. Dryout conditions are reached within 5 hours.

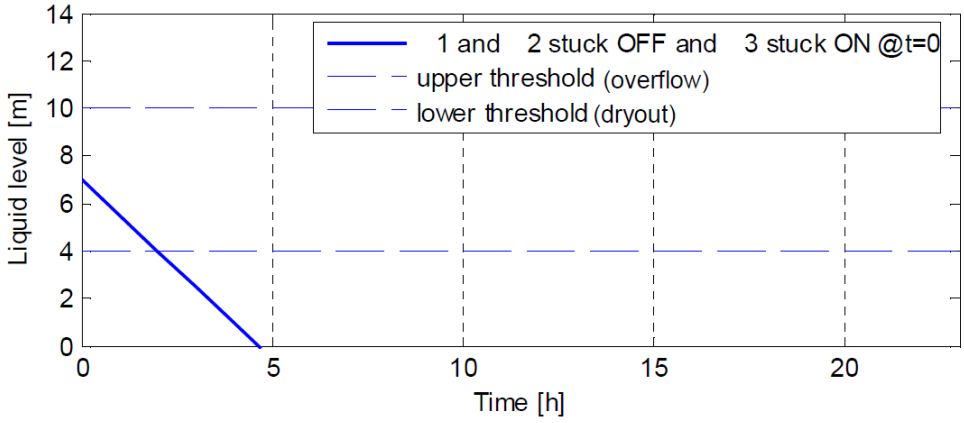

Figure 7. Dryout failure mode: liquid level evolution resulting from the MCS occurrence at time *t*=0

The system failure analysis performed is based on the conservative assumption that the units failure events occur at t=0; this hides the role of the ordering and timing of the failure events occurrence in determining the time at which the system reaches a faulty end state. The information on the system time of failure is an important parameter for determining the available time of recovery of the different accident situations.

The empirical probability density functions (pdfs) of the times of system overflow and dryout, obtained from a total number of 76556 Markov/CCMT simulations of system evolutions with different component failures ordering and timing, actually show that the time and order of the failures of the components affect the system failure time.

The proposed approach illustrated in the following Sections provides a method for analyzing the large number of dynamic accident scenarios which need to be considered to identify the system end state, including the identification of the system failure time (i.e., the time in which the system safety parameters exceed pre-defined safety thresholds).

3 CLUSTERING DYNAMIC SCENARIOS

The processing of the dynamic scenarios of the LCDS aims at identifying classes of behavioral similarity in the scenario evolutions and at relating these with characteristic features of the accident sequence, e.g. its end state and timing. The classification approach is based on combining information from the stochastic timing and magnitude of the component failure events along the sequence.

The uncertainties and overlaps in the definition of the different classes of accident sequences can be modeled within a fuzzy clustering paradigm for classification. The scheme here adopted is founded on a supervised evolutionary optimization of a Fuzzy-C-Means (FCM) clustering algorithm developed by the Laboratorio di Analisi di Segnale ed Analisi di Rischio (LASAR, Laboratory of Analysis of Signals and of Analysis of Risk, http://lasar.cesnef.polimi.it) of the Department of Energy of the Polytechnic of Milan, Italy [24]. Clustering in a predefined feature space is performed based on a Mahalanobis metric which is iteratively optimized so that the obtained clusters are close to the actual scenario classes [24].

The basic steps of the scenario classification approach are sketched in Figure 8. The first step is the selection of the relevant classification features. The successive steps of the procedure are typical of a supervised classification scheme: training of the classifier on patterns of known classes and testing of the classifier on new patterns.

3.1 The supervised evolutionary clustering classifier

In general, the task of pattern classification may be viewed as a problem of partitioning of objects (hereafter also called data, patterns) into classes. The traditional unsupervised clustering algorithm based on a Euclidean metric to measure the similarity among the patterns of features leads to spherical clusters that rarely are adequate to represent the actual features data partition in practice. On the contrary, using specific Mahalanobis metrics for defining the different clusters allows obtaining different ellipsoidal shapes and orientations of the clusters that can more adequately fit the actual data partition [25].

Furthermore, the information on the membership of the patterns $\vec{x}_k$, k=1,…, N, to the c a priori known classes, can be used to supervise the clustering algorithm for finding the optimal

Mahalanobis metrics such as to achieve geometric clusters as close as possible to the a priori known physical classes.

The supervised training procedure for the optimization of the Mahalanobis metrics defining the clusters is carried out via an evolutionary procedure, presented in the literature within a supervised fuzzy clustering scheme [19] and further extended to diagnostic applications [25].

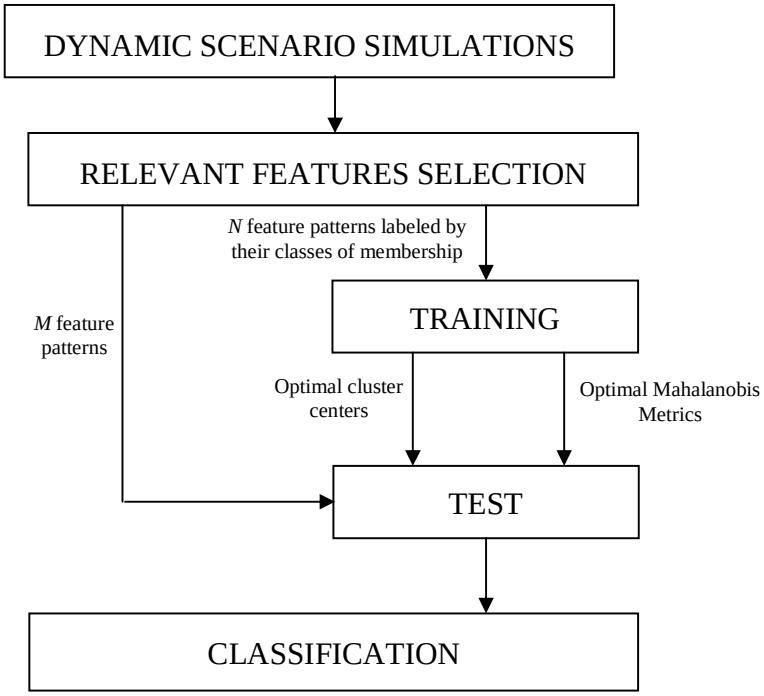

Figure 8. The steps of the scenario classification approach

The target of the supervised optimization is to find *c* optimal Mahalanobis metrics which define *c* geometric clusters of the available data set such to minimize the distance $D\left(\Gamma^t,\Gamma\right)$ between the a priori known physical class partition $\Gamma^t \equiv \left(\Gamma_1^t,\Gamma_2^t,...,\Gamma_c^t\right)$ and the obtained geometric cluster partition $\Gamma \equiv \left(\Gamma_1,\Gamma_2,...,\Gamma_c\right)$:

$$D(\Gamma^t,\Gamma\ ) = \sum_{i=1}^{c} \frac{D(\Gamma_i^t,\Gamma_i)}{c} = \sum_{i=1}^{c}\sum_{k=1}^{N} \frac{|\,\mu_i^t\left(\vec{x}_k\right) - \mu_i^*\left(\vec{x}_k\right)|}{N\cdot c}$$

Where $0 \le \mu_i^t\left(\vec{x}_k\right) \le 1$ is the a priori known membership of the *k*-th pattern to the *i*-th physical class (possibly not known with absolute precision, in which case it has a membership less than one) and $0 \le \mu_i^*\left(\vec{x}_k\right) \le 1$ is the membership to the corresponding geometric cluster in the feature space.

When fed with a new pattern of features $\vec{x}$ characteristic of a given dynamic scenario, the classification algorithm provides the values of the membership functions $\mu_i^*(\vec{x}), i = 1,2,..., c$, to the different clusters which represent the scenario classes in the stochastic and process variable feature space.

4 LCDS SCENARIO CLASSIFICATION BY FUZZY CLUSTERING

For the fault scenarios of the LCDS, seven classes (c=7) have been identified for classifying the accident scenarios with respect to the end state of the accident sequence (overflow, safe and dryout) and with sufficient practical resolution in terms of its time of occurrence (in [0,8] h, in [9,16] h, in [17,24] h):

Class 1: Overflow failure in [0,8] [h].
Class 2: Overflow failure in [9,16] [h].
Class 3: Overflow failure in [17,24] [h].
Class 4: Safe transient.
Class 5: Dryout failure in [0,8] [h].
Class 6: Dryout failure in [9,16] [h].
Class 7: Dryout failure in [17,24] [h].

The features selected as relevant for the characteristics of the accident sequence evolutions are:

Feature 1: s_1: Unit 1 state $s_1 \in \{0,1,2\}$;
Feature 2: t_1: Unit 1 failure time $t_1 \in [0,24]$;
Feature 3: s_2: Unit 2 state $s_2 \in \{0,1,2\}$;
Feature 4: t_2: Unit 2 failure time $t_2 \in [0,24]$;
Feature 5: s_3: Unit 3 state $s_3 \in \{0,1,2\}$;
Feature 6: t_3: Unit 3 failure time $t_3 \in [0,24]$;

The evolutionary training of the FCM classifier has been performed on the basis of a set of N=500 class-labeled patterns randomly extracted from the 76556 total scenario simulations, each one constituted by an input vector of 6 values for the 6 features (times t_1, t_2, t_3, states s_1, s_2, s_3) and one output value (the class label 1, 2,…,7).

Once the classifier has been constructed, it needs to be tested before it can be used to classify any pattern of dynamic scenario. In the present work, the classifier has been cross-validated by feeding it with 10 different batches of input/output samples of M=250 different scenario simulations random drawn from the 76556-N available for test. The mean values of the performances of the classifiers and the ±1σ uncertainty bandwidths are shown in Figure 9.

The upper subplot of Figure 9 shows the fraction of patterns that have been correctly classified into a class with a membership value to such class larger than the threshold in abscissa; the second subplot shows the fraction of patterns that have been incorrectly classified into a class with a membership value to such class larger than the threshold in abscissa; the lower subplot shows the fraction of patterns that, due to the insufficient membership to all of the classes, with respect to the threshold value in abscissa, have not been classified as belonging to anyone of the predefined classes. The threshold value in abscissa is an arbitrary choice: the optimum value is that which maximizes the fraction of correct classifications and minimizes that of incorrect ones.

In Table III, numerical results are reported with a threshold value equal to 0.7, which has been found to provide satisfactory classification performances for the purpose of the present work.

In alternative, patterns can be simply assigned to the class for which the membership value of the pattern is the highest, independently of whether this value is above a given threshold; in this case, none of the patterns can be classified as ambiguous (Table III).

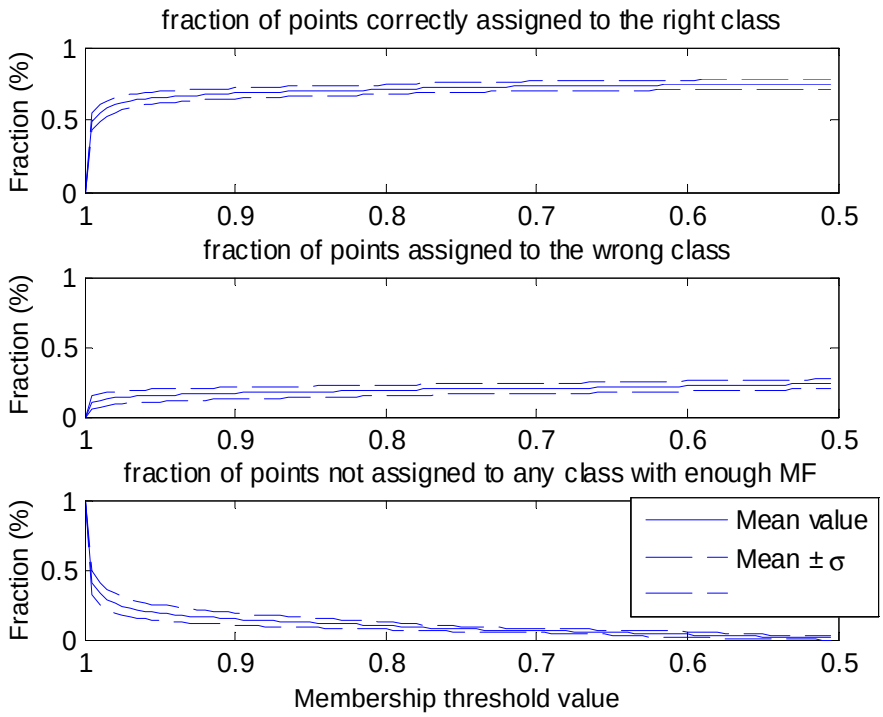

Figure 9. Classifier performance with respect to 10 sets of 250 test patterns: fraction of patterns correctly (top), incorrectly (middle) and not (bottom) classified, as function of the membership threshold value in abscissa

Table III. Numerical results of the classifier performance on test patterns

Assignment strategy	Class assignment		
	Correct	Error	Ambiguous
Membership threshold (=0.7)	73.24 ± 3.47%	20.28 ± 3.64%	6.48 ± 1.49%
Maximum membership	75.76 ± 3.48%	24.24 ± 3.48%	0

5 CONCLUSIONS

Dynamic safety and reliability analyses of realistic systems inevitably lead to the evaluation of a large number of different time-dependent scenarios, which need to be post-processed to identify critical states. This work has treated the post-processing of accident scenarios for identifying system fault classes.

Dynamic scenarios have been simulated for the LCDS of an heated holdup tank. The safety parameter of interest is the level of liquid contained in the heated tank which cannot drop under the value of 4 [m] (dryout failure mode) nor exceed the value of 10 [m] (overflow failure mode). The timing of component failures has been shown to impact the system failure time, which is in turn a relevant parameter for determining the available time of recovery of the different accident situations. A partition of the system failure time into three intervals has been considered practically sufficient for the purpose of the work; this gives rise to seven classes, defined based on safe/faulty end states and corresponding system failure times.

Classification of the dynamic scenarios of system evolution with similar characteristics has been carried out on the basis of the features of the occurred stochastic events, i.e. the components failure times and modes, by an optimized fuzzy clustering scheme. The classification performance achieved is of approximately 75%, which demonstrates the feasibility of the approach.

6 REFERENCES

1. Rutt, B., Catalyurek, U., Hakobyan, A., Metzroth, K., Aldemir, T., Denning, R., Dunagan, S. and Kunsman, D., “Distributed Dynamic Event Tree Generation for Reliability and Risk Assessment”, 1-4244-0420-7/06, *IEEE* (2006).
2. Siu, N., “Risk assessment for dynamic systems: an overview”, *Reliability Engineering and System Safety*, pp.43, 43-73 (1994).
3. Aldemir, T., Stovsky, M.P., Kirschenbaum, J., Mandelli, D., Bucci, P., Mangan, L.A., Miller, D.W., Sun, X., Ekici, E., Guarro, S., Yau, M., Johnson, B.W., Elks, C., Arndt, S.A., “Dynamic Reliability Modeling of Digital Instrumentation and Control Systems for Nuclear Reactor Probabilistic Risk Assessments”, *NUREG/CR-6942, U.S. Nuclear Regulatory Commission*, Washington, D.C. (2007).
4. Devooght, J., “Dynamic reliability”, *Advances in Nuclear Science and Technology*, **25**, pp. 215-278 (1997).
5. Labeau, P.E., Smidts, C., Swaminthan, S., “Dynamic Reliability: Towards an Integrated Platform for Probabilistic Risk Assessment”, *Reliability Engineering and System Safety*, **68**, pp. 219-254 (2000).
6. Dufour, F. and Dutuit, Y., “Dynamic Reliability: A New Model”, *13th ESREL2002 European Conference*, Lyon-France, 18-21 March 2002.
7. Guarro, S., Yau, M. and Oliva, S., "Conditional Risk Model Concept for Critical Space Systems Software", *Proc. of the Probabilistic Safety Assessment and Management: PSAM 7-ESREL '04*, 158-163, Springer – Verlag, London, U.K., (2004).
8. Dixon, S, Yau, M. and Guarro, S., "Demonstration of the Context-Based Software Risk Model Method for Risk Informed Assurance and Test of Software-Intensive Space Systems", *Proceedings of the 9th International Conference on Probabilisitc Safety Assessment and Management*, Hong Kong, China, (2008).
9. Shields, E.J., Apostolakis, G., and Guarro, S., "Determining the Prime Implicants for Multi-State Embedded Systems", *Proceedings of PSAM-II, 7-12, International Asoociation for Probabilistic Assessment and Management*, San Diego, California, (1994).

10. Yau, M., Apoatolakis, G. and Guarro, S., "The Use of Prime Implicants in Dependability Analysis of Software Controlled Systems", *Reliability Engineering and System Safety*, **62**, 23-32, (1998).
11. Swaminathan, S. and Smidts, C., "The Event Sequence Diagram Framework for Dynamic PRA", *Reliability Engineering and System Safety*, **63**, pp. 73-90 (1999).
12. Peterson, J. L., "Petri Nets", *ACM Computing Surveys*, **9**, pp.223-252 (1977).
13. Guarro, S., Yau, M. and Motamed, M., "Development of Tools for Safety Analysis of Control Software in Advanced Reactors", *NUREG/CR-6465, U.S. Nuclear Regulatory Commission*, Washington, D.C. (1996).
14. Garret, C. J. and Apostolakis, G. E., "Automated Hazard Analysis of Digital Control Systems", *Reliab.Engng & System Safety*, **77**, pp.1-17 (2002).
15. Marchand, S., Tombuyes, B. and Labeau, P., "DDET and Monte Carlo Simulation to Solve Some Dynamic Reliability Problems", *PSAM 4*, **3**, pp.2055-2060, New York (1998).
16. Amendola, A. and Reina, G., "DYLAM-1, A Software Package for Event Sequence and Consequence Spectrum Methodology", *EUR-924, CEC-JRC*, Ispra, Commission of the European Communities (1984).
17. Cojazzi, G., "The DYLAM Approach to the Dynamic Reliability Analysis of Systems", *Reliab.Engng & System Safety*, **52**, pp.279-296 (1996).
18. Acosta, C. and Siu, N., "Dynamic Event Trees in Accident Sequence Analysis: Application to Steam Generator Tube Rupture", *Reliab.Engng & System Safety*, **41**, pp.135-154 (1993).
19. Yuan B., Klir G. and Swan-Stone J., "Evolutionary fuzzy c-means clustering algorithm", *Proc. Fourth IEEE International Conference on Fuzzy Systems*, pp. 2221–2226 (1995).
20. Bezdek, J.C., "Pattern Recognition With Fuzzy Objective Function Algorithms", *Plenum*, New York (1981).
21. Aldemir, T., "Utilization of the Cell-To-Cell Mapping Technique to Construct Markov Failure Models for Process Control Systems", *Proc. of Probabilistic Safety Assessment and Management: PSAM1*, 1431-1436, Elsevier, New York, (1991).
22. Hsu, C. S., *Cell-to-cell Mapping: A Method of Global Analysis for Nonlinear Systems*, Springer-Verlag, New York, NY (1987).
23. Bucci, P., Kirschenbaum, J., Aldemir, T., Smith, C. L and Wood, R.T., "Constructing Dynamic Event Trees From Markov Models", M. STAMATALETOS and H. S. BLACKMAN (Eds.), *PSAM8: Proceedings of the 8th International Conference on Probabilistic Safety Assessment and Management*, CD-ROM Version, Paper # 369, ASME Press, Inc., (2006).
24. Zio E. and Baraldi P., "Identification of Nuclear Transients via Optimized Fuzzy Clustering", *Annals of Nuclear Energy*, **32**, pp.1068-1080 (2005).
25. Yuan B., and Klir G., *Intelligent Hybrid Systems Fuzzy Logic, Neural Network and Genetic Algorithms*, Ruan, D. (Ed.), Kluver Academic Publishers (1997).