Advanced methods for loss-of-flow accident precursors identification in a superconducting magnet cryogenic cooling circuit

Advanced Methods for Loss-Of-Flow Accident Precursors identification in a Superconducting Magnet Cryogenic Cooling Circuit

Vincenzo Destino, Roberto Bonifetto, Nicola Pedroni, Laura Savoldi

NEMO group, Dipartimento Energia, Politecnico di Torino, Italy.
E-mail: vincenzo.destino@studenti.polito.it, {roberto.bonifetto, nicola.pedroni, laura.savoldi}@polito.it

Francesco Di Maio, Enrico Zio

Dipartimento di Energia, Politecnico di Milano, Italy. E-mail: {francesco.dimaio, enrico.zio}@polimi.it

Enrico Zio

MINES ParisTech, PSL Research University, CRC, Sophia Antipolis, France
Eminent Scholar at Kyung Hee University, Republic of Korea

In nuclear fusion systems, such as ITER, Superconducting Magnets (SMs) will be employed to magnetically confine the plasma. A Superconducting Magnet Cryogenic Cooling Circuit (SMCCC) must keep the SMs at cryogenic temperature to preserve their superconductive properties. Thus, a Loss-Of-Flow Accident (LOFA) in the SMCCC is to be avoided. In this work, a three-step methodology for the prompt identification of LOFA precursors (i.e., those component failures leading to a LOFA) is developed. First, accident scenarios are randomly generated by Monte Carlo sampling of the SMCCC components failures and the corresponding transient system response is simulated by a deterministic thermal-hydraulic code. In this phase, fast-running Proper Orthogonal Decomposition (POD)-based Kriging metamodels, adaptively trained to mimic the behavior of the detailed long-running code, are employed to reduce the associated computational burden. Second, the scenarios generated are grouped by a Spectral Clustering (SC) embedding the Fuzzy C-Means (FCM), in order to characterize the principal patterns of system evolution towards abnormal conditions (e.g., a LOFA). Third, an On-line Supervised Spectral Clustering (OSSC) approach is developed to assign signals measured during plant operation to one of the prototypical clusters identified, which may reveal the corresponding LOFA precursors (in terms of combinations of failed SMCCC components). The devised method is applied to the simplified model of a cryogenic cooling circuit of a single module of the ITER Central Solenoid. Results show that the approach developed timely identifies 95% of LOFA events and approximately 80% of the corresponding precursors.

Keywords: ITER Central Solenoid Magnet, Cryogenic Cooling Circuit, Loss-Of-Flow Accident, Precursors identification, Spectral Clustering, Adaptive Kriging meta-model

1. Introduction

ITER will be the first facility to produce a net amount of energy by means of thermonuclear fusion reactions occurring in a magnetically confined plasma of Deuterium and Tritium (ITER (2019)). Magnetic confinement is realized by different Superconducting Magnets (SMs) (Bigot (2019)): one Central Solenoid (CS), eighteen Toroidal Field (TF) coils, six Poloidal Field (PF) coils and eighteen Correction Coils (CCs). Each of the six Central Solenoid Modules (CSMs), which compose the CS, must sustain high currents (~45kA), in order to generate high magnetic fields (several T) to confine the plasma. In these challenging conditions, ohmic heating must be eliminated by guaranteeing the magnets superconducting properties (Takahashi et al. (2006)). These properties are preserved thanks to the cooling provided by a Superconducting Magnet Cryogenic Cooling Circuit (SMCCC), which is in charge of the extraction of the heat from the CSMs with Supercritical Helium (SHe) at 4.5 K and 0.5-0.6 MPa and of its transfer to pools of saturated Liquid Helium (LHe) (Mitchell et al. (2008)).

A Loss-Of-Flow Accident (LOFA) in the SMCCC is, thus, of major concern because it may impair the CS cooling capability. In this condition, the CS temperature and pressure may escalate rapidly due to the ohmic heating: if the temperature and the pressure exceed 150 K and 25 MPa, respectively, the CS structural integrity and superconducting properties could be lost (Takahashi et al. (2006)).

In this work, an automatic, three-step data-driven approach is elaborated to timely recognize patterns of signals measured during

Proceedings of the 30th European Safety and Reliability Conference and the 15th Probabilistic Safety Assessment and Management Conference
Edited by Piero Baraldi, Francesco Di Maio and Enrico Zio
Published by Research Publishing, Singapore.
ISBN: 978-981-14-8593-0; doi:10.3850/978-981-14-8593-0

plant operation and to identify LOFA precursors (in terms of failed SMCCC components). First, a "database" of accident scenarios is built by repeated Monte Carlo Sampling (MCS) of the SMCCC components failures and the simulation of the corresponding transient system response by the deterministic thermal-hydraulic Cryogenic Circuit Conductor and Coil (4C) code (Savoldi et al. (2010)). In this phase, fast-running Proper Orthogonal Decomposition (POD)-based Kriging metamodels (Marrel et al. (2014)) are trained by the Adaptive Kriging – Monte Carlo Sampling (AK-MCS) algorithm to mimic the behavior of the detailed long-running code and reduce the associated computational burden (Turati et al. (2018); Grishchenko et al. (2019)). In the second step, a Spectral Clustering (SC) embedding the Fuzzy C-Means (FCM) is used to cluster the scenarios thereby generated according to a measure of similarity between them, in order to characterize the principal patterns of the system evolution towards abnormal conditions (e.g., a LOFA). This allows identifying the prototypical component failures (i.e., the precursors) leading the system to abnormal conditions. Thirdly, the information collected is employed in an On-line Supervised Spectral Clustering (OSSC) to promptly assign *new developing* scenarios (measured during plant operation) to the identified groups and reveal the corresponding LOFA precursors (Di Maio et al. (2016)).

The remainder of the paper is organized as follows. In Section 2, a description of the SMCCC and of the thermo-hydraulic code employed to simulate its behavior is given. In Section 3, the method developed for LOFA precursors identification is presented. The approach is tested in Section 4, where the main results are shown. Finally, conclusions are drawn in Section 5.

2. The Superconducting Magnet Cryogenic Cooling Circuit (SMCCC)

Supercritical Helium (SHe) is circulated in the SMCCC to cool the six CSMs. Fig. 1 sketches a simplified scheme of the circuit, where only one CSM is connected to the cooling system. At nominal operational condition, the Centrifugal Pump (CP) guarantees a nominal flow $G_0 = 0.32\ kg/s$ in the two cryolines and a discharge pressure $p_0 = 0.42\ MPa$; Liquid Helium (LHe) at saturated condition ($T_{sat} = 4.5\ K$) is employed in the heat exchangers HX1 and HX2 to remove the heat produced in the CP and in the CSM, respectively; the Control Valves (CV1 and CV2) are Normally Open (NO), whereas the two Safety Valves (SV1 and SV2) and the By-pass Valve (BV) are Normally Closed (NC); controllers C1 and C2 receive signals from flow meters and pressure sensors, respectively.

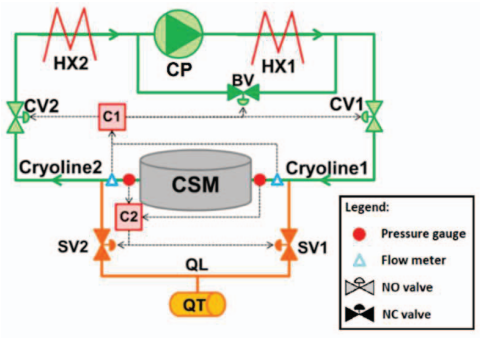

Fig. 1 Simplified SMCCC

In case of a LOFA, the coolant flow goes below 10% of the nominal value for more than the validation time ($\tau_{val} = 1s$) both at the CSM inlet and the CSM outlet (Savoldi et al. (2018)) and:

- C1 closes CV1 and CV2, opens BV preventing the CP damage and dumps the current inside the CSM in 30s: by doing so, the SHe flows only through the by-pass line, isolating the CSM and closing a loop on the CP to protect it from operation in surge condition.
- C2 opens the two SVs with a PID controller when the pressure in the CSM goes beyond $p_{lim} = 1.8\ MPa$, sending SHe in the Quench Tank (QT), at pressure $p_{QT} = 0.35\ MPa$ and temperature $T_{QT} = 300\ K$, to avoid pressurizing above 25 MPa and to guarantee the integrity of the connections to the CSM during quench.

The closed cooling circuit is simulated for a mission time $t_{miss} = 3600s$ with the 4C code (Savoldi et al. (2010)).

For each *i-th* simulation, $Z = 3$ variables $y_i^k(t)$ $[k = 1,2,\dots,Z]$ are monitored at time *t*: the pressure $p_{CSM,in}$ at the inlet of the CSM ($k = 1$), that must not exceed $p_{lim} = 1.8\ MPa$

to guarantee connections and headers integrity in the SMCCC; the hotspot temperature T_{hs} in the CSM ($k = 2$) and the ratio I/I_{cr} between the current flowing in the CSM conductors and the critical one ($k = 3$), that must not exceed the current sharing temperature $T_{cs} = 7.3\ K$ and $(I/I_{cr})_{lim} = 0.5$, respectively, in order to guarantee the superconducting properties of the CSM.

Failures of CP, CV1, CV2, BV, SV1 and SV2 may reduce the cooling capability of the SMCCC and lead any of the limits for $p_{CSM,in}$, T_{hs} and I/I_{cr} to be overcome.

3. The proposed framework for LOFA precursors identification

A method for the "*on-line*" characterization of accident scenarios developing in the SMCCC, based on the analysis of the $Z = 3$ time-varying monitored signals $p_{CSM,in}$, T_{hs} and I/I_{cr}, is here proposed in three steps.

3.1 Step 1: Creation of a "database" of simulated accidental scenarios

Each *i-th* accident scenario constituting the database is represented by a sequence of events, which is encoded by a vector of $M = 12$ elements $\boldsymbol{X}_i = [\, m_{CP}, \tau_{CP}, m_{CV1}, \tau_{CV1}, m_{CV2}, \tau_{CV2}, m_{BV}, \tau_{BV}, m_{SV1}, \tau_{SV1}, m_{SV2}, \tau_{SV2}]$ and generated by Monte Carlo Sampling (MCS): for each component, the magnitude (m) of the failure and the time (τ) at which the failure occurs are listed.

The *magnitude (m)* is assumed as follows:

- The magnitude of the CP can be a value between 0 and 4. If the component is not failed, $m_{CP} = 0$. Instead, m_{CP} values equal to 1, 2, 3 or 4 correspond to a reduction of the mass flow rate of 75%, 50%, 25% or 0% of the nominal value, respectively, due to a decrease of its rotational speed.
- The magnitude of NO valves (CV1 and CV2) can assume a value between 0 and 3. If the component works correctly, $m = 0$. Instead, if m is equal to 1, 2 or 3, the valve remains stuck opened, partially closed with a reduction of the flow section area of 50% or completely closed, respectively.
- For the magnitude of NC valves (BV, SV1 and SV2) there are four possible values, too. If the component is not failed, $m = 0$. Otherwise, if m is equal to 1, 2 or 3, the valve remains stuck closed, partially opened with the flow section area at 50% of the one completely opened or completely opened, respectively.

The *failure time (τ)* is a discrete value between 0s and 1800s (the discretization step is chosen equal to 0.01s). If the component works correctly, τ is null.

Once the vector $\boldsymbol{X}_i$ is sampled, it is fed as input to the 4C code, which delivers the three critical variables $y_i^k(t)\ [k = 1,2,3]$ and the mass flow rates, $G_{CSM,in}(t)$ and $G_{CSM,out}(t)$, at the inlet and the outlet of the CSM, respectively. When $G_{CSM,in}(t) < 0.032\ kg/s$ and $G_{CSM,out}(t) < 0.032\ kg/s$ for more than the validation time ($\tau_{val} = 1s$), a LOFA takes place and is detected by controller C1 (Savoldi et al. (2018)): the LOFA detection time is indicated as $t_{LOFA,C1,i}$.

Notice that each simulation of the system transient behavior by the 4C code requires on average two days on an Intel Core i3-7100 3.9GHz 3MB Cache. Thus, in this phase *fast-running* Proper Orthogonal Decomposition (POD)-based Kriging metamodels (Marrel et al. (2014)) are *trained* to mimic the behavior of the detailed long-running 4C code and to reduce the computational burden associated to the creation of the accident scenario "database".

3.1.1 Proper Orthogonal Decomposition (POD)-based Kriging metamodels

First, a training set of N_{krig} transients is generated by the Adaptive Kriging - Monte Carlo Sampling (AK-MCS) algorithm (Turati et al. (2018)). The aim of AK-MCS is to *preferably* include in the training set "interesting" scenarios lying in proximity of system failure configurations (i.e., LOFA conditions). Such scenarios are characterized by values of the critical variables close to (or exceeding) the system safety thresholds $\boldsymbol{Y}_{thr} = [Y_{thr}^1, Y_{thr}^2, Y_{thr}^3] = [p_{lim} = 1.8\ MPa, T_{cs} = 7.3\ K, (I/I_{cr})_{lim} = 0.5]$. This criterion is used to *adaptively* and *intelligently* drive the simulations towards "severe" scenarios and system conditions, without wasting computational time in the exploration of safe (not interesting) areas of the system state-space. Further details about the AK-MCS algorithm

are not reported here for brevity: the interested reader is referred to the cited references.

The resulting training set (enriched around the system failure region) is employed to construct POD-based kriging metamodels. In all generality, a POD - truncated at the H_k-*th* basis - applied to the generic *i-th* training scenario ($i = 1, \dots, N_{krig}$) is (Marrel et al. (2014)):

$$y_i^k(t) = \sum_{h=1}^{H_k} a_{ih}^k(\boldsymbol{X}_i) \cdot \varphi_h^k(t)\,, \qquad (1)$$

where $\varphi_h^k(t)$ (depending *only* on time t) is the orthogonal basis function of the *k-th* variable for the *h-th* base, and $a_{ih}^k(\boldsymbol{X}_i)$ (depending *only* on the input configuration $\boldsymbol{X}_i$) is its coefficient. Thanks to the orthogonality property, each coefficient $a_{ih}^k(\boldsymbol{X}_i)$ $(i = 1, \dots, N_{krig};\ h = 1, \dots, H_k)$ can be easily expressed as:

$$a_{ih}^k(\boldsymbol{X}_i) = \int_{t=0s}^{t=t_{miss}} y_i^k(t) \cdot \varphi_h^k(t) \cdot dt\,. \qquad (2)$$

In order to exploit this strategy for the fast simulation of *new* transients, the following procedure is carried out for each *k-th* critical output variable $y^k(t)$ $(k = 1,2,3)$:

Step 1) Matrix $\bar{\bar{Y}}^k[N_{krig}, L]$, containing the value y_{il}^k of the *k-th* variable of the *i-th* training scenario at the *l-th* time step, is built using the N_{krig} training scenarios (obtained by the AK-MCS simulations).

Step 2) Singular Value Decomposition (SVD) is used as a discretized version of POD to decompose $\bar{\bar{Y}}^k$:

$$\bar{\bar{Y}}^k = \bar{\bar{\Psi}}^k \cdot \bar{\bar{\Lambda}}^k \cdot \bar{\bar{\Phi}}^k\,, \qquad (3)$$

where $\bar{\bar{\Psi}}^k[N_{krig}, N_{krig}]$ and $\bar{\bar{\Phi}}^k[L, L]$ are matrixes that contain in their columns left-singular vectors and right-singular vectors, respectively, and $\bar{\bar{\Lambda}}^k[N_{krig}, L]$ is a diagonal matrix containing the nonnegative Λ_h^k singular values in decreasing order.

Step 3) The singular values Λ_h^k $(h = 1, \dots, N_{krig})$ are employed to identify the best number H_k of bases to use:

$$\Gamma_h^k = \frac{\sum_{j=1}^{h} \Lambda_j^k}{\sum_{h=1}^{N_{krig}} \Lambda_h^k}, \qquad (4)$$

where Γ_h^k indicates the percentage of the variability of the real N_{krig} transients that is "explained" by the POD decomposition truncated at *h-th* basis. In this work, H_k is the number of (ordered) basis for which Γ_h^k reaches a value of 0.99 (i.e., for which the POD decomposition "retains" the 99% of the total variability of the real transients). Matrix $\bar{\bar{\Phi}}^k[L, H_k]$ is then truncated at the H_k-*th* column: its generic element φ_{lh}^k corresponds to the value of the *h-th* orthogonal basis $\varphi_h^k(t)$ at *l-th* time step for output *k*.

Step 4) Matrix $\bar{\bar{A}}^k[N_{krig}, H_k]$ containing the coefficients a_{ih}^k is calculated as:

$$\bar{\bar{A}}^k = \bar{\bar{Y}}^k \cdot \bar{\bar{\Phi}}^{k^T}, \qquad (5)$$

with $\bar{\bar{\Phi}}^{k^T}[H_k, L]$ the transposal of $\bar{\bar{\Phi}}^k$. Eq. (5) is just a discretized form of Eq. (2).

Step 5) For each *h-th* base and *k-th* critical variable a Kriging metamodel $\mathcal{MM}_h^k$ is built. The training set is constituted by the system configurations $\bar{\boldsymbol{X}} = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \dots, \boldsymbol{X}_{N_{krig}}\}$ (inputs) and the corresponding basis coefficients $\boldsymbol{a}_h^k = \{a_{1h}^k, a_{2h}^k, \dots, a_{N_{krig}h}^k\}$ (outputs). In this way, for a new generic input configuration $\boldsymbol{X}$, Kriging metamodel $\mathcal{MM}_h^k$ is trained to predict the *coefficients* of the corresponding POD decomposition as $\hat{a}_h^k(\boldsymbol{X}) = \mathcal{MM}_h^k(\boldsymbol{X})$.

Step 6) The time-varying output $y_j^k(t)$, corresponding to a *new* generic (sampled) *j-th* system configuration $\boldsymbol{X}_j$, is then approximated at the *l-th* time step $(l = 1, \dots, L)$ by resorting to a discretized version of Eq. (1):

$$\tilde{y}_{jl}^k = \sum_{h=1}^{H_k} \hat{a}_h^k(\boldsymbol{X}_j) \cdot \varphi_{lh}^k\,, \qquad (6)$$

where $\tilde{y}_{jl}^k$ is the estimate of y_{jl}^k, resulting from the metamodel-based POD.

The same procedure is also adopted to approximate $G_{max,j}(t)$, i.e., the maximum between the helium mass flow rates at the inlet and outlet of the CSM, (see Eq. (7)), which is here employed in the estimation of $t_{LOFA,C1,j}$ (i.e., of the time when a LOFA occurs and is detected by C1) for the *j-th* scenario:

$$G_{max}(t) = \max\left(G_{CSM,in}(t), G_{CSM,out}(t)\right). \qquad (7)$$

The approximation of $G_{max,j}(t)$ by POD-based Kriging metamodels for a generic *j-th* scenario at *l-th* time step is indicated as $\tilde{G}_{max,lj}$. Then, $\tilde{t}_{LOFA,C1,j}$ (i.e., the POD metamodel-based approximation of $t_{LOFA,C1,j}$) is the time where $\tilde{G}_{max,lj} < 0.032\ kg/s$ for more than the validation time $\tau_{val} = 1s$ (thus, mimiking the operation of the controller C1).

Thus, by resorting to multiple Kriging metamodels *replacing* the 4C code, the computational time per simulation is sharply reduced and a very large number N_{POD} (typically, $N_{POD} \gg N_{krig}$) of new scenarios can be simulated and included in the "database".

3.2 Step 2: Identification of prototypical transients and components failure modes

From the previous step a database is available constituted by N_{krig} and N_{POD} scenarios. Relying on Spectral Clustering (SC), the $N_{data} = N_{krig} + N_{POD}$ transients are classified in C clusters (Von Luxburg (2007)) (Section 3.2.1). Then, each *c-th* cluster is post-processed to extract its main features, in terms of prototypical time evolutions towards failure and of the corresponding component failures (i.e., the accident precursors) (Section 3.2.2).

3.2.1 Spectral Clustering (SC) embedding the Fuzzy C-Means (FCM)

SC lets N_{data} objects (i.e., the transients of the database) to be classified in C clusters through a similarity measure between them. Each similarity is here calculated using the three critical variables $y_i^k(t)\ [k = 1,2,3]$ of L duration and collected in a similarity matrix $\bar{\bar{W}}$, from which the Normalized Laplacian matrix $\bar{\bar{L}}_{sym}$ is calculated. Features needed to classify the N_{data} object are extracted from $\bar{\bar{L}}_{sym}$ and fed to the FCM code (Baraldi et al. (2015)). The algorithm produces two matrices: i) matrix $\bar{\bar{\mathcal{A}}}[C, C]$ containing in each *c-th* row the eigenspace coordinates of the *center* of the *c-th* cluster; ii) a matrix $\bar{\bar{\mathcal{M}}}[C, N_{data}]$ whose generic element is the M_{ci} membership degree of the *i-th* transient of the database with respect to the *c-th* cluster: a transient belongs to a given cluster, if the corresponding membership exceeds a certain limit ($M_{lim} = 0.7$).

3.2.2 Post-processing of the clusters to identify the LOFA occurrence times and the component failure modes

The $N_{data,c} < N_{data}$ scenarios belonging to each *c-th* cluster are post-processed to evaluate the following quantities: $P_{LOFA,l}(c)$ (8), representing the probability that a LOFA is occurred at the *l-th* time step in a system configuration belonging to cluster *c*; and $P_{FAIL,l}(e|c)$ (9), representing the probability that component *e* is failed at the *l-th* time step, given that a LOFA has occurred due to a system configuration of cluster *c*.

$$P_{LOFA,l}(c) = = \frac{\sum_{i=1}^{N_{data,c}} \Theta(t_{LOFA,C1,i} \epsilon [t_o, t_{o+1}])}{N_{data,c}} \quad (8)$$

$$P_{FAIL,l}(e|c) = = \frac{\sum_{i=1}^{N_{data,c}} \Theta(\tau_{e,i} \epsilon [t_o, t_{o+1}] \ \wedge \ m_{e,i} \neq 0)}{N_{data,c}} \quad (9)$$

$$\text{with} \qquad \Theta(x) = \begin{cases} 0 & \text{if } x \textit{ is} \text{ false} \\ 1 & \text{if } x \text{ is true} \end{cases}. \quad (10)$$

Here, $[t_o, t_{o+1}]$ is the time interval (bin) including the *l-th* time step; $\Theta(x)$ (Eq. (10)) is an indicator function used to count the events of interest during that interval, i.e., the number of LOFA occurrences in Eq. (8) and the number of *e-th* component failures ($m_e \neq 0$) in Eq. (9). Notice that the length of the *o-th* interval (bin) $[t_o, t_{o+1}]$ is here selected equal to 300s, as a satisfactory compromise between analysis detail and statistical robustness: such intervals should be small enough to provide a fine description of the time evolution of the probabilities, but large enough to include a statistically meaningful number of samples.

3.3 Step 3: On-line Supervised Spectral Clustering (OSSC) for timely LOFA precursors identification

An On-line Supervised Spectral Clustering (OSSC) method is trained with the available N_{data} scenarios to timely identify LOFA precursors during the development of a new *j-th* accident scenario, different from the training ones. The *j-th* scenario, characterized by the monitored variables $y_j^k(t)\ (k = 1, 2, \dots, Z = 3)$, is *compared* to those of the created database $y_i^k(t)\ (i = 1, 2, \dots, N_{data})$ at each *l-th* time:

Step 1) The *k-th* trajectory $y_j^k(t)$ is recorded every $\Delta t = 0.01s$ time step from 0s to 3600s, obtaining values y_{jl}^k $(l = 1,2, \dots, L)$ at each *l-th* time step. Thus, $L = 360001$ points for each *k-th* variable are stored.

Step 2) The value y_{jl}^k $(l = 1,2, \dots, L; k = 1, \dots, N_k)$ is normalized in the interval [0.2, 0.8] and saved as $y_{n,jl}^k$.

Step 3) The Euclidean pointwise distance $\delta_{l,ji}$ between the *j-th* new scenario and the *i-th* training scenario $(i = 1,2, \dots, N_{data})$ at the *l-th* time step $(l = 1,2, \dots, L)$ is evaluated as:

$$\delta_{l,ji} = \sum_{k=1}^{Z} \sum_{p=1}^{l} \left| y_{n,jp}^k - y_{n,ip}^k \right|. \quad (11)$$

Step 4) The similarity indices $w_{l,ji}$ are calculated at each *l-th* time step as $w_{l,ji} = e^{-F \cdot \delta_{l,ji}^2}$, with $F = 1.7 \cdot 10^{-9}$ (Di Maio et al. (2016)). The higher $w_{l,ji}$, the higher the similarity between the *j-th* test scenario and the *i*-th training scenario until the *l-th* time step.

Step 5) The similarity indices $w_{l,ji}$ are fed to the FCM to obtain membership $M_{l,cj}$ $(c = 1, \dots, C)$, which measures the "degree" of confidence with which the *j-th* scenario at the *l-th* time step "belongs" to the *c-th* cluster.

Step 6) The pointwise difference $M_{rel,l,cj}$ between $M_{l,cj}(c = 1,2, \dots, C)$ and $M_{l,c0}$ (i.e., the membership to the *c-th* cluster at *l-th* time respect of a scenario at nominal conditions with no failures) is calculated. In this way, the background contribution of a standard scenario is removed from the trend of the membership of the *j-th* test transient to cluster *c*.

Step 7) Calculate $V_{rel,l,cj}$ $(c = 1,2, \dots, C)$, i.e., a discrete estimator of the derivative of the membership $M_{l,cj}$ at the *l-th* time:

$$V_{rel,l,cj} = = \begin{cases} 0 & if\ l = 1 \\ \dfrac{M_{rel,l,cj} - M_{rel,(l-1),cj}}{\Delta t} & if\ l \neq 1 \end{cases}. \quad (12)$$

Step 8) LOFA precursors are identified as follows:

Step 8a) Each $V_{rel,l,cj}$ $[c = 1, \dots, C]$ is compared to a limit threshold $V_{lim,LOFA,l}$. In extreme synthesis, if at least two values of $V_{rel,l,cj}$, namely $V_{1st} = \max_c(V_{rel,l,cj})$ and $V_{2nd} = \max_{c \neq c_{1st}}(V_{rel,l,cj})$, overcome the threshold $V_{lim,LOFA,l}$ at time *l*, then indicator $P_{LOFA,l,j}$ is assigned a value (different from zero) as:

$$P_{LOFA,l,j} = \frac{P_{LOFA,l}(c_{1st,l}) + P_{LOFA,l}(c_{2nd,l})}{\sum_c^C P_{LOFA,l}(c)}. \quad (13)$$

Notice that Eq. (13) indicates the probability that a LOFA has happened at time *l*, due to a system configuration of either cluster $c_{1st,l}$ or $c_{2nd,l}$. Finally, if $P_{LOFA,l,j}$ (13) exceeds a limit threshold $P_{LOFA,lim}$, the algorithm identifies the LOFA precursors (see step 9b. below); otherwise, no precursor is identified.

Notice that thresholds $V_{lim,LOFA,l}$ and $P_{LOFA,lim}$ are properly determined and tuned from the training data, in order to minimize the sum of false positives (i.e., not occurred failure events wrongly identified as occurred) and false negatives (i.e., occurred events not identified) encountered in the LOFA identification process *on the training scenarios*. Further algorithmic details are not reported here for brevity.

Step 8b) Quantities $f_{Cl,l,j}(c)$ and $P_{Cl,l}(c)$ are calculated by Eq. (14) and Eq. (15), respectively, for each *c-th* cluster. The former assumes a value equal to $P_{LOFA,l}(c)$ [see Eq. (8)], only if the corresponding membership $M_{rel,l,cj}$ overcomes the threshold value $M_{lim,FAIL,l}$ (see Eq. (17) below); otherwise, it is set to 0. The latter indicates the degree with which cluster *c* is "responsible" for the failure (i.e., the probability that a system configuration of cluster *c* is responsible for the LOFA): in fact, Eq. (15) represents a "re-normalization" of Eq. (8), conditional to the fact that a LOFA has occurred and has been identified.

$$f_{Cl,l,j}(c) = = \begin{cases} P_{LOFA,l}(c) & \text{if } M_{rel,l,cj} > M_{lim,FAIL.l} \\ 0 & \text{if } M_{rel,l,cj} \leq M_{lim,FAIL.l} \end{cases} \quad (14)$$

$$P_{Cl,l}(c) = \frac{f_{Cl,l,j}(c)}{\sum_c^{C=9} f_{Cl,l,j}(c)} \quad (15)$$

Then, $P_{FAIL,l,j}(e)$ is computed with Eq. (16) (Theorem of Total Probability) for each *e-th* component: it represents the unconditional probability that in the *j-th* test scenario component *e* is failed at time *l*.

$$P_{FAIL,l,j}(e) = \\ = \sum_{c}^{c=9} P_{FAIL,l}(e|c) \cdot P_{Cl,l,j}(c) \qquad (16)$$

Finally, if $P_{FAIL,l,j}(e)$ overcomes the threshold $P_{FAIL,lim}(e)$, then the *e-th* component is considered failed (i.e., it is identified as a precursor).

Notice that the trend of $M_{lim,FAIL.l}$ and the values of each $P_{FAIL,lim}(e)$ are determined from the N_{data} training scenarios, too. In particular, $M_{lim,FAIL,l}$ is calculated as

$$M_{lim,FAIL.l} = \mathcal{S} \cdot (l^{th}\ time). \qquad (17)$$

It is worth mentioning that $M_{lim,FAIL,l}$ is assumed to be linearly dependent on time, because δ_{ij} (11) increases from $t = 0s$ to $t = t_{miss} = 3600s$ and it is used to calculate $M_{rel,l,cj}$. $\mathcal{S}$ is the value that maximises the number of training scenarios whose components failures are correctly identified as LOFA precursors, minimizing at the same time the time delay between components failures and the time of precursors identification. Instead, for each *e-th* component, $P_{FAIL,lim}(e)$ is set to minimize the sum between false positive and false negatives, related to the identification of the *e-th* component as precursor over the N_{data} training scenarios.

4. Results

4.1 Step 1: Creation of a "database" of simulated accidental scenarios

The AK-MCS procedure has been iterated to produce a set of $N_{krig} = 119$ scenarios, used to train the POD-based Kriging metamodels (Section 3.1). These have allowed simulating $N_{POD} = 700$ *new* time-varying transients with a sharp reduction in the computational burden with respect to the 4C code (from an average of two days to about 1.1s per simulation). The resulting (new) database is thus constituted by $N_{data} = 819$ scenarios.

4.2 Step 2: Identification of prototypical transients and components failure modes

The $N_{data} = 819$ scenarios have been grouped into $C = 9$ clusters by SC embedding FCM (Section 3.2). Each cluster is characterized by different values of $P_{LOFA,l}(c)$ (8) and of $P_{FAIL,l}(e|c)$ (9), sketched in Figs. 2 and 3, respectively, with reference to cluster $c = 4$ only for illustration purposes.

According to Fig. 2, for system configurations belonging to cluster $c = 4$ the probability of LOFA is significant (i.e., around 30-50%) in time interval [0s, 900s], whereas it is negligible in [900s, 1800s].

The conditional failure probability $P_{FAIL,l}(e|c)$ (9) of each component e for cluster $c = 4$ as a function of time l is reported in Fig. 3, as an example. It clearly shows some relevant information: if a LOFA occurs in a system configuration of cluster 4, it will be due with high probability to an "early" failure (i.e., within [0,600s]) of either CP or CV1 or CV2 or BV (or combination of such components).

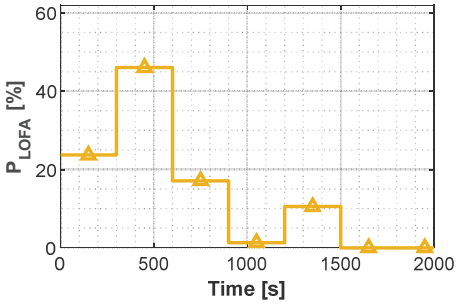

Fig. 2 Probability of the LOFA occurrence in time $P_{LOFA,l}(c)$ (8) for cluster c = 4

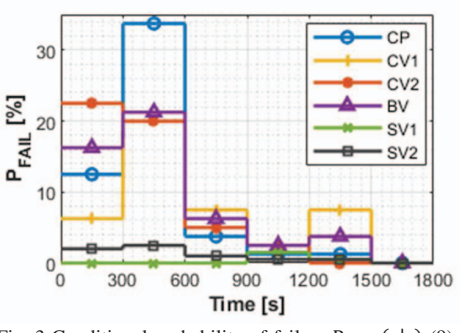

Fig. 3 Conditional probability of failure $P_{FAIL,l}(e|c)$ (9) of each component e for cluster c=4, as a function of time l

4.3 Step 3: OSSC for timely LOFA precursors identification

In Tab. 1, the results of the application of the LOFA precursors identification algorithm on $N_{test} = 38$ new test scenarios (different from the training ones) are summarized.

Table 1 LOFA identification results on $N_{test} = 38$ scenarios

Scenarios with LOFA	32
LOFA predicted in advance	24
LOFA not predicted in advance	8
Scenarios with NO LOFA	6
Correct identification NO LOFA	4
False positive LOFA	2

It can be seen that in general, a LOFA is predicted in advance in most scenarios, with a negligible number of "false positives" in scenarios without a LOFA: globally, the elaborated method recognized 95% of the LOFA events and it is able to predict them in advance in the 75% of the cases. In Tab. 2 the results of precursors identification are reported.

Table 2 Results of the precursor identification approach for the 32 test scenarios with LOFA

	Correct precursor ID	False NEG.	Correct ID of normal operation	False POS.
CP	21	2	2	7
CV1	16	0	2	14
CV2	13	2	9	8
BV	14	2	3	13
SV1	1	1	25	5
SV2	2	2	20	8

About 80% of the precursors are identified correctly by the OSSC algorithm, despite the relatively large number of false positives (i.e., 56%) for components CP, CV1 and BV: this however does not endanger the SMCCC, because *conservatively overestimating* the number of failed components (and, thus, the risk associated to the system). On the other side, it reduces its *availability* (due, e.g., to unnecessary inspections following the precursors identification).

5. Conclusions

A computational framework for LOFA precursors identification in a superconducting magnet cryogenic cooling circuit has been developed. The approach has been tested on 38 accidental scenarios to verify its robustness. Results have shown that 95% of the scenarios are correctly classified as "safe" or "faulty" and 80% of LOFA precursors are correctly identified. In this light, the proposed method may be employed to guide *prioritization actions* for *inspection* and *maintenance* of the SMCCC components. Also, notice that thanks to the use of metamodels, these satisfactory results have been obtained at the expense of *very few* (i.e., around 120) runs of the detailed, long-running simulation code.

References

Baraldi, P., F. Di Maio, M. Rigamonti, E. Zio and R. Seraoui (2015). Clustering for unsupervised fault diagnosis in nuclear turbine shut-down transients. *Mech Syst Signal Process 58–59*, 160-178.

Bigot, B. (2019). ITER construction and manufacturing progress toward first plasma. *Fusion Eng Des 146*, Part A, 124-129.

Di Maio, F., M. Vagnoli and E. Zio (2016). Transient identification by clustering based on Integrated Deterministic and Probabilistic Safety Analysis outcomes. *Ann. Nucl. Energy 87-2*, 217-227.

ITER (2019). ITER - the way to new energy [Online] - http://www.iter.org/.

Grishchenko, D., S. Galushin, P. Kudinov (2019). Failure domain analysis and uncertainty quantification using surrogate models for steam explosion in a Nordic type BWR. *Nuc Eng Des 343*, 63-75.

Marrel, A., N. Pérot and C. Mottet (2014). Development of a surrogate model and sensitivity analysis for spatio-temporal numerical simulators. *Stoch Environ Res Risk Assess 29-3*, 959-974.

Mitchell, N., D. Bessette, R. Gallix, C. Jong, J. Knaster, P. Libeyre, C. Sborchia and F. Simon (2008). The ITER magnet system. *IEEE Trans. Appl. Supercond. 18-2*, 435-440.

Savoldi, L., F. Casella, B. Fiori and R. Zanino (2010). The 4C Code for the Cryogenic Circuit Conductor and Coil modelling in ITER, *Cryogenics 50-3*, 167-176.

Savoldi, L., R. Bonifetto, N. Pedroni and R. Zanino (2018). Analysis of a protected Loss Of Flow Accident in the ITER TF coil cooling circuit. *IEEE Trans. Appl. Supercond. 28-3*, 1-9.

Takahashi, Y., K. Yoshida, Y. Nabara, M. Edaya and N. Mitchell (2006). Simulation of Quench Tests of the Central Solenoid Insert Coil in the ITER Central Solenoid Model Coil. *IEEE Trans. Appl. Supercond. 16-2*, 783-786.

Turati, P., A. Cammi, S. Lorenzi, N. Pedroni and E. Zio (2018). Adaptive simulation for failure identification in the Advanced Lead Fast Reactor European Demonstrator. *Progress in Nuclear Energy 103*, 176-190.

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Stat Comput 17*, 395–416.