Dynamic reliability methods (in support to a safety assessment)

Accidental scenarios simulation and processing

Francesco DI Maio
Energy Department
francesco.dimaio@polimi.it
The objectives of (nuclear) safety

**THREAT** (Release of radioactivity from a Nuclear Power Plant (NPP))

**REGULATIONS**

- *Design*
- *Operation*
- *Decommissioning*

**Objectives of a safety assessment:**

- ensure that (nuclear) facilities operate **normally** and without an excessive risk of operating staff and the environment
- prevent **incidents** and limit the consequences of any incidents that might occur.
Application: The LBE-XADS

- Accelerator Driven System (ADS)
  - A means of transmuting nuclear waste
  - A new type of fission reactor
  - Both

- How?
  - Some of the “afterheat” of spent nuclear fuel can be captured in a power generator, instead of a mountain
    - Goal is 95% of Minor Actinides (MA) & Long-Lived Fission Products (LLFPs) transmuted
The Concept

Proton accelerator creates neutrons by **spallating** high-Z target nuclei (Z=atomic/proton number)

Spallation neutrons used to maintain fission reaction where not normally possible

- Subcritical piles
- In waste actinides
- Chain reaction can’t exist w/o accelerator: To stop, just unplug it
Lead Bismuth Eutectic Accelerator Driven System (LBE-XADS)
General View inside the Subcritical Assembly
Deterministic Safety Assessment (DSA)
Probabilistic Safety Assessment (PSA)
Complementary methods of safety analysis are used jointly in evaluating the safety:

1. Deterministic Safety Analysis (DSA)
2. Probabilistic Safety Analysis (PSA)
Deterministic Safety Assessment (DSA)

Probabilistic Safety Assessment (PSA)
The basic principles of DSA

1) Presence of leaktight **barriers** between the radioactive source and the public/environment

- fuel cladding (1)
- primary reactor coolant system (2 and 3)
- containment building (4)
The basic principles of DSA

2) **Defence-in-depth**, which applies to both the design and the operation of the facility

*Accidents may still occur*

*Systems are to be designed and installed to combat them and to ensure that their consequences are limited to a level that is acceptable for both the public and the environment*

[Commission’s White Paper, February, 1999]
The defence-in-depth principle

Prevention and surveillance:
- Items of equipment are designed with adequate safety margins

Protection:
- Provisions are made to detect incidents and to prevent them from escalating → safety systems are designed to restore the plant to a normal state and maintain it under safe conditions.

Safeguard:
- Special safety systems are therefore designed to limit the consequences of severe accidents to an acceptable level.

Off-site and emergency procedures:
- Designed to provide protection against severe conditions under which defences at the three levels described above prove inadequate.
**Design Basis Accidents (DBA)** [NUREG/CR-6042, USNRC, 1994]

- Very unlikely events
- Small in number
- Conservative initial conditions

**CONCERNS ABOUT UNCERTAINTY MANAGEMENT**
“Conservatism” of DSA

Protection against assumed conservative threat

Normal condition

Assumed threats profile

Assumed conservative case
Importance of the principles of DSA

SAFETY BY DESIGN
Subcritical reactor

MULTI-BARRIERS:
- Pellet
- Safety vessel
- Containment
- Secondary coolant
The accident in 1986 at Chernobyl (I)

- The accident, caused by a sudden surge of power, destroyed the reactor and released massive amounts of radioactive material into the environment.
- Lack of containment.
The accident in 1986 at Chernobyl (I)

- The accident, caused by a sudden surge of power, destroyed the reactor and released massive amounts of radioactive material into the environment
- Lack of containment

The Chernobyl experience should remain a valuable part of the information to be taken into account when dealing with reactor safety issues in the future

→ importance of the principles of DSA
Deterministic Safety Assessment (DSA)

Probabilistic Safety Assessment (PSA)
Credible (DBAs) VS incredible accidents

not logical

ALL is based on assumptions in expert judgment about the credible cases

Classification of events according to the probability of its occurrence and potential consequences

• **Design-basis accident (DBA):**
• **Beyond design-basis accidents (BDBA):**

Accident sequences that are possible but were not fully considered in the design process because they were judged to be too unlikely.
Prior Beliefs

Protect against Large LOCA
Core Damage Frequency (CDF) is low
   (about once every 100 million years, $10^{-8}$ per reactor year)
Consequences of accidents would be disastrous

Major Findings

Dominant contributors: Small LOCAs and Transients
CDF higher than earlier believed
   (best estimate: $5 \times 10^{-5}$, once every 20,000 years; upper bound: $3 \times 10^{-4}$ per reactor year, once every 3,333 years)
Consequences significantly smaller
Support systems and operator actions very important
The concept of risk

Risk and Safety are considered as two opposed terms, that can be used to assess a same thing

Remember PRA = PSA

• What can go wrong? (accident sequences or scenarios) $S_i$
• How likely are these scenarios? $\pi_i$
• What are their consequences? $C_i$

Definition of Risk

$$R = \{S_i, \pi_i, C_i\}$$

Three different aspects of risk: Risk assessment, Risk perception, Risk communication

Decision makers can be influenced by all of these
The concept of risk

"Farmer curve"
"Conservatism" of DSA $\rightarrow$ PSA

Protection against assumed conservative threat

"Surprise" Ignored scenario

Real Risk profile

Assumed threats profile

Assumed Conservative case
The accident in 1979 at the Three Mile Island (TMI-2)

- Combination of equipment malfunctions, design-related problems and worker errors led to TMI-2's partial meltdown and very small off-site releases of radioactivity
- Small radioactive releases
- No detectable health effects on plant workers or the public

→ Reveals potential consequences of failure to manage NPP safety only based on DSA
Application: LBE-XADS

The model of the LBE-XADS has been embedded within an MC-driven fault injection engine to sample component failures.

4 control/actuator faults
64 accident scenarios


Level I PSA: an example

Level I

PLANT MODEL
(Physics, Fuel and Thermal Hydraulic Models)

Fault Tree Analysis

Results
Accident sequences leading to plant damage states
(CDF frequencies)
Limitations of PSA (I)

PSA contain **uncertainties** arising from three main sources:

- **lack of comprehensive data**: It is impossible to demonstrate the exhaustiveness of a PSA

- **reliability of data**: frequency of initiating events, common-mode failures and failures resulting from human actions.

- **modelling assumptions** that cannot easily be quantified
The accident in 2011 at the Daichi NPP

- Following a major earthquake, a 15-metre tsunami disabled the power supply and cooling of three Fukushima Daiichi reactors, causing a nuclear accident on 11 March 2011.
- All three cores largely melted in the first three days.

- Combination of external events (earthquake and tsunami)
- Multi-unit severe accident
Deterministic Safety Assessment (DSA)

Probabilistic Safety Assessment (PSA)

Integrated Deterministic Probabilistic Safety Assessment

\[ \text{IDPSA} = \text{DSA} + \text{PSA} \]
Complementary **methods** of safety analysis are used jointly in evaluating the safety of an NPP:

1. Deterministic Safety Analysis (DSA)
2. Probabilistic Safety Analysis (PSA)
Integrated Deterministic-Probabilistic Safety Analysis (IDPSA) is a collective name for the variety of different tools which use tightly coupled probabilistic and deterministic approaches to address in a consistent manner:

- *aleatory* (stochastic aspects of scenario) uncertainties
- *epistemic* (modelling) uncertainties
Benefits of integration

IDPSA can provide additional help to PSA and DSA practitioners in:

1) **Resolving time dependent interactions between**
   - physical phenomena,
   - equipment failures,
   - safety and non-safety systems interactions,
   - control logic and operator actions.

2) **Identification of a-priori unknown vulnerable scenarios, or “sleeping threats”**

3) **Reduction of reliance on expert judgment and simplifying assumptions about complex interdependencies**
   - PSA and DSA can calculate what we know as an issue
   - PSA and DSA are not capable of revealing what, and to what extent, we do not know
The problem:
Unknown plant vulnerabilities are left unknown in traditional DSA and PSA.
The problem:
Unknown plant vulnerabilities are left unknown in traditional DSA and PSA

Protection against assumed conservative threat

“Surprise” ignored scenario

Real Risk profile

Assumed threats profile

Assumed Conservative case (Precautionary principle)

APPROXIMATED threat profiles

WORRY!!!!
Wrong safety assessment
Integrated Deterministic and Probabilistic Safety Assessment (IDPSA=DSA + PSA)

Protection against assumed conservative threat

IDPSA IDENTIFIED SCENARIO!!!

Scenarios generation
Scenarios modelling
Scenario Processing

PRIME IMPLICANTS

«CORRECT» THREAT PROFILE

Real Risk profile

Assumed threats profile

Assumed Conservative case
(Precautionary principle)
The model of the LBE-XADS has been embedded within an MC-driven fault injection engine to sample component failures.


• **Static:** PSA is “Yes/No”, “black-and-white”
LBE-XADS: magnitude effects

Dependence of the accident sequence end state with respect to the components faults magnitudes

Scenario: PID fails at t=1500 [s]
- PID controller fails at time t=1500, output @ 500 [kg/s]
- PID controller fails at time t=1500, output @ 220 [kg/s]
- PID controller fails at time t=1500, output @ 100 [kg/s]
- PID controller fails at time t=1500, output @ 0 [kg/s]

Scenario: PID fails at t=1500 [s], air coolers fail at t=2000 [s]
- PID output @ 500 [kg/s], air coolers @ 300 [kg/s]
- PID output @ 220 [kg/s], air coolers @ 500 [kg/s]
- PID output @ 100 [kg/s], air coolers @ 500 [kg/s]
- PID output @ 0 [kg/s], air coolers @ 300 [kg/s]
LBE-XADS: effect of the ORDER of failure events

Dependence of the accident sequence end state from the ordering of the components faults in the sequence

Scenario: PID fails at t=1500 [s], air coolers fail at t=2000 [s]

Scenario: Air coolers fail at t=1500 [s], PID fails at t=2000 [s]
Dependence of the accident sequence end state on the exact timing of the components failures
LBE-XADS: dynamic effects

- Low-temperature failure mode
- Safe mode
- High-temperature failure mode

Frequency of the end state occurrence

Failure sequence
Dynamic methodologies are defined as those that explicitly

- account for the time element in probabilistic system evolution
- are needed when the system has hardware/process/software/human interactions

- SCENARIOS GENERATION
- SCENARIOS MODELLING
- SCENARIOS POST-PROCESSING
Dynamic accidental scenarios generation
Review of the basics of Monte Carlo (MC) simulation of system failure processes

- Inverse Transform method
- Rejection Method
- Direct/Indirect MC simulation

Multi-Valued Logic (MVL) for modelling the system fault states
Review of the basics of Monte Carlo (MC) simulation of system failure processes
System failure process

- **Failure (EN 13306, 2001)**
  - Termination of the ability of an item to perform a required function

- **Fault (EN 13306, 2001)**
  - State of an item characterized by inability to perform a required function, excluding the inability during preventive maintenance or other planned actions, or due to lack of external resources

- **Degradation (EN 13306, 2001)**
  - Irreversible process in one or more characteristics of an item with either time, use or an external cause
Failures are stochastic events

**failure is a well described random phenomena**

\[ R = \Pr \{ \text{system does not fail into } [0, t] \} \]

"Probability that the state of the system is S1 at time t, assuming that it was in between 0 and t"

\[ f_{\theta}(t) = \Pr(\theta = t) \]

\[ F_{\theta}(t) = \Pr(\theta \leq t) \]

\[ R(t) = 1 - F_{\theta}(t) = 1 - \int_{\theta=0}^{t} f_{\theta}(u) \, du \]

- if the random variable \( \theta \) denotes the time to failure
Monte Carlo simulation of failures

The random variable \( \vartheta \) denotes the time to failure \( T \)

DYNAMIC ACCIDENTAL SCENARIO GENERATION consists in generating random failures times of the system components (items) and recording the system evolution in time

MONTE CARLO can be used to sample the components failure times
Monte Carlo (MC) sampling
Sample R from $U_R(r)$ and find $X$:

$$X = F_X^{-1}(R)$$

**Question:** which distribution does $X$ obey?

$$P\{X \leq x\} = P\{F_X^{-1}(R) \leq x\}$$

Application of the operator $F_X$ to the argument of $P$ above yields

$$P\{X \leq x\} = P\{R \leq F_X(x)\} = F_X(x)$$

**Summary:** From an $R \sim U_R(r)$ we obtain an $X \sim F_X(x)$
Sampling by the Rejection Method: the von Neumann Algorithm

- Given a pdf $f_X(x)$ limited in $(a,b)$, let

  $$h(x) = \frac{f_X(x)}{f_M}$$

  so that $0 \leq h(x) \leq 1, \forall x \in (a,b)$

- Operative Procedure:

  1. Sample $X' \sim U(a,b)$
  2. Sample $R \sim U[0,1]$

  If $R \leq h(X')$, then Accept $X'$.

  ![Diagram](image)
MC simulation for IDPSA
Scenario Generation entails

- generating random walks which guide the system from one configuration to another, at different times, for exploring (at best) all the possible failure scenarios.

During a trial, starting from a given system configuration $k'$ at $t'$, we need to determine

- *when* the next transition occurs
- *which* is the new configuration reached by the system as a consequence of the transition
- *what* is the state condition (safe/failed)

This can be done by assuming the probabilities governing the stochastic transitions.
Stochastic Transitions: Governing Probabilities

\[ T(t \mid t'; k') \, dt = \text{conditional probability of a transition at } t \in dt, \text{ given that the preceding transition occurred at } t' \text{ and that the state thereby entered was } k'. \]

\[ C(k \mid k'; t) = \text{conditional probability that the plant enters state } k, \text{ given that a transition occurred at time } t \text{ when the system was in state } k'. \]

Both these probabilities form the "transport kernel":

\[ K(t; k \mid t'; k') \, dt = T(t \mid t'; k') \, dt \, C(k \mid k'; t) \]

\[ \varphi(t; k) = \text{ingoing transition density or probability density function (pdf) of a system transition at } t, \text{ resulting in the entrance in state } k \]
Random walk = a realization of the system life generated by the underlying state-transition stochastic process.
The indirect approach consists in:

1. Sampling first the time $t$ of a system transition from the corresponding conditional probability density $T(t|t', k')$ of the system performing one of its possible transitions out of $k'$ entered at time $t'$.

2. Sampling the transition to the new configuration $k$ from the conditional probability $C(k|t, k')$ that the system enters the new state $k$ given that a transition has occurred at $t$ starting from the system in state $k'$.

3. Repeating the procedure from $k'$ at time $t$ to the next transition.
Direct Monte Carlo

The direct approach differs from the indirect one in that the system transitions are not sampled by considering the distributions for the whole system but, rather:

- directly samples the times of all possible transitions of all individual components of the system
- arranges the transitions along a timeline, in accordance to their times of occurrence
- Updates the timeline is after each transition occurs, to include the new possible transitions that the transient component can perform from its new state.

In other words, during a trial starting from a given system configuration $k'$ at $t'$:

1) We sample the times of transition $t_{j_i\rightarrow m_i}^i$, $m_i = 1, 2, \ldots, N_{s_i}$ of each component $i$, $i = 1, 2, \ldots, N_c$ leaving its current state $j'_i$ and arriving to the state $m_i$ from the corresponding transition time probability distributions $f_{T_{j_i\rightarrow m_i}}^i(t|t')$.

2) The time instants $t_{j_i\rightarrow m_i}^i$ thereby obtained are arranged in ascending order along a timeline from $t_{\text{min}}$ to $t_{\text{max}} \leq TM$. 
3) The clock time of the trial is moved to the first occurring transition time \( t_{\text{min}} = t^* \) in correspondence of which the system configuration is changed, i.e. the component \( i^* \) undergoing the transition is moved to its new state \( m_{i^*} \).

4) At this point, the new times of transition \( t_{m_{i^*} \rightarrow l_{i^*}}^i, l_{i^*}^s = 1, 2, ..., N_{S_i}^i \), of component \( i^* \) out of its current state \( m_{i^*} \) are sampled from the corresponding transition time probability distributions \( f_{t_{l_{i^*} \rightarrow m_{i^*}}}^i (t|t^*) \), and placed in the proper position of the timeline.

5) The clock time and the system are then moved to the next first occurring transition time and corresponding new configuration, respectively.

6) The procedure repeats until the next first occurring transition time falls beyond the mission time, i.e. \( t_{\text{min}} > TM \).

Compared to the indirect method, the direct approach is more suitable for systems whose components failure and repair behaviours are represented by different stochastic distribution laws

\[ \rightarrow \text{Direct Method is more suitable for IDPSA} \]

\( (i.e., \text{large number of components and large number of different reliability behaviors}) \)
MVL modelling of accidental scenarios
Dynamic reliability analysis (vs traditional PSA):

- components can fail at **ANY time** other than the initial time
- components can fail with **ANY magnitude** values other than the most conservative

Generation of scenarios is intractable within a continuous time and infinite states MC framework

Multi-Valued Logic Modeling of Accidental Scenarios
Multi-Valued Logic (MVL) in support of dynamic reliability analysis (vs traditional PSA):
- components (e.g., 4 components) can fail at discrete times other than the initial time
- components can fail with discrete magnitude values (e.g., 4 magnitudes) other than the most conservative
**Multi-Valued Logic** (MVL) in support of dynamic reliability analysis:
- components can fail at **discrete times** other than the initial time
- components can fail with **discrete magnitude** values other than the most conservative

For each accidental scenario, Monte Carlo fault injection engine:
- **random** components failure time $T(t \mid t'; k) \Delta t$
- **random** components failure magnitude $C(k \mid k'; t)$

![Diagram](image)
Example: U-Tube Steam Generator (UTSG)

Controlled Variable

Input variables

“High Level Failure mode”

Upper threshold

“Safe”

Lower threshold

“Low Level Failure mode”


i-th component of $N_c$ suitably connected components
Scenarios generation: Monte Carlo engine

i-th component of $N_c$ suitably connected components

Mission time

Failure time

Failure magnitude

Failure time  Failure magnitude  Failure order

2  3  [Blank]
Scenarios generation: Monte Carlo engine

i-th component of $N_c$ suitably connected components

Mission time

Failure time

Failure magnitude

$2 \rightarrow 3 \rightarrow 1 \rightarrow N_c$
Scenarios generation: Monte Carlo engine

\[ \overline{X}_n = (2, 3, 1, 3, \ldots, 4), \text{ generic } n\text{-th vector representative of scenario} \]
Dynamic accidental scenarios modeling
Dynamic Methodologies for IDPSA

- CET (Continuous Event Tree)
- CCMT (Cell-to-Cell Mapping Technique)
- DET (Dynamic Event Tree)
- MCDET (Monte Carlo Dynamic Event Tree)
Conventional safety assessment modelling methodologies (such as Event Trees and Fault Trees (ET/FT)) are extensively used as tools to perform reliability and safety assessment of complex and critical engineering systems.

Disadvantages:
- Analysis is based on few TH calculations (chosen from limiting cases by experts)
- Limited evaluation of the effects of the variability of system and operator responses
- Timing/sequencing of events and system dynamics is not explicitly accounted for in the analysis.

It is difficult to address:
- Variability of time & variability of operator strategies \( \rightarrow \) Alternative ways of succeeding exist
- Variability of system response
- Plant effect on crew performance and vice versa
- Interaction between them
Example: motorbike accident

Threshold

Safety Margin
CLASSICAL APPROACH

Speed $v$ is known, 100% braking force
$\rightarrow$ success if minimum distance (safety margin) $>> 0$
$\rightarrow$ failure if too late braking or bracking force $< 100\%$ or no action

DYNAMIC APPROACH

- What is the pilot doing?
- What does he see?
- What is his decision?
- What is the distribution of responses?
- If the driver acts on the braking system, what is the braking force?

A simulator calculates the speed at safety margin $= 0$ for different road conditions, tyres conditions and vehicle types,
$\rightarrow$ consequences for different hypotheses of crashes are calculated
Dynamic approaches

- attempt to integrate deterministic (simulator) and stochastic processes (degradation and failure event occurrences)
- Explicitly model the plant-crew interactions (the driver)
- Give variability of these interactions
- Model the evolution of the operator understanding
Dynamic methodologies for IDPSA

Continuous:  
CET (Continuos Event Tree)  
CCCMT (Continuous Cell-to-Cell Mapping Technique)

Discrete:  
CCMT (Cell-to-Cell Mapping Technique)  
DET (Dynamic Event Tree)  
MCDET (Monte Carlo Dynamic Event Tree)  
RET (Repairable Event Tree)  
RFT (Repairable Fault Tree)
Dynamic methodologies for IDPSA

Continuous: **CET** (Continuous Event Tree)
**CCCMT** (Continuous Cell-to-Cell Mapping Technique)
Continuous Event Trees (CET)

Describes the system behavior in terms of the probability $p_n(x,t)$ of finding the system in the state-space ($x$-space) with configuration $n$ at a given time $t$.

**Input:**
- Configuration transition rates $h(n|m, x, t) = \text{transition rates along the sequence of events}$
- Probability $F_n(x,t)$ that the system leaves configuration $n$ before time $t$
- Initial condition $p_0(x,0)$

**Output:**
$p_n(x,t)$

**Solution Method:**
Monte Carlo simulation
Continuous Cell-to-Cell Mapping (CCCMT)

Describes the system behavior in terms of the probability $\pi_i(j,t)$ of finding the system in the cell $j$ of the state-space ($x$-space) with configuration $i$ at a given time $t$.

- Derivable from CET

Input:
- Cell-to-cell transition probabilities $g(j|j',i',t)$ in the state space = transition rates along the sequence of events
- Configuration transition rates $F_i(j',t)$ that the system leaves configuration $i$ before time $t$
- Initial condition $\pi_0(j_0,0)$

Output: $\pi_i(j,t)$

Solution Method: Monte Carlo simulation
Dynamic methodologies for IDPSA

Discrete:  
- CCMT (Cell-to-Cell Mapping Technique)  
- DET (Dynamic Event Tree)  
- MCDET (Monte Carlo Dynamic Event Tree)  
- RET (Repairable Event Tree)  
- RFT (Repairable Fault Tree)
Cell-to-Cell Mapping (CCMT)

Describes the system behavior in terms of the probability $\pi_i(j,k)$ of finding the system in the cell $j$ of the state-space ($x$-space) with configuration $i$ at a given time $k$ ($k=0,1,...$).

- Derivable from CET

**Input:**
- Cell-to-cell transition probabilities $g(j'|j,i,k)$ in the state space
- Configuration transition rates $F_i(j',k)$ that the system leaves configuration $i$ before time step $k$
- Initial condition $\pi_0(j_0,0)$

**Output:**
$\pi_i(j,k)$

**Solution Method:**
Monte Carlo simulation
## Comparison

<table>
<thead>
<tr>
<th>Feature</th>
<th>CET</th>
<th>CCCMT</th>
<th>CCMT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solution Time</td>
<td>Long</td>
<td>Moderate</td>
<td>Moderate</td>
</tr>
<tr>
<td>Memory Requirement</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Programming Effort</td>
<td>High</td>
<td>Moderate</td>
<td>Low</td>
</tr>
</tbody>
</table>
AIM:

- identify branching points (nodes) in the system evolution (i.e. time-points along the simulation at which stochastic events occur)
  - A node is generated whenever a system, a component or an operator action is called for
  - Each branch represents a possible outcome of the stochastic event

- save the state of the system at each branching point
pursue the simulation of all branches

Simulating the scenario evolution from all branching points allows exploring all possible behaviors of the system parameters and process variables
DET: the interactive model

SYSTEM SIMULATOR
Physical model (mass, energy, momentum)

Equipment model (failure modes / repairs and probabilities)
Setpoints, equipment actuation
Event engine

Crew model (procedures, training)
Operator actions
Alarms generated
Monitor plant indicators
• Based on an user defined branching logic, driven by Probabilistic Density Functions (PDFs), an event occurs at a certain time instant.
• The simulation spoons n different branches. In each of them, the branching event determines a different consequence (including an associated probabilities)
• Each sequence continues until another event occurs and a new set of branching is spooned.
• The simulation ends when an exit condition or a maximum mission time is reached.
This leads to a more realistic and mechanistically consistent analysis of the system taken in consideration. Thus, the DET methodology (along with other dynamic PRA methods) is designed to take explicitly into account the timing of events which can become very important especially when uncertainties in complex phenomena are considered.

Starting from an initiating event, the main idea behind the DET methodology is to let a system code (i.e., RELAP5-3D, RELAP-7, etc.) determine the pathway of an accident scenario within a probabilistic “environment”.
Branching rules can be specified by users-input CDFs or by active components setpoints.
A practical example: route selection

Goal: get to Garibaldi train station *on time* for catching the train, from my office.
Dynamic constraints, different strategies, competing goals have to be jointly considered

- What was I doing in the office? *Dynamic constraint*
- How long would it take to switch off the PC? *Dynamic constraint*
- How long would it take to cross the campus? *Dynamic constraints & Different strategies*
- Is it hot or too cold? Is the pavement frozen (→ need of slow down the pace) or it is too hot to speed up the pace (→ need of slow down the pace)? *Competing goals*
- How long would it take to get to closest metro station? Is the traffic light green? *Dynamic constraints & Different strategies*
- Do I take the right metro to get to the station?
- Is the train at the same platform as usual?

Dynamic Event Tree

→ Different consequences for different hypotheses are calculated

Uncertainties
Alternative A

Map showing public transportation routes and times.
The PC takes too long to switch off

Traffic light is green

The train is at the same platform as usual

The train is at the same platform as usual

The pavement is frozen

The pavement is frozen

The pavement is frozen

The pavement is frozen
1. The number of scenarios that arise in the analysis is much larger than that of the classical FT/ET and thus not only the computational burden of the simulation is increased but also the a posteriori information retrieval becomes difficult.

2. Advantages: from the point of view of the completeness of the analysis and of the information content made available.
   
   • Identification of accident scenarios which may have been overlooked by the analyst when building the accident sequence models at the basis of the fault/event trees.
   • Conservative simplifying assumptions made by the analyst, for example on the evolution of some process parameters, can be relaxed as the process evolution is simulated directly by the underlying dynamics model.
   • Additional informative content becomes available as a result of the dynamic analysis, in the form of time-dependent probability density functions of components states and process parameters values.
Dynamic accidental scenarios post-processing
Failure Scenarios Characterization

**End State identification**
Fuzzy C-Means
Fuzzy Similarity

**Prime Implicants and Near Misses Identification**
Set Covering Problem (SCP)
Evolutionary algorithms
Clustering algorithms
Self Organizing maps
End State identification
Classifier
Similarity Matching
Classifier
A posteriori information retrieval

The number of the scenarios that are simulated is much larger than that of the classical ET/FT approaches.


Aim: Identify and group the scenarios of a dynamic safety assessment by characterizing the principal patterns of system evolution towards failure with an evolutionary Fuzzy C-Means (FCM) clustering algorithm.
Application: LBE-XADS

Nominal

Maximum diathermic oil temperature

- Upper threshold
- Lower threshold

"High-temperature failure mode"

"Safe"

"Low-temperature failure mode"

PID output stuck @ t=0

Feedforward control stuck @ t=0 AND failed communication between air coolers and actuators
Application: LBE-XADS

Failed communication between air coolers actuators and PID @ t=0

Air coolers stuck @ t=0

Feedforward control stuck @ t=0

“High-temperature failure mode”

“Low-temperature failure mode”

Maximum diathermic oil temperature

“Safe”

Upper threshold

Lower threshold
Application: LBE-XADS

- **Failed communication between air coolers actuators and PID @ t=0 AND Air coolers stuck**
  - PID output stuck @ t=0
  - Air coolers stuck @ t=0

- **Upper threshold**
  - “High-temperature failure mode”

- **Lower threshold**
  - “Low-temperature failure mode”
  - “Safe”
The method

\[ N + M \] DYNAMIC SCENARIO SIMULATIONS

RELEVANT FEATURES SELECTION

\[ N \] patterns labeled with known \( c \) classes

\[ M \] patterns of unknown class

TRAINING

(evolutionary optimization)

Optimal cluster centers

Optimal Mahalanobis Metric

TEST

Classification to \( i^{th} \) class or ambiguous

Classifier
Similarity Matching
Problem statement and methodology

On-line estimation of the Failure Mode in a developing accidental scenario, based on monitored signals related to its evolution

Similarity-based approach for failure mode classification

Library of reference trajectory patterns

Data from failure dynamic scenarios of the system

comparison

New developing accidental scenario

prediction

Failure Mode
Methodology: Similarity Matching

$n$-long developing test trajectory

- i-th reference trajectory
- Upper safety threshold
- Lower safety threshold

Monitored signal

Time x50 [s]
Prime Implicants and Near Misses Identification
Set Covering Problem (SCP)
Evolutionary algorithms
Clustering algorithms
Self Organizing maps
Integrated Deterministic and Probabilistic Safety Analysis

Prime Implicants (PI)
minimal combination of accident component failures that lead the system into failure

Safe Scenarios
sequences of failure events that keep the system into safe operational conditions

Possible scenarios
Integrated Deterministic and Probabilistic Safety Analysis

Prime Implicants (PI)
minimal combination of accident component failures that lead the system into failure

Near Misses
dangerous sequences of failure events that incidentally keep the system into safe but endangered and insecure operational conditions

Safe Scenarios
sequences of failure events that keep the system into safe operational conditions
Prime Implicants Identification
Set Covering Problem (SCP)
Evolutionary algorithms
A Set Covering Problem

Given:
- a universe $X$ of $n$ elements,
- a collection of subsets of $X$, $F = \{f_1, \ldots, f_k\}$,
- a cost function $c : f_i \rightarrow Q^+$,

find a **minimum-cost sub-collection** of $F$ that covers all elements of $X$

{f1, f3, f4} is a subset of F covering X
{f1, f2, f3, f4} is a subset of F covering X
{f2, f3, f4, f5} is a subset of F covering X

{f1, f3, f4} is a minimum cover set
Evolutionary optimization for PI identification

Differential Evolution (DE)

Component Y Failure Time

Component X Failure Time

Component Z Failure Time

Fitness Value

Feature 1

Feature 2

PI

Non-PI Accidental Scenario
Near Misses Identification
Clustering algorithms
Self Organizing maps
Clustering algorithms
Innovative risk-based characterization of the remaining scenarios:

\[ Risk(t) = \text{probability}(t) \cdot \text{consequence}(t) \]
Post-Processing: Near Misses Identification

Innovative **risk-based** characterization of the remaining scenarios:

\[
Risk (t) = probability (t) \cdot consequence(t)
\]

- Probability \((p(t))\)
- Consequence \((c(t))\)
- Risk \((r(t))\)

\[\text{increase as } N_{rl} \text{ moves further away from } N_{ref}\]
Innovative risk-based characterization of the remaining scenarios:

\[ \text{Risk}(t) = \rho(t) \cdot c(t) \]

**Probability function** \((\rho)\)

\[ \rho(t) = \varphi \left( \frac{N_{rl}(t) - (\mu + 5 \sigma)}{\sigma} \right) = \int_{-\infty}^{N_{rl}(t)} \frac{1}{\sqrt{2\pi} \sigma} e^{-\frac{(N_{rl}(t) - (\mu + 5\sigma))^2}{2\sigma^2}} dN_{rl} \]
Innovative risk-based characterization of the remaining scenarios:

\[ \text{Risk} (t) = p(t) \cdot c(t) \]

**Consequences** \( c(t) \)

\[ c(t) = (A \cdot 100) \frac{N_{rl}(t) - (\mu + 3\sigma)}{N_{rl}(t) - \mu} \]

- \( N_{hl} \), first alarm threshold
- \( N_{vh} \), second alarm threshold
- \( N_{high} \), Upper failure threshold

<table>
<thead>
<tr>
<th>Condition</th>
<th>Consequences</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N_{rl} &gt; N_{high} )</td>
<td>Catastrophic</td>
</tr>
<tr>
<td>( N_{rl} &gt; N_{vh} )</td>
<td>Medium consequences</td>
</tr>
<tr>
<td>( N_{rl} &gt; N_{hl} )</td>
<td></td>
</tr>
<tr>
<td>( N_{rl} \sim N_{ref} )</td>
<td>No consequences</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Elapsed time from IE (s)</th>
<th>1-1000</th>
<th>1001-2000</th>
<th>2001-3000</th>
<th>3001-4000</th>
</tr>
</thead>
</table>
Near Misses Identification: Unsupervised-Clustering Problem
Near Misses Identification: Unsupervised-Clustering Problem

- Unknown number of cluster \( k \)
- Unknown best set of features of \( p(t) \), \( c(t) \) and \( r(t) \) to be used for clustering the scenarios that are not PIs
Near Misses Identification: Results

Risk-based clustering results

✓ The best features are:
  1) The root mean square of $r(t)$ along 4000 (s)
  2) The standard deviation of $c(t)$ along 4000 (s)
  3) The standard deviation of $r(t)$ along 4000 (s)

✓ 5 clusters

Rule-based selection: The higher risk (1) and the longer the time of abnormal behavior ((2) and (3)), the more dangerous scenario
Near Misses Identification: Results

The diagram illustrates the RMS risk over time of abnormal behavior in seconds. Different markers and colors represent various clusters and time points. The x-axis represents the time of abnormal behavior in seconds, and the y-axis shows the RMS risk.

Legend:
- First cluster
- Second cluster
- Third cluster
- Fourth cluster
- Fifth cluster
Near Misses Identification: Results

332 NEAR MISSSES
Post-Processing Results

Prime Implicants scenarios

Near Misses scenarios

Safe Scenarios

Possible scenarios
On-Line Clustering: Risk-Based Clustering

- At each time $t$ we can compute:
  - The *residual* $r = N_{rl} - N_{ref}$
  - The *Hamming distance*
On-Line Clustering: Risk-Based Clustering

- At each time $t$ we can compute:
  - The residual $r = N_{rl} - N_{ref}$
  - The Hamming distance

- The developing scenarios is assigned to the cluster with the smallest Hamming distance
Conclusions

- The **reliability analysis** of dynamic systems calls for the **complementation of DSA and PSA into IDPSA**

- **A Complete IDPSA** of a has been shown: generation, modelling and post-processing

- **The Post-processing** analysis has been carried out for the identification of PI and Near Misses
  - The **Near Misses identification** problem has been solved with a **innovative risk-based clustering method**
  - The **characteristics** of the **Near Misses** scenarios have been identified solving a **Multi-Objective optimization problem**

- **The On-line analysis** has been performed for the **identification and prediction of accident progression** with **novel risk-based clustering method**