

Data-driven reduced order modelling with malfunctioning sensors recovery applied to the Molten Salt Reactor case

Stefano Riva¹, Carolina Introini¹, Enrico Zio^{1,2}, and Antonio Cammi^{3,1,*}

¹ Politecnico di Milano, Energy Department- Nuclear Engineering Division, 20156 Milano, Italy

² MINES Paris, PSL University, CRC, Sophia Antipolis, France

³ Emirates Nuclear Technology Center (ENTC), Department of Mechanical and Nuclear Engineering, Khalifa University, Abu Dhabi 127788, United Arab Emirates

Received: 27 May 2025 / Received in final form: 6 August 2025 / Accepted: 7 August 2025

Abstract. This work presents the use of two Data-Driven Reduced Order Modelling techniques in predicting the transient response of a Molten Salt Fast Reactor when one or more sensors fail and, thus, provide wrong information; Supervised Machine Learning techniques are used to compensate for the failed sensors. Data-Driven Reduced Order Modelling integrate the physical knowledge contained in high-fidelity mathematical models with that coming from data measured on the actual system. This enables refining and updating the mathematical model, and address the challenges related to local-only observations, allowing for global state estimation. These methods are of interest when both sources of information are present, albeit incomplete, as is the case of the Molten Salt Fast Reactor. In these designs, typically operating in the fast neutron spectrum, the fuel is liquid, and no solid structures are foreseen in the core, thus making sensing and monitoring of safety-critical parameters and quantities quite challenging. Additionally, most literature studies on Data-Driven Reduced Order Modelling take the experimental observations as (noisy) *ground-truth*: very few works consider the case in which sensor fail or malfunction, and how this affect the state estimation.

1 Introduction

When analysing the behaviour of complex engineering systems, and even more so for safety-critical applications such as those of nuclear reactor engineering, the state-of-the-art approach is to use physics-based high-fidelity numerical models, as accurate as possible. Despite the advancements in computational hardware, the drawback becomes the computational cost of these models, both in terms of storage (and data-sharing) and simulation time; a cost that becomes even more unsustainable for online control and dynamic monitoring. To address this drawback, several algorithms for Reduced Order Modelling (ROM) have been developed [1,2]. The literature on this topic is quite abundant, and the authors of this work have focused mainly on Reduced Basis approaches [3]. Briefly, given a dataset of solutions of the Full Order Model (FOM), typically called *snapshots*¹, through this approach, one can extract the fundamental modes, which describe the dominant spatial physics within the dataset [4]. These modes, then, define a surrogate space, whose dimension is much

smaller than the degrees of freedom of the starting FOM, upon which a reduced model is built. This surrogate model is now much more efficient, in terms of computational burden, compared to the FOM, at the cost of some accuracy: however, it is possible to tune the dimension of the surrogate space to achieve the desired accuracy level [2]. Its uses span from inverse problems like parameter estimation, that is, retrieving physical parameter values characterising transients not included in the starting dataset without the need of solving the FOM again, and forecasting of the future state of the system. Additionally, optimisation problems such as those within Data Assimilation (DA) algorithms [4] can now be solved on the reduced space, with a significant reduction in complexity and computational cost: as, traditionally, the optimisation problem was the bottleneck for most state-of-the-art variational DA methods [5], their integration with ROM techniques looks very promising.

For this purpose, the authors of the present work have defined a novel framework, called Data-Driven Reduced Order Modelling (DDROM) [6], a branch of Scientific Machine Learning [7] fusing physically consistent data and governing equations with data-driven machine learning methodologies. This framework aims to update and

* e-mail: antonio.cammi@polimi.it

¹ These are usually spatio-temporal/parametric data.

correct the background mathematical model using observations (either from experimental sensors or some higher-fidelity model) and to select the optimal positioning of sensors to maximise the amount of information collected [8,9]. Most ROM techniques can be adapted within the DDROM framework: the present work restricts the focus on two of them, the Generalised Empirical Interpolation Method (GEIM) [9] and the Parameterised-Background Data-Weak (PBDW) formulation [8], already tested for nuclear reactor engineering problems [10–14]. Although these studies consider noisy measurements, proposing also regularisation techniques if needed [15,16], all of them assume that the data coming from the sensors are the *ground-truth*, implicitly assuming that a sensor cannot malfunction nor it can fail; thus, there is no consideration regarding the robustness of the DDROM algorithms and their performance in presence of malfunctioning sensors.

However, faulty sensors imply a faulty ground-truth, meaning that the DDROM algorithm is fed wrong reference values; in an engineering system, sensors may fail or malfunction for multiple reasons, and any methodology that relies on measurements should implement corrective actions to retain its reliability. Therefore, to use the DDROM framework outside laboratory test cases, it becomes mandatory to assess how its algorithms perform in the presence of malfunctioning or faulty sensors, to discriminate between working and faulty sensors and to determine strategies to retrieve the missing information [17]. Available literature on this topic, to the best of the authors' knowledge, is quite scarce: it is worth mentioning the work of [18], which developed a framework for 'robust data assimilation' by changing the traditional norm formulation from L_2 to L_1 to dynamically adjust the importance weight of the data according to their deviation from the mean forecast, the work of [19], in which the missing values of incomplete sensors are approximated with the Gappy-POD technique, and the work of [20], which developed a robust Dynamic Mode Decomposition by suppressing outliers in the dataset; there has not been an in-depth investigation on the performance of GEIM and PBDW methods in presence of faulty sensors [21]. Regarding the implementation, this work follows a companion approach as [19], adding another building block to the DDROM framework by implementing Machine Learning (ML) techniques to retrieve the missing information (or correct the wrong information) of the faulty sensor: more in detail, Gaussian Process Regression (GPR) is used [22]. Additionally, this work also includes a preliminary analysis on faulty sensor recognition, both by adopting a Random Forest algorithm [23] for a priori classifying and labelling sensor faults within the training dataset, and by leveraging the trained GPR to identify unexpected signals coming from the system in quasi-real-time, and thus automatically substituting the faulty measure with the GPR prediction without the need of knowing a priori which sensor has failed.

As a test case, this work builds from the results obtained in [14], which applied DDROM algorithms to the study of an accidental transient scenario of the European Molten Salt Fast Reactor [24] for online monitoring in presence of noisy measurements, introducing the possi-

bility of having one or more malfunctioning sensors under the assumption that a single (unknown) sensor can fail at the beginning of the transient. Regarding sensor positioning, the premise of this work is to assess whether auxiliary sensors, located in 'safer' (from the point of view of the sensor failure probability) regions of the domain can be used to recover the missing information from a failed sensor located in a safety-critical region of the domain, characterised by much higher sensor failure probability due to the harsher environment conditions: for the MSFR, the two regions are, respectively, the solid outside reflector and the liquid core².

Section 2 includes a brief presentation of the DDROM framework and the two algorithms (GEIM and PBDW) used in this work; Section 3 then describes the strategy for sensor recovery. Section 4 reports the main results, first assessing the performance of the algorithms in the presence of a malfunctioning sensor, exploring two different recovery strategies and discussing autonomous faulty sensor classification and recognition. Finally, Section 5 summarises the key findings of the study and reports some future perspectives regarding this research topic.

2 Data-driven reduced order modelling

Reduced Order Modelling techniques aim to reduce the computational complexity of high-fidelity numerical models by efficiently compressing the information while maintaining the desired level of accuracy. Being based on ROM techniques, the DDROM framework shares the typical offline-online decomposition [1], or training/testing decomposition typical of ML approaches, with the additional step of optimal sensor placement algorithms to include experimental measurements. Figure 1 shows the scheme of the DDROM framework [6]. The DDROM framework, then, combines the background knowledge in the mathematical model with the information retrieved from the experimental sensors. Compared to stand-alone ROM methods, the accuracy of the FOM no longer bounds the accuracy of the state estimate/reconstruction; by integrating measurements of the fields of interest, the latter can be improved by including unforeseen uncertainties and non-modelled physics [6,16,25].

- In the **offline phase**, the FOM is solved several times for different values of the model parameters $\boldsymbol{\mu} \in \mathcal{D} \subset \mathbb{R}^p$ (including time and/or thermo-physical properties, input parameters, boundary conditions): each full-order solution or *snapshot* $u_{FOM}(\mathbf{x}; \boldsymbol{\mu}_n) \in \mathbb{R}^{\mathcal{N}_h}$, which depends on the spatial coordinate and the vector of parameters $\boldsymbol{\mu}$ within the training dataset Ξ_{train} , is then collected and stored. From this dataset, through a dimensionality compression technique of choice, it is possible to extract a set of *basis functions* $\{\psi_n(\mathbf{x}) \in \mathbb{R}^{\mathcal{N}_h}\}_{n=1}^N$, whose physical interpretation depends on the reduction algorithm used; these

² For the purpose of this work, it is assumed that sensors of any quantity of interest can be located inside the liquid core, although this scenario, given the current design, is not fully realistic.

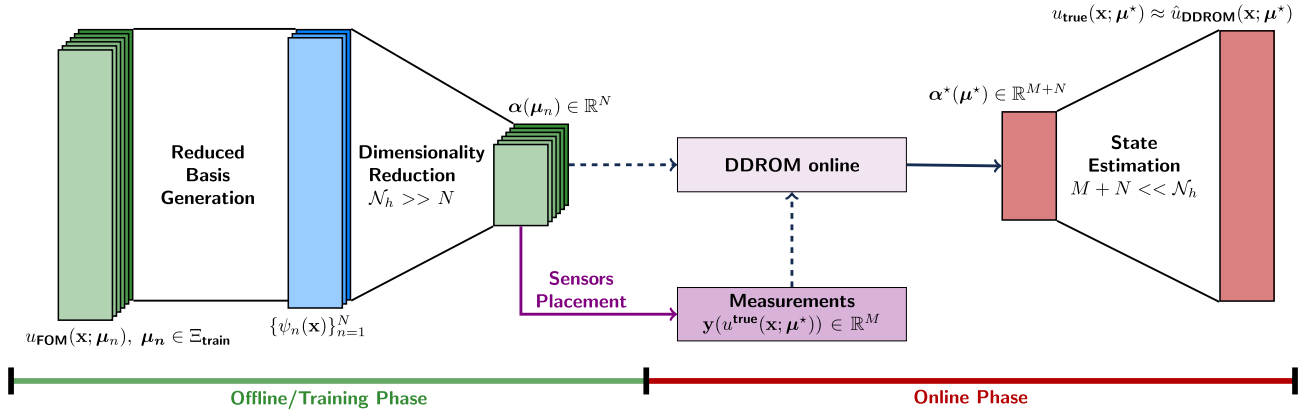


Fig. 1. Schematic of the DDROM framework, highlighting the offline-online decomposition: from a series of snapshots, the ROM algorithm extracts the fundamental modes, performing dimensionality reduction and building the DDROM model; in the online phase, measurements are given as input to the reduced model to obtain an updated state estimation. Taken from [21].

basis functions span a linear subspace of the solution space, and they embed the fundamental spatial dependence of the training dataset. The FOM is, then, projected onto this surrogate space, whose dimension N is much smaller than the degree of freedoms \mathcal{N}_h of the FOM, to build a Reduced Order Model, from which the modal coefficients $\{\alpha(\boldsymbol{\mu}_n)\}_{n=1}^N$, which embed the parametric dependence (including time) of the training dataset, can be retrieved. Finally, the problem of finding the optimal positioning of sensors within the physical system to maximise the amount of information that can be extracted and solved in the reduced space, significantly reducing its computational cost: this implies searching for the optimal configuration within a large space of possible combinations, but the costs associated to this task can be greatly reduced through the aforementioned reduction process.

- In the **online phase**, measurements $\mathbf{y}(u_{true}(\mathbf{x}, \boldsymbol{\mu}^*)) \in \mathbb{R}^M$ are collected from aptly-positioned sensors to retrieve an *augmented* state estimate $\hat{u}_{DDROM}(\mathbf{x}, \boldsymbol{\mu}^*)$ that accounts for both source of information. The ROM is solved quickly and accurately due to its much lower dimension. Finally, the output of the reduced model can be decoded back to the full state of the system, approximating the true solution.

Regarding the selected reduction techniques, this work adopts the Generalised Empirical Interpolation Method and the Parameterised-Background Data-Weak formulation, both have been implemented within the *pyforce* package, leveraging the Python programming language [6,14]. The former is part of the ROSE framework (Reduced Order multi-phySics data-drivEN), developed and maintained by the authors and available, under the MIT license, at <https://github.com/ERMETE-Lab/ROSE-pyforce>.

2.1 Generalised empirical interpolation method

The Generalised Empirical Interpolation Method is a DDROM technique, first proposed in [9]. The basis functions $\psi_n(\mathbf{x})$ and the optimal locations of sensors are

selected following a greedy procedure. More in detail, the GEIM approximates a given function u with a suitable interpolant:

$$u(\mathbf{x}, \boldsymbol{\mu}) \simeq \mathcal{I}_M[u](\mathbf{x}, \boldsymbol{\mu}) = \sum_{m=1}^M \beta_m(\boldsymbol{\mu}) \cdot q_m(\mathbf{x}) \quad (1)$$

$$\text{s.t. } \{v_m(u) = v_m(\mathcal{I}_M)\}_{m=1}^M,$$

where the magic/basis functions $\{q_m(\mathbf{x})\}_{m=1}^M$ embed the spatial behaviour, whereas the coefficients $\{\beta_m(\boldsymbol{\mu})\}_{m=1}^M$ contain the parametric dependence. Each magic function is associated with a magic sensor $v_m(\cdot, \mathbf{x}_m, s)$, which can be represented mathematically by a linear functional centred in $\mathbf{x}_m \in \Omega$ with a point-spread $s \in \mathbb{R}^+$ representing the area onto which the sensor collects data, to simulate the fact that, since sensors have a physical dimension, they do not collect point-wise data; the value of the point-spread depends on the sensor itself and the spatial discretisation of the FOM [16].

- In the offline phase, the greedy algorithm returns a set of magic functions and sensors by minimising the interpolation error between the interpolant \mathcal{I}_M and the training dataset. The magic sensors are selected from the library \mathcal{Y} , representing the set of available locations in the domain.
- In the online phase, measurements are collected from the magic sensors:

$$y_m = v_m(u_{true}(\mathbf{x}), \mathbf{x}_m, s) + \epsilon_m \quad (2)$$

$$\text{with } m = 1, \dots, M,$$

with $\epsilon_m \sim \mathcal{N}(0, \sigma^2)$ being uncorrelated Gaussian random noise and σ^2 its standard deviation, also called the noise level. The reduced coefficients $\boldsymbol{\beta} \in \mathbb{R}^M$ can, then, be determined by solving the linear system (having dimension $M \ll \mathcal{N}_h$) resulting from the interpolation condition equation 1. To avoid having an unbounded error due to noisy data [26], this work adopts the Tikhonov-regularised version of the GEIM, henceforth

called TR-GEIM, developed by the authors in [16]; this version weakens the interpolation condition by adding a penalty term λ , and the equivalent linear system of dimension M^2 to be solved becomes:

$$(\mathbb{B}^T \mathbb{B} + \lambda \mathbb{T}^T \mathbb{T}) \boldsymbol{\beta} = \mathbb{B}^T \mathbf{y} + \lambda \mathbb{T}^T \mathbb{T} \bar{\boldsymbol{\beta}}, \quad (3)$$

given $\mathbb{B}_{ij} = v_i(q_j)$, \mathbb{T} the regularisation matrix, which depends on the standard deviation of the training coefficients $\boldsymbol{\beta}$ and λ the regularisation parameter, whose optimal value is σ^2 in the optimal case of unconstrained sensors [14].

For more details on the mathematical implementation of the TR-GEIM algorithm, interested readers can refer to [16].

2.2 Parameterised-background data-weak formulation

Among all reduction techniques that are built upon the Reduced Basis framework, the PBDW (Parameterised Background Data-Weak) [8] offer a general formulation for coupling with additional data. Derived from the general DA problem statement [5,27], the PBDW algorithm aims at approximating the state $u(\mathbf{x}, \boldsymbol{\mu})$ through the linear combination of the available sources of information, namely the physical knowledge from the mathematical model z_N and the information from the data η_M :

$$\begin{aligned} u(\mathbf{x}, \boldsymbol{\mu}) &\simeq z_N(\mathbf{x}, \boldsymbol{\mu}) + \eta_M(\mathbf{x}, \boldsymbol{\mu}) \\ &= \sum_{n=1}^N \alpha_n(\boldsymbol{\mu}) \cdot \xi_n(\mathbf{x}) + \sum_{m=1}^M \theta_m(\boldsymbol{\mu}) \cdot g_m(\mathbf{x}), \end{aligned} \quad (4)$$

where $\{\xi_n\}_{n=1}^N$ is the basis of the N -dimensional reduced space spanned by the mathematical model and $\{\alpha_n\}_{n=1}^N$ is the associated weight coefficient, whereas $\{\theta_n\}_{n=1}^M$ is the basis of the M -dimensional update space obtained from the data, with $\{g\}_{n=1}^M$ its weight coefficients.

- In the offline phase, to build the reduced space of the mathematical model, any Reduced Basis techniques can be used: in particular, the present work uses the Proper Orthogonal Decomposition (POD) [1]; for what concerns the update space, which typically refers to the experimental data, a sGREEDY procedure [25,27,28] is used, with the overall goal of minimising the reconstruction error by selecting the optimal positioning of the available sensors $\{v_m\}_{m=1}^M$. In particular, the basis functions g_m of the update space are the Riesz representation of the linear functional v_m [6,8,25].
- In the online phase, the weight coefficients $\boldsymbol{\alpha} \in \mathbb{R}^N$ and $\boldsymbol{\theta} \in \mathbb{R}^M$ are computed by solving the following $(N + M)^2$ linear system:

$$\begin{bmatrix} \chi M \mathbb{I} + \mathbb{A} & \mathbb{K} \\ \mathbb{K}^T & 0 \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\theta} \end{bmatrix} = \begin{bmatrix} \mathbf{y} \\ 0 \end{bmatrix}, \quad (5)$$

given $\mathbb{A}_{m,m'} = (g_m, g_{m'})_{L^2(\Omega)}$, $\mathbb{K}_{m,n} = (g_m, \xi_n)_{L^2(\Omega)}$, and χ a hyperparameter that should be tuned using cross-validation to improve the performance of the algorithm [14], and \mathbb{I} being the identity matrix.

The PBDW algorithm is generally stable in the presence of noise; therefore, it does not require additional regularisation, and the hyperparameter χ becomes a weight of the relative importance of models and measurements [28]. For more information on the algorithm and its implementation, interested readers can refer to [6,8].

3 Strategy for sensor recovery

With noisy measurements as input data, the TR-GEIM and the PBDW formulation are reliable and robust in performing state reconstruction and state estimation, given a correct tuning of the hyperparameters λ and χ . However, to the best of the authors' knowledge, research regarding the performance of these algorithms (and more in general, of DDROM and DA-based ROM) in presence of one or more malfunctioning sensor is still scarce [17], despite the significant importance for the deployment of such techniques in industrial settings.

As a first analysis, this work focuses on two classes of possible malfunctions, both represented by the general formulation:

$$y_m = v_m(u_{true}(\mathbf{x}), \mathbf{x}_m, s) + \epsilon_m + \mathcal{N}(\kappa, \rho). \quad (6)$$

High values of κ correspond to drifting, that is, a time-uniform shift compared to the average value; high values of ρ , conversely, indicate unexpected spikes (that is, not related to random noise). Modelling the lifetime of the sensor during the incidental transient is outside the scope of this work; instead, this work includes a preliminary analysis on faulty sensor recognition and supervised classification, discussed more detail in Section 3.1. The recovery strategy proposed in this work relies on the use of Supervised Learning (SL) methods and the premise of redundancy: the missing information about the measurement of the malfunctioning sensor is retrieved through the use of auxiliary sensors, located in 'safer' regions and trade measurement accuracy with reliability and robustness. For the case of nuclear reactors, where the in-core region is characterised by high neutron fluences and therefore the sensors are subjected to significant radiation, this implies positioning sensors outside the core, where the radiation field, and thus the probability of failure, will be lower. Therefore, the problem in principle becomes two-fold, as the algorithm must be able to infer the state of the system in one region of the domain from sensors located in another part of the domain: both TR-GEIM and PBDW are quite suited for the latter problem [13,14].

To solve the former, Supervised Learning methods can be used to learn the input-output relationship between primary sensors and auxiliary ones, that is, the map $y_k(t) = \mathcal{F}(\mathbf{y}_{ext})$; once this map is learned and a surrogate model is built, it is possible to map the signal from auxiliary sensors back to the in-core ones, thus retrieving the expected measurement despite the presence of a faulty signal. The above strategy is shown in Figure 2 for the case study of the Molten Salt Fast Reactor (MSFR) and assuming that the observable field is temperature T (although the strategy hereby discussed can be adapted

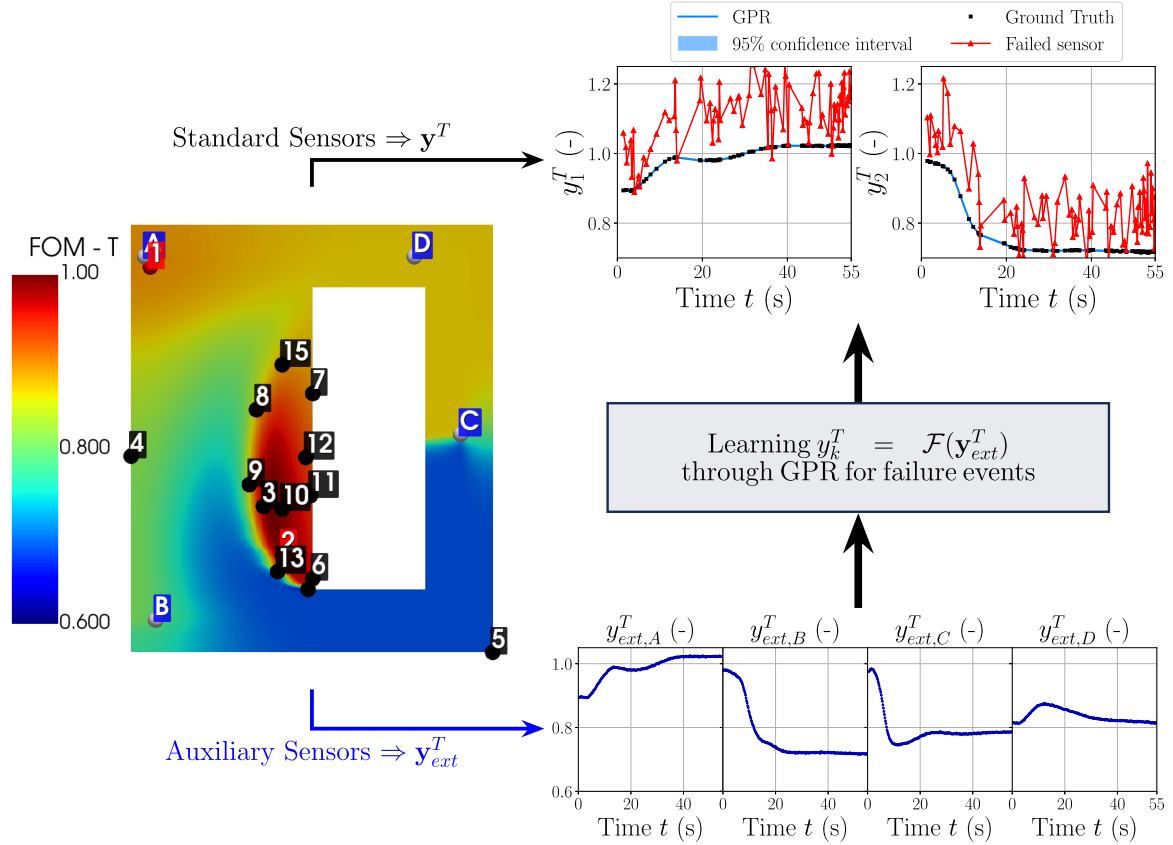


Fig. 2. Recovery of missing information of failed sensor y_k^T (in black on the left) from external sensors \mathbf{y}_{ext}^T (in blue), for the temperature field T . Taken from [21].

for any case study and any observable field of interest). In-core measurements \mathbf{y}^T are labelled from 1 to 15, and they are represented by black dots; positioning of these sensors is determined by a greedy procedure, meaning that they are ordered according to their importance. Auxiliary out-core measurements \mathbf{y}_{ext}^T are labelled from A to D and represented by blue dots. The Supervised Learning technique selected for this work is the Gaussian Process Regression (GPR) method [22], whose output replaces the missing information from the malfunctioning sensor.

Generally speaking, SL algorithms find a model f that can infer the existing relationship between input and output; in particular, the GPR infers a probabilistic distribution, used for predicting y_i in the presence of unknown input values \mathbf{x} . Then, the prior distribution of f is a Gaussian Process (GP) $f(\mathbf{x}) \sim GP(\mu(\mathbf{x}, \mathcal{K}(\mathbf{x}, \mathbf{x}')),$ with mean value $\mu(\mathbf{x})$ (typically taken as a constant function based on the training measurements) and covariance function $\mathcal{K}(\mathbf{x}, \mathbf{x}')$; this latter term is also called *kernel*, whose selection is a hyperparameter of the GPR model. This work uses the default Radial Basis Function (RBF) kernel. The kernel function depends on the hyperparameters $\boldsymbol{\vartheta}$, whose values must be tuned by maximising a log-marginal likelihood loss function [21].

GPR models include the knowledge within the training dataset \mathcal{X} into the prior distribution to obtain a posterior distribution. Given a set of training input measurements

$\mathbb{X} = [\mathbf{x}_1 | \mathbf{x}_2 | \dots | \mathbf{x}_{N_s}] \in \mathbb{R}^{d \times N_s}$, a set of corresponding outputs $\mathbf{y} = [y_1, y_2, \dots, y_{N_s}]^T \in \mathbb{R}^{N_s}$ and a set of unseen test input data $\mathbb{X}^* \in \mathbb{R}^{d \times M}$, predictions $\mathbf{y}^* \in \mathbb{R}^M$ can be made by using the theorem of conditional Gaussian, in which the conditional distribution of the predicted GP is $\mathbf{y}^* = f(\mathbb{X}^* | \mathbf{y}, \mathbb{X}, \mathbb{X}^*) \sim GP(\hat{\mu}(\mathbb{X}^*), \hat{\mathcal{K}}(\mathbb{X}, \mathbb{X}^*))$ [29].

The GPR implementation used in this paper is from the Python package GPpy (<https://github.com/SheffieldML/GPy>), which includes the hyperparameter optimisation.

3.1 Faulty sensor recognition

Compared to the preliminary work done by the authors in [21], in which the state of each sensor was known a-priori and thus it was an input to the GPR algorithm, the present work removes this hypothesis by implementing faulty sensor recognition. In particular, two different strategies have been used.

The first strategy assumes that the map $y_k(t) = \mathcal{F}(\mathbf{y}_{ext})$ has been learnt for each sensor k . Then, given a new set of measurements (retaining the assumption that only one sensor fails at time $t = 0$), the malfunctioning sensor can be recognised by comparing each trajectory $y_k(t)$ with the one predicted by the GPR. For the

comparison, the z -score has been defined as:

$$z_{score} = \frac{\|y_k(t)\|_{L^2} - \|\hat{\mu}(\mathbb{X}^*)\|_{L^2}}{\sqrt{\hat{\mathcal{K}}(\mathbb{X}, \mathbb{X}^*)}} \quad (7)$$

recalling $\mathbf{y}^* \sim GP(\hat{\mu}(\mathbb{X}^*), \hat{\mathcal{K}}(\mathbb{X}, \mathbb{X}^*))$, that is, the trajectory predicted by the GPR being described by the mean $\hat{\mu}$ and the covariance $\hat{\mathcal{K}}$. This quantity is, then, compared to a threshold, whose value must be tuned to discriminate between measurement noise and the actual faulty signal. It is worth mentioning that this strategy does not require labelled data, as it works by simply comparing the incoming signal to its expected value.

In the second strategy, instead, a supervised learning classification algorithm is used to classify sensors a-priori: this allows a more thorough training of the GPR algorithm, as now the model can discriminate between healthy and faulty sensors. The selected classifier is Random Forest (RF) [23], with output being the class of sensor failure (for the present test case, four different labels are possible: ‘Healthy’, ‘Drift’, ‘Spikes’, ‘Both’). The implementation used here is from the *sklearn* Python package [30]. It is worth mentioning that in actual applications with real-world sensors, labelled data may be unavailable, and the RF classifier may be substituted, for example, with a simple binary classifier, only discriminating between ‘Healthy’ and ‘Faulty’ sensors, which will be enough for the algorithm.

4 Numerical results

This work considers the Unprotected Loss of Fuel Flow (ULOFF) accidental transient scenario for the MSFR, using the 2D axisymmetric wedge of the EVOL geometry [24] with an external Hastelloy reflector layer [31] as in Figure 3.

For sensor placement, main sensors (used as input to the DDRM) can be located freely within the domain, whereas auxiliary sensors (aimed at recovering the information from failed sensors) can be located only in the solid reflector region, where the neutron fluence is lower. The simulation also includes a momentum source, representing the primary loop pump and a heat sink, the primary-to-intermediate loop heat exchanger. The model is solved within the OpenFOAM environment using the custom-made *msfrPimpleFoam* coupled solver, developed by Politecnico di Milano [32] and adapted by the authors to include the solid reflector layer [31]: in particular, the modified solver imposes the continuity of the variables at the interface between regions, whereas at the external reflector, wall vacuum and adiabatic conditions are imposed, respectively, on the neutron fluxes and temperature fields.

The FOM dataset then includes 275 snapshots, with time domain [0:0.2:55] seconds. This dataset Ξ is split into three subsets:

- train set Ξ_{train} , including 75% of the starting dataset up to time instant $t_i = 250$ s;

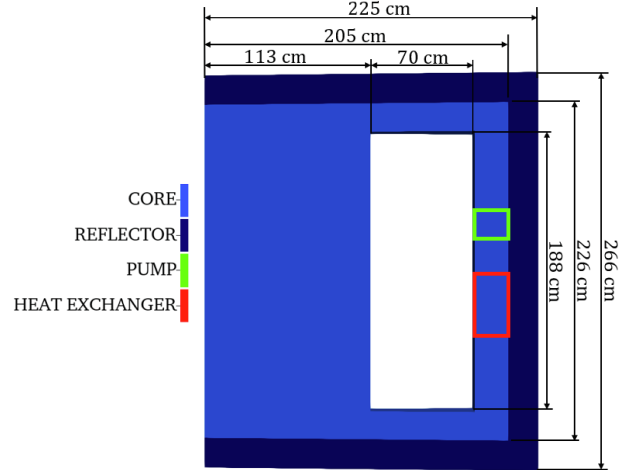


Fig. 3. EVOL geometry. The dark blue external layer represents the Hastelloy solid reflector, whereas the lighter blue domain is the primary loop containing the liquid fuel. The location of the pump and heat exchanger (green and red, respectively) is also reported.

- test set Ξ_{test} , including 25% of the starting dataset up to time instant $t_i = 250$ s, used for cross-validation;
- predict set $\Xi_{predict}$, which includes the last snapshots from $t_i = 250$ s to $t_M = 275$ s, to analyse the forecast capabilities of the reconstruction algorithm, and includes 25 snapshots.

The train-test split was done randomly using the Python package *scikit-learn*. To ensure training stability and to improve the performance of the algorithms, given that different fields have quite different orders of magnitude, all FOM snapshots were normalised to the maximum value of the initial condition: given a generic field u :

$$u(\mathbf{x}, t) \leftarrow u(\mathbf{x}, t) \cdot \left[\max_{\mathbf{x} \in \Omega} u(\mathbf{x}, 0) \right]^{-1}. \quad (8)$$

Sensors are described by linear functionals with a Gaussian kernel centred in the sensor position and with point-spread $s = 0.025$ [14]. Normalised measurements were synthetically generated during the online phase of TR-GEIM and PBDW using the sensors selected by (2), and corrupted by Gaussian white noise with variance $\sigma^2 = 10^{-3}$.

4.1 Performance of TR-GEIM and PBDW with malfunctioning sensors

The performance of TR-GEIM and PBDW algorithms in the presence of a malfunctioning sensor and without implementing any recovery strategy will be analysed first. Assuming that a single sensor fails at time $t = 0$, its output y_k will be corrupted according to (6), using $\kappa = 0.1$ and $\rho = 0.5$ to simulate the generic malfunctioning scenario of a fixed drift from the truth value and oscillations caused by an increase in noise due, for example, to electrical contacts (Fig. 4).

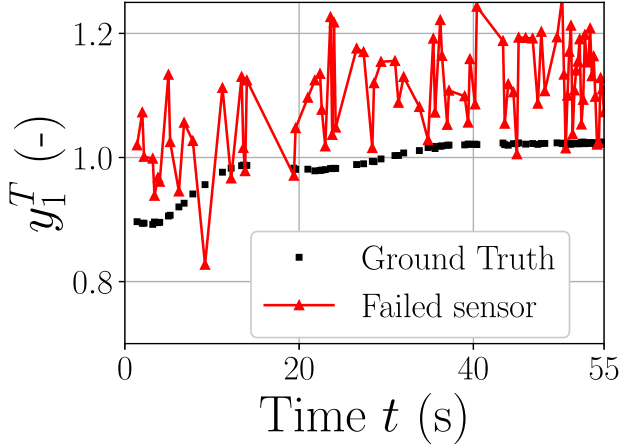


Fig. 4. Example of malfunctioning sensors with both a drift from the true value ($\kappa = 0.1$) and spikes due to unwanted noise ($\rho = 0.5$).

To measure the performance of the two DDROM algorithms, the absolute reconstruction error is used, given $r[u(\cdot, t)](\mathbf{x}) = |u(\cdot, t) - \mathcal{R}_M[u(\cdot, t)]|$ as the residual field:

$$E^k[u] = \frac{1}{\dim(\Xi^*)} \sum_{t \in \Xi^*} \|r(\mathbf{x}, t)\|_{L^2(\Omega)}, \quad (9)$$

where $M = 15^3$ is the number of sensors for the field u , chosen during the offline phase by measuring the reducibility of the problem [1], $\mathcal{R}_M[u(\cdot, t)]$ is the reconstruction operator for TR-GEIM, E^k is the average absolute error measured in L^2 -norm assuming that the k^{th} has failed, and given $\Xi^* \subset \mathcal{D}$ a subset of the parameter space with unseen data (either test or predict sets).

Figure 5 shows the reconstruction error for the two algorithms when a single sensor k fails, compared against the reference case of no malfunctioning sensors (which serves as a best-case scenario for the error). As expected, the error increases for both algorithms, meaning that the reconstruction is becoming less and less accurate: in particular, PBDW shows a larger deterioration in performance, with the error increasing by three orders of magnitude. Generally, the errors increase compared to the reference case for the sensor k depends on the variability of the measurement itself: sensors that measure less variability will influence less the performance of the algorithms, whereas sensors which record more variability will influence more in reconstruction. Still, regardless of their impact, a single failed sensor is enough to significantly worsen the reconstruction in the whole domain, as can be seen in Figure 6 for temperature and, for sake of brevity, the TR-GEIM algorithm only, with PBDW showing a similar behaviour (for the scenario $\kappa = 0.1$ and $\rho = 0.5$, and failed sensor #4): reconstruction is visibly worse in

³ The optimal number of sensors can be retrieved by looking at the inf-sup constant $\beta_{N,M}$: for a reduced space with dimension $N = 10$, selecting $M = 15$ offers a good compromise between computational cost of the greedy procedure and additional amount of information.

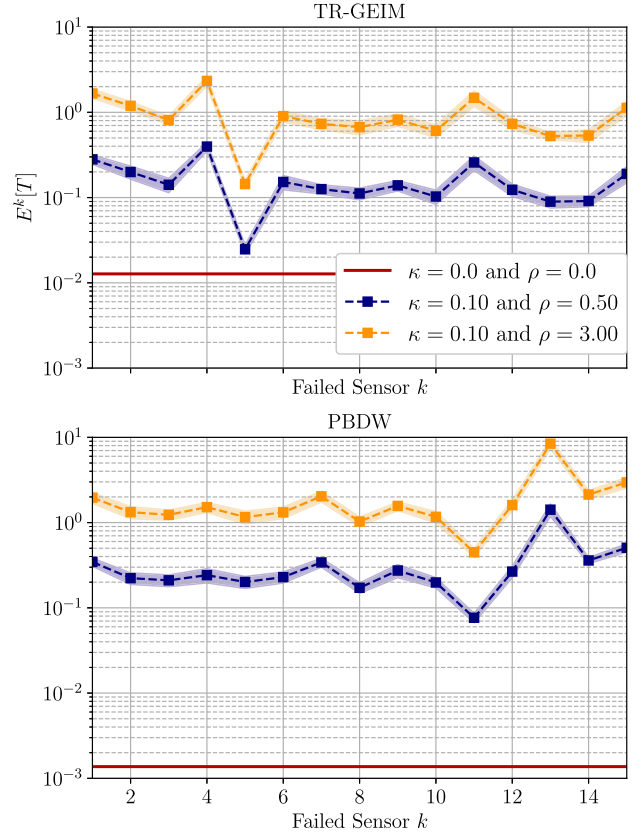


Fig. 5. Reconstruction error (computed using Eq. 9 for the most general case of both κ and ρ different from zero. The red line reports the error for the reference case without malfunctioning sensors, serving as a lower bound. The shaded area indicates the uncertainty band of the error (each scenario is repeated 20 times randomly sampling the measurement noise, to obtain statistical relevancy).

the entire domain, not only in the region near the failed sensor.

In principle, once the failed sensor has been correctly identified, the algorithm could remove the corrupted measurements from the associated linear system. The effectiveness of this strategy depends strongly on the field under consideration and the selected DDROM algorithm. Indeed, for TR-GEIM, deleting the k^{th} observation means deleting also the associated k^{th} basis function: since TR-GEIM adopts a greedy approach to select sensors in a hierarchical manner, the loss of information is inversely proportional to the index of the sensor [26]. Instead, in the PBDW algorithm, sensors are used for updating the background space, and therefore their correction is more on a local level, whereas the global spatial behaviour is given by the numerical model: sensors are still selected hierarchically, but in the update space only. For PBDW, issues may arise if more than one sensor fails, such that $M < N$: when the update space is larger than the background space, the algorithm may become unstable, losing its good convergence properties [8]. Such behaviour can be seen in Figure 7.

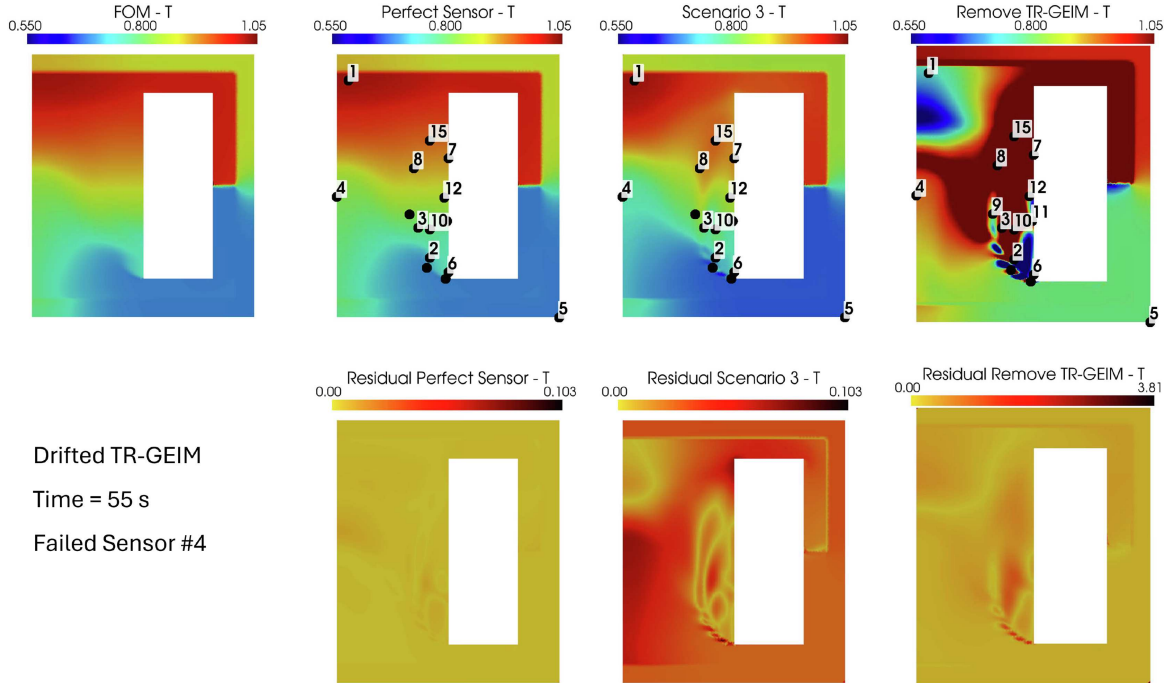


Fig. 6. Absolute error in the domain for the reconstruction of the temperature field, assuming malfunctioning sensor #4.

The hierarchy of sensor importance is clear for TR-GEIM and temperature, with the error improving as sensors with higher index are removed; conversely, PBDW is quite robust against a single sensor failure, with only a slight worsening of the error. Interestingly, the temperature and neutron flux errors show a markedly different behaviour in the TR-GEIM algorithm, with the former being much more affected by the removal of high index sensors. Indeed, whereas for temperature good error behaviour is recovered only after $k = 9$, for neutron flux it is recovered starting from $k = 2$.

To explain this behaviour, it is possible to look at the complexity of the two fields, which is linked to their reducibility: more complex fields will be less reducible, meaning that a larger number of basis will be needed to properly capture its fundamental spatial behaviour: a way to measure how much a given field is reducible is by decomposing its snapshot matrix through Singular Value Decomposition and plot the obtained singular values (see Appendix A). Figure 8 reports the normalised first forty singular values λ_i for the neutron flux and the temperature: the singular values for the former decay much faster compared to those for the latter, indicating that less basis functions are needed to retain all the relevant information regarding the neutron flux (in particular, most information is retained by the first basis function): therefore, for this field the only critical sensor from the point of view of removal is the very first one.

4.2 Faulty sensor recognition

As seen in the previous section, simply removing the malfunctioning sensor may not be enough to recover the good

performance of the DDROM algorithm for all fields of interest, especially for TR-GEIM in which sensors are hierarchical: thus, in the following section the recovery strategy described in Section 3 is implemented and used on the discussed test case. Then, given the k^{th} sensor which fails at time $t = 0$, GPR learns the input-output relationship between the external auxiliary measurements \mathbf{y}_{ext} and the output of the malfunctioning sensor y_k : from the mathematical standpoint, the prediction obtained with the GPR fills the missing term in the measurement vector \mathbf{y} appearing in the RHS of TR-GEIM and PBDW linear systems, respectively equations 3 and 5.

Whereas in [21] the index k of the failed sensor was known a-priori and was given as a fixed input to the trained GPR to recover the missing information, in this work, two different strategies have been implemented to recognise the malfunctioning data. The most straightforward method can be seen as a a-posteriori identification: assuming that a new set of measurements comes at $t = 0$ from the system and that the GPR has been thoroughly and correctly trained, it leverages equation 7 to compare each new trajectory to the expected output obtained from the GPR: if the z -score for the k^{th} sensor is below a certain threshold, the k^{th} is considered failed, and its trajectory is substituted with the one predicted by the GPR in the measurement vector that serves as an input to the DDROM.

The performance of this method is strongly dependent on the selection of the threshold value z_{th} and on the disturbance level. Figure 9 reports (not on scale) the True Positive Rate (recall) for different values of z_{th} and of the disturbance magnitude (computed as $\kappa + \epsilon$ for visualisation purposes). The recall metric is computed as the ratio between the number of true positives (the faulty sensor

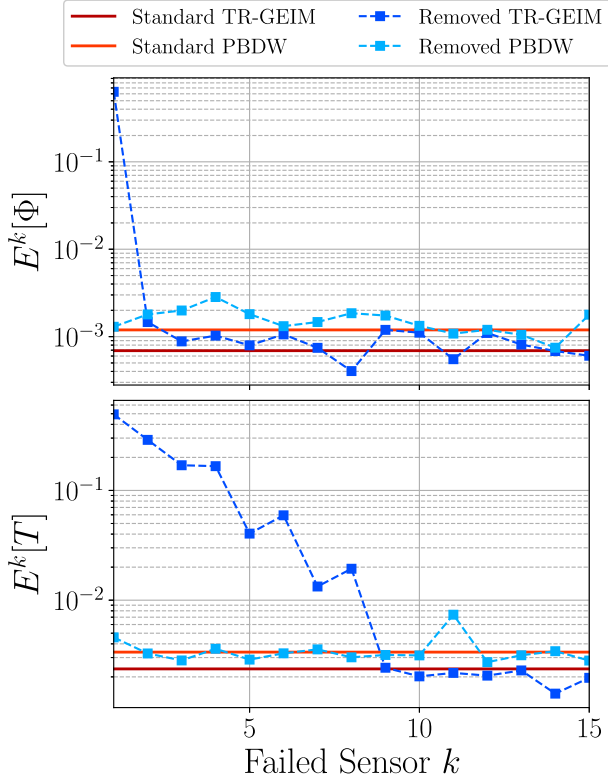


Fig. 7. Absolute error in the domain for the reconstruction of the temperature field when the (single) malfunctioning sensor k^{th} is completely removed from the algorithms' input.

is correctly identified) and the sum of true positives and false negatives (the faulty sensor is missed), and it is the key metric for safety-critical applications, that is, when missed faults must be minimised. As seen in the Figure, the only region in which the recall is not 1 is the bottom right one, corresponding to low fault magnitudes and high threshold: for low values of fault magnitude, the recognition algorithm tends to confound them with measurement noise, thus being unable to correctly detect the presence of a malfunction.

To better understand why, instead, the bottom left region (low fault magnitudes and low thresholds) show high recall values, it is useful to check the value of the precision metric, defined as the ratio between true positives and the sum of true positives and false positives (healthy sensors flagged as faulty): high precision values means few false alarms, and this metric should be preferred when false positives are costly from the point of view of maintenance. Figure 10 shows the precision heatmap. Indeed, for low threshold the precision decreases, meaning that more healthy sensors are wrongfully flagged as 'faulty': for low thresholds, then, the algorithm tend to consider even the measurement noise as due to a malfunction, so that regardless of the fault magnitude, all noisy sensors are considered as faulty, giving rise to many false positives but also correctly flagging the faulty sensor.

Thus, the z -score-based algorithm is correctly able to identify the malfunctioning sensor, provided that the fault

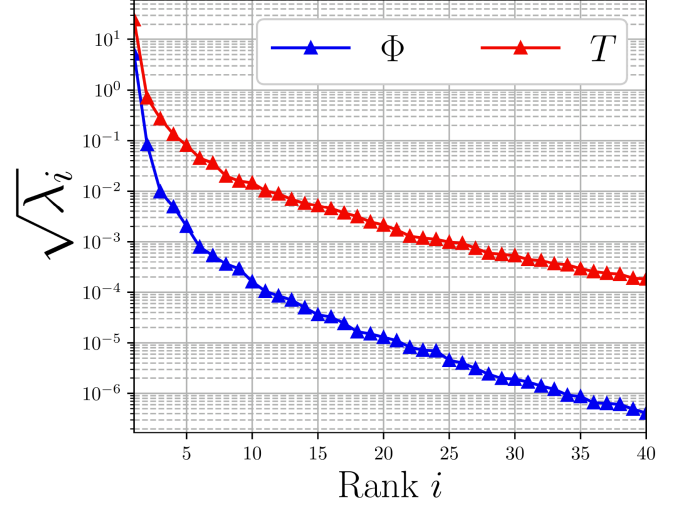


Fig. 8. SVD singular values for the neutron flux and the temperature: the faster they decay, the more the field is reducible, and a few basis can adequately describe it.

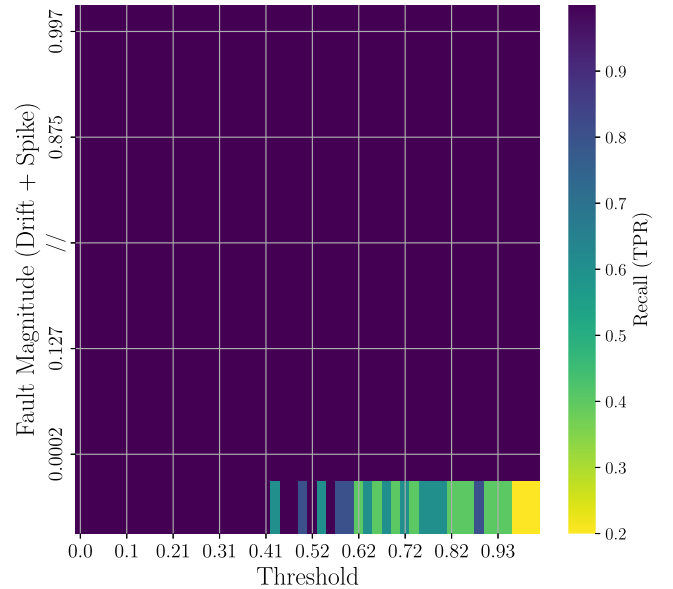


Fig. 9. Recall heatmap, reporting the ratio between true positives and the sum between true positives and false negatives. High recall values means that faults are rarely missed.

magnitude is larger than the measurement noise magnitude: the overall (i.e., considering all threshold values and fault magnitudes) recall metric is equal to 0.99, whereas the overall precision is equal to 0.9 (in both cases, values range between 0 and 1).

4.2.1 Random forest classifier

Despite its good performance, the above algorithm has two drawbacks: it requires an accurately trained GPR model for each sensor and cannot recognise the type of malfunction. Therefore, a companion approach based on a Random Forest classification algorithm is implemented. The goal of this algorithm is twofold: 1) classify the

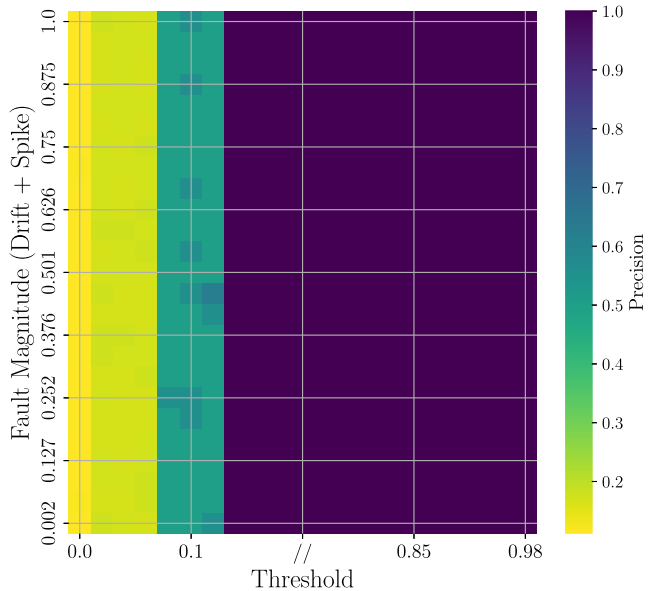


Fig. 10. Precision heatmap, reporting the ratio between true positives and the sum between true positives and false positives. High recall values means that there are few false alarms.

malfunction and 2) feed the GPR classified sensor data to improve the training and testing performance, by providing the GPR a-priori information on the shape of possible sensor failure. For the latter, if the state of each sensor is not known a priori, one GPR model for each sensor must be trained and optimised: instead, if the state of each sensor (either healthy or faulty) is known a-priori, it is possible to select the GPR model to be optimised and later used to recover the missing information; conversely, having information about the type of malfunction can help in the maintenance process.

For training the classifier, a labelled dataset $\Xi_{r,f} \in \mathbb{R}^{N_t \times 2}$ has been created, with N_t being the size of the dataset (equal to 5000); labels can be 0 (“Healthy”), 1 (“Drift”), 2 (“Spikes”) and 3 (“Both”), and this dataset has been divided into train (80%) and test (20%) sets. Following training, the classification report on the test set is given in Table 1: this report gives, for each class, the precision, recall and F1-score metrics, computed as:

$$\text{F1-Score} = 2 \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \quad (10)$$

as well as the weighted average metrics. The classifier offers excellent performance on the ‘Healthy’ and ‘Spikes’ classes; for the ‘Drift’ class, almost all drift malfunctions are caught (high recall), but given the lower precision metrics, some ‘Both’ cases are misclassified as drift. Indeed, the ‘Both’ class is the weakest one, having a recall of 0.83 (meaning that 17% of ‘Both’ malfunctions are either missed or mislabelled): this is likely due to an overlap of the feature space with the ‘Drift’ or ‘Spikes’ classes, given that the dataset has been split in a balanced way to have almost the same number of samples for all four classes.

To better breakdown the performance of the classifier, the confusion matrix is reported in Figure 11.

Table 1. Random Forest classification report on the test set for the different classes, and weighted (over the number of samples in each class) average metrics.

	Precision	Recall	F1-Score	Samples
Healthy	0.98	1.00	0.99	262
Drift	0.89	0.98	0.93	243
Spikes	0.94	0.97	0.96	241
Both	0.97	0.83	0.90	254
Accuracy			0.94	1000
Average	0.95	0.95	0.94	1000

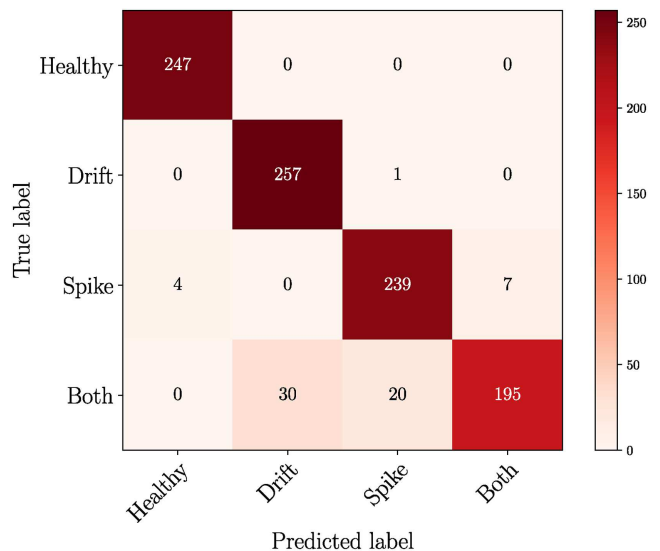


Fig. 11. Confusion matrix for the test set.

Focusing on the “Both” class, only 195 out of 245 malfunctions were correctly identified, with 30 misclassifications as “Drift” and 20 misclassifications as “Spikes”, thus explaining the lower recall metric. This suggests that the classifier struggles somewhat to distinguish compound malfunctions from individual ones, likely due to feature overlap, as even though oversampling “Both” malfunctions, no significant improvement in the classifier performance was observed. Regardless, as the classifier can correctly detect malfunctioning sensors from healthy ones, the performance is considered satisfactory.

4.3 GPR-based sensor recovery

In the following, two different recovery strategies have been considered:

1. the failed sensor is removed from the system and the corresponding reduced coefficient is reconstructed using the others: the ML model recovers the failed coefficient from the estimation of the non-failed ones.
2. The GPR algorithm directly reconstructs the missing measurements by using the information from the

auxiliary sensors: the linear system for both TR-GEIM and PBDW retains the starting reduced dimension.

It is worth mentioning that, for the purpose of faulty sensor recovery, the information on the type of failure is not used: rather, the output of both faulty sensor recognition techniques is the index of the faulty sensor, to be fed to the recovery procedure.

4.3.1 Failed sensor removal and coefficient recovery

In the first proposed recovery approach, the k^{th} failed sensor is removed from the input vector of the measurements, and GPR is used to recover its reduced coefficients β_k starting from the training dataset. This strategy is tested only for the TR-GEIM algorithm, since it is the most affected by the removal of the failed sensor⁴, as seen in Section 4.1: the input vector will, therefore, have dimension $M - 1$, and the missing coefficients are computed a posteriori using GPR. Other possibilities can actually be investigating, such as the use of the magic functions and magic sensors as basis for the PBDW and then remove the failed one, since this algorithm is less sensitive to the hierarchy of sensors. Moreover, regularisation strategies and variants of EIM-like methods [33] can be adopted: this will be matter of future analysis.

The absolute reconstruction error under this recovery approach is shown in Figure 12; the reconstructed coefficients are shown in Figure 13; finally, the residual field for temperature and failed sensor $k = 2$, for the three cases (perfect sensors, removed sensors, recovered coefficients) are shown in Figure 14. Globally, reconstruction is quite good, albeit somewhat noisy, except for coefficients with behaviour similar to β_9^{Φ} , which show small oscillations around a constant value. Still, given the hierarchical nature of TR-GEIM, these high-index coefficients have significantly less impact on the performance. Regarding the absolute error, there is an improvement of the performance, especially noticeable with temperature, for $k > 1$: interestingly enough, this improvement is not monotonous, especially for $k = 3$ and $k = 4$, and further studies regarding this peculiar behaviour are currently in progress. Regardless, this strategy fails when $k = 1$, hence the need for a more robust recovery strategy.

4.3.2 Failed sensor recovery

The strategy proposed in the previous section, in which the information from the malfunctioning sensor is completely removed from the measurement input vector and the reduced coefficients about it are recovered using GPR, shows good but not great performance, especially for TR-GEIM when high-index sensors fail. Indeed, by removing an input measure, the associated basis function gets removed as well, thus impoverishing the reduced space. This unwanted behaviour is especially significant when $k = 1$, that is, the first sensor fails: for TR-GEIM, this is the one containing most of the information of the starting dataset, and, as seen in Figure 12, its loss cannot be

⁴ At the moment, the failed index is not provided by the methods presented in the previous section, rather it is provided as a-priori information.

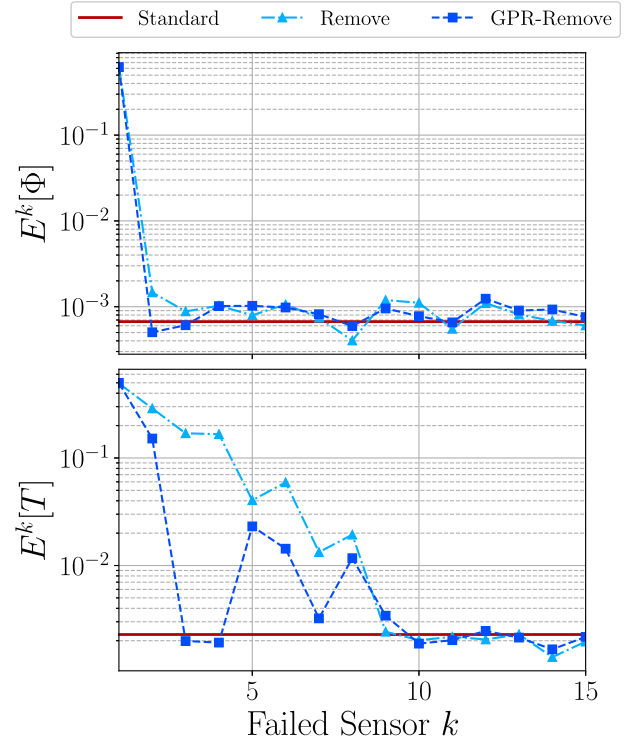


Fig. 12. Absolute reconstruction error for TR-GEIM when the failed measurement is removed from the input vector and the associated missing coefficients are estimated through GPR.

compensated by the recovery of the associated reduced coefficients through GPR. The PBDW, on the other hand, can weigh the trust in either the model or the measurements through the hyperparameter χ and hence, for failed sensors, the model can be considered more reliable and produce a state estimation based on the background information [25,27].

Therefore, a second recovery strategy is hereby proposed: the k^{th} failed sensor is not removed from the measurement input vector, but rather, it is substituted by the output provided by the GPR, obtained using the auxiliary sensors located in ‘safer’ regions of the domain (in the case of the MSFR, the solid reflector layer). The reduced linear system then maintains its starting dimension M , and the basis function associated with the failed sensor is retained. Then, the online phase proceeds as in the case of perfect sensors.

Figure 15 shows the reconstructed measurement for sensor $k = 1$, for both temperature and neutron flux: clearly, an adequately-trained GPR can correctly recover the missing measurements, even in presence of drifting and unphysical spikes, meaning that it is possible to substitute the GPR output into the measurements input vector and, then, proceed with the ‘standard’ online phase for DDROM. The good performance of this approach compared to the previous one can be seen in Figure 16, which shows the average absolute error, computed as:

$$E = \frac{1}{M} \sum_{k=1}^M E^k [T], \quad (11)$$

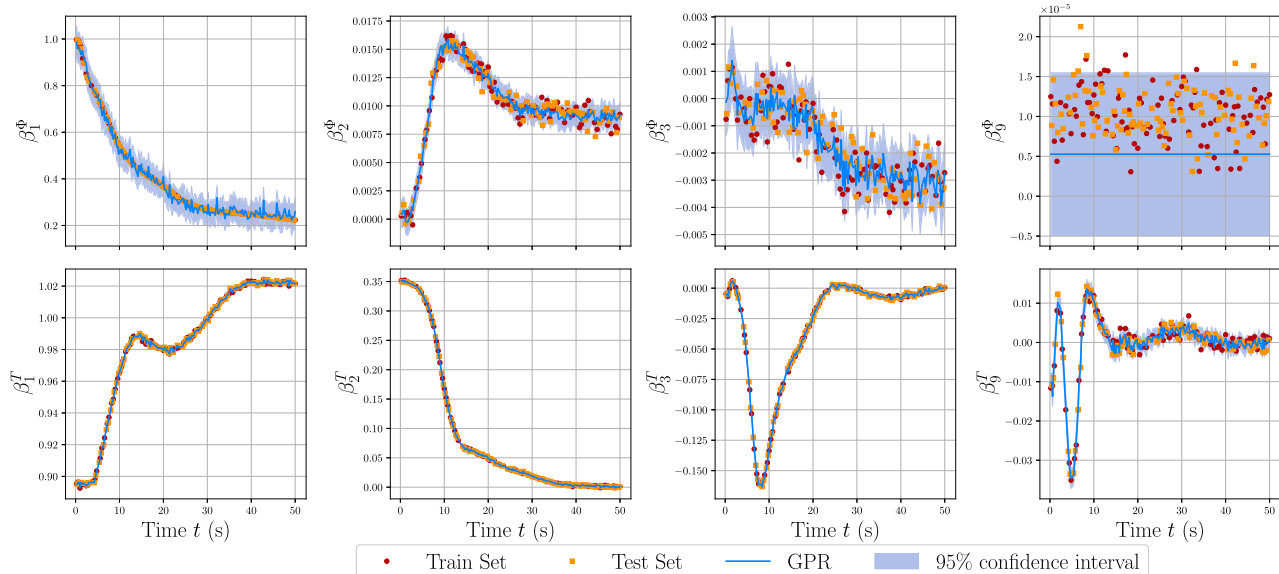


Fig. 13. Reconstructed coefficients β for TR-GEIM for four different failed sensors, both for temperature and for the neutron flux. Reconstruction is relatively good albeit somewhat noisy, except for constant or almost-constant coefficients: however, since sensors in TR-GEIM are hierarchical, high-order oscillating coefficients have less effect on the performance of the algorithm.

given E^k defined in equation 9. Only temperature is considered for brevity, as it is the field that is more influenced by malfunctioning sensors. Four different scenarios, considering a time instant outside the training dataset, have been considered: 1) perfect case, in which all sensors are correctly operating; 2) malfunctioning sensors with $\kappa = 0.1$ and $\rho = 3$; 3) scenario in which the measurement y_k^T associated with the failed sensor is removed from the measurement input vector; 4) GPR-aided case in which the input-output map learned by GPR during the training phase through auxiliary external sensors is used to recover the output of the failed sensor, for the case $\kappa = 0.1$ and $\rho = 3$. The difference in the removal strategy for TR-GEIM and PBDW is evident: generally speaking, algorithms that adopt greedy procedures to select the optimal location of sensors are sensitive to malfunctioning ones, especially for high-index sensors; thus, the improvement in performance when the offending measurement is removed is minimal.

Conversely, by recovering the missing information through GPR, TR-GEIM shows a significant improvement in the performance, and the error returns comparable to the perfect case; the fact that the GPR-aided TR-GEIM shows better performance than the truth case is related to the fact that the GPR tends to output smoother measurements compared to the original ones, thus lowering the noise introduced in the algorithm. Due to the fact that sensors are constrained to stay on the boundary the value of λ can be further tuned to achieve more accurate results: however, for the sake of simplicity and brevity it has been chosen to be equal to the variance of the noise⁵

⁵ If this information is not provided, which can be the case for real scenarios, a hyperparameter tuning algorithm should be implemented similar to [14,28].

Even locally, the improvement in performance through the recovery of the failed sensor is noticeable, as seen in Figure 17 for the worst scenario of $k = 1$ and for a time instant outside the training dataset. By recovering the missing measurements, both algorithms can correctly reconstruct the field of interest, even in forecast mode, with the performance returning comparable to that of the perfect case.

5 Conclusions

This paper has presented two Data-Driven Reduced Order Modelling methods, the Generalised Empirical Interpolation Method and the Parameterised-Background Data Weak formulation for combining high-fidelity numerical simulation and measurements collected by physical sensors, in the realistic scenario of sensor malfunction. Two different classes of malfunctions have been considered, namely a drift from the true value and non-physical spikes, under the assumptions that only one sensor can fail at a time, and that the state of the sensor is known a priori (these assumptions will be relaxed in future works). As a test case, this paper uses the Molten Salt Fast Reactor, and in particular, it considers an Unprotected Loss of Fuel Flow accidental scenario. The test case is interesting because it features a liquid fuel and a fast neutron spectrum: in the optimal case that in-core sensing would be possible for the main quantities of interest (in this work, the neutron flux and the temperature), in-core sensors will be subjected to high neutron fluences and very hot molten salt, hence their failure probability will be higher; conversely, sensors located in the solid reflector layer will be more robust, thus they can act as ‘redundant’ sensors.

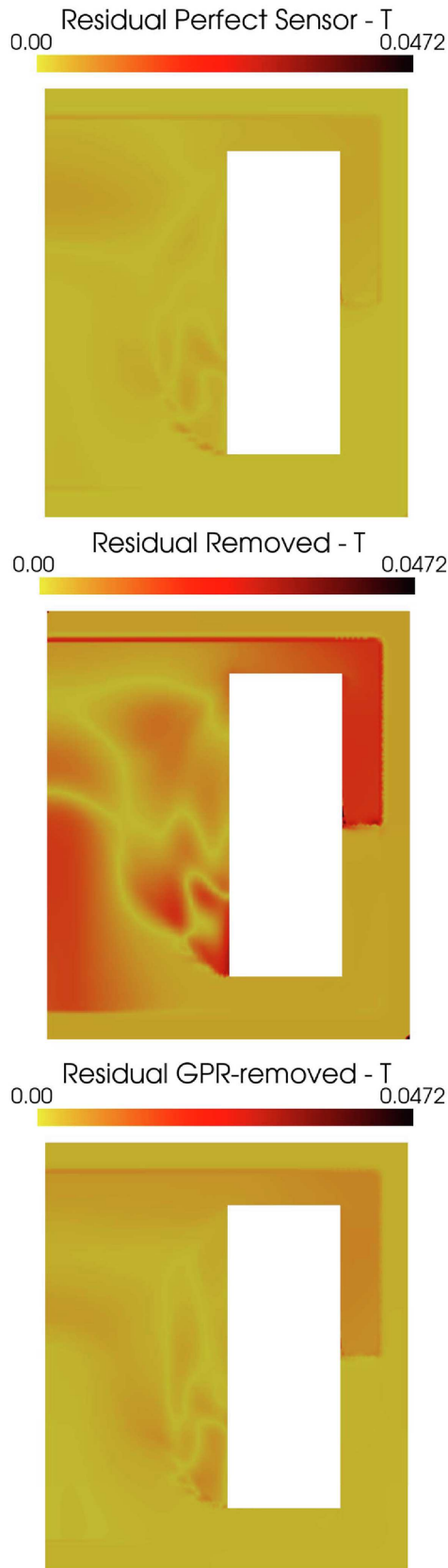


Fig. 14. Residual field for temperature in the three scenarios (perfect sensors, removed sensor and recovered coefficients), for sensor $k = 2$ at the end of the transient.

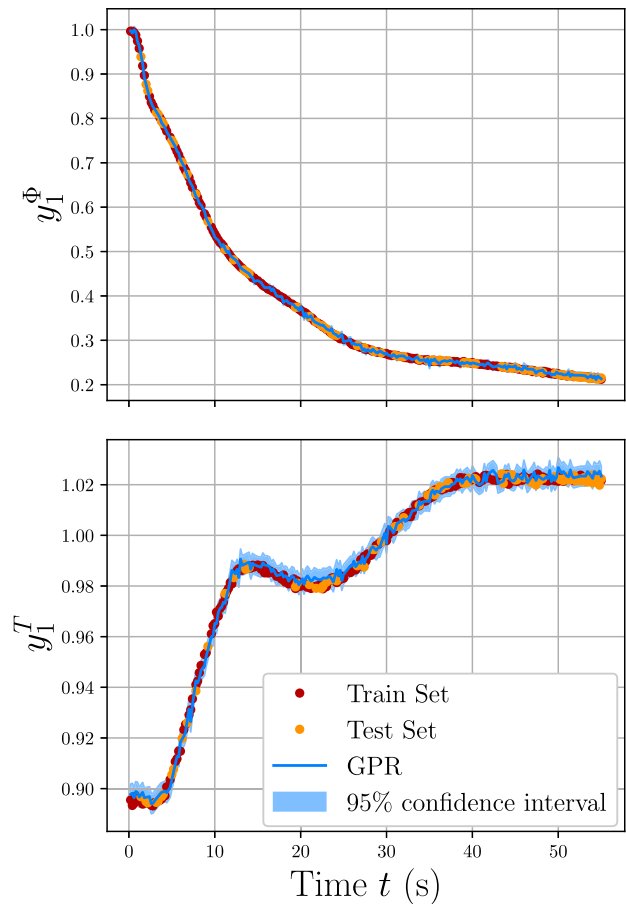


Fig. 15. Reconstructed measurements for sensor $k = 1$ (temperature and neutron flux) using GPR.

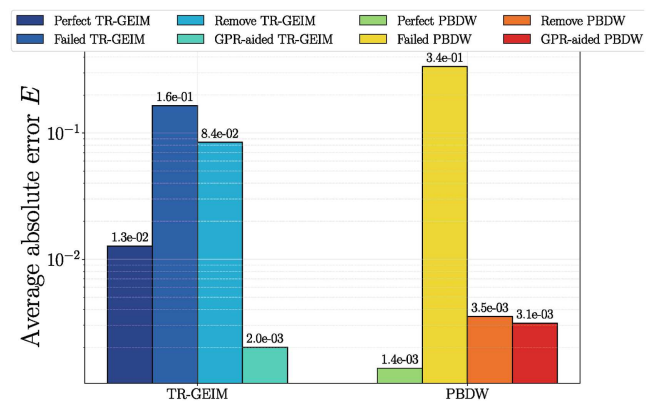


Fig. 16. Comparison of the average absolute error using different algorithms, for the following scenarios: perfect sensors, failed sensor, removed failed sensor, recovered failed sensor.

Indeed, even a single drifted measure coming from a malfunctioning sensor significantly worsens the capabilities of the two DDROM techniques, for all sensors, resulting in unbounded errors and unphysical results. The straightforward removal strategy, that is, removing the failed sensor from the measurement input vector, shows different results: whereas it seems viable for PBDW, which

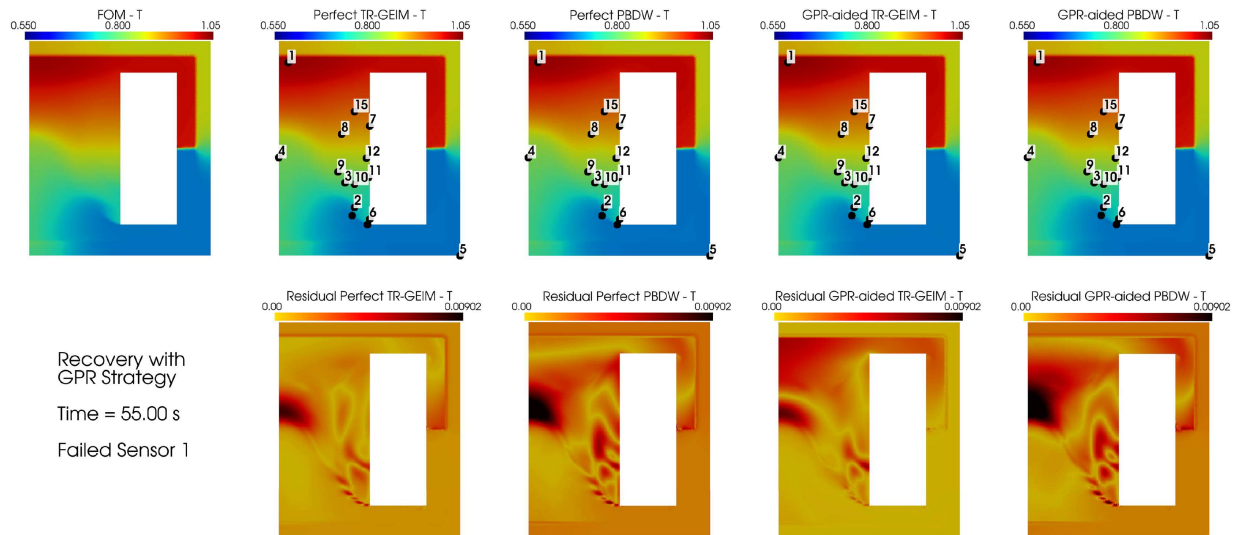


Fig. 17. Contour plots of the temperature field T for the FOM, TR-GEIM and PBDW (the latter two both in perfect and GPR-aided conditions), for failed sensor $k = 1$ at time $t^* = 55 \text{ s} \in \Xi_{\text{predict}}$.

selects sensors in a non-greedy manner, for TR-GEIM, removing an input means also removing the associated basis function and thus impoverishing the reduced space. Especially for temperature and high-index sensors, since TR-GEIM greedily selects the optimal sensor positioning, the good performance of the algorithm cannot be fully recovered this way. A study of the performance of DDROM-like techniques accounting for the realistic possibility of sensor failure is recommended, regardless of the selected algorithm.

To overcome this limitation of the DDROM framework, a Machine Learning technique, Gaussian Process Regression, has been used to try to recover good performance. The auxiliary sensors located in the ‘safer’ region of the solid reflector have been used alongside the in-core sensor to learn the existing input-output relationship between them. This map is used to 1) either recover the reduced coefficients of the removed sensor or 2) directly reconstruct the missing measurement. Between the two, the second strategy has been proved to be more successful even for the worst-case scenario of failed sensor $k = 1$ and temperature reconstruction with TR-GEIM, allowing the algorithm to correctly forecast the field of interest even for a time step outside the training dataset.

To correctly identify the failed sensor, to serve as an input to the trained GPR for the recovery step, two different strategies have been adopted. In the first one, based on the evaluation of the z -score of the incoming sensor trajectory and its comparison with the respective prediction of the GPR, good performance were observed as long as the fault magnitude can be distinguished from the measurement noise: low fault magnitude and high threshold for discriminating between healthy and malfunctioning sensors gave rise to false negatives, where faulty sensors were not correctly labelled as such; conversely, for low thresholds the number of false positives is high, meaning that the algorithm tends to consider all kind of deviations, including measurement noise, as malfunctions. Regard-

less, the overall recall score was 0.99, with a precision of 0.9.

To further study the classification problem, a Random Forest classifier was also implemented to discriminate between the types of faults and to give a-priori information to the GPR algorithm during the training phase. Performances were quite good also for the RF classifier, with an overall recall metric of 0.95. In terms of correct classification, the class with the worst performance was the ‘Both’ class (recall metric 0.83), due to misclassification of ‘Both’ malfunctions either into the ‘Drift’ or the ‘Spikes’ class: given the balance in the test set, this indicates a feature overlap, which may indicate the need of performing feature engineering on the dataset to further improve the performance of the classifier.

This work and its promising results are the first step of a methodological pathway for engineering applications where the algorithm itself can recognise the state of each sensor in time and retrieve the information from each failed sensor when needed to develop autonomous systems from the point of view of monitoring, control and diagnosis. Future studies in this direction are currently underway, including the use of unsupervised classifiers, accounting for the actual sensor failure probability and simulating the lifetime of the sensor, and accounting for the possibility of concurrent malfunctions.

Funding

This research received no external funding.

Conflicts of interest

The authors have nothing to disclose.

Data availability statement

This work adopts the *pyforce* package; the extended code for failed sensors will be made available upon completion of the review process, and it will be made available at <https://github.com/ERMETE-Lab/ROSE-pyforce.git>

Author contribution statement

Conceptualization, S.R., C.I. and A.C.; Methodology, S.R.; Software, S.R. and C.I.; Formal Analysis, S.R.; Investigation, S.R. and C.I.; Data Curation, S.R.; Writing – Original Draft Preparation, S.R. and C.I.; Writing – Review & Editing, A.C. and E.Z.; Visualization, S.R.; Supervision, A.C.

θ	PBDW update space basis function
ϑ	GPR hyperparameters
\mathbf{v}	TR-GEIM magic sensors
Υ	Library of magic sensors
$\xi(\mathbf{x})$	PBDW model basis functions
Ξ^*	Splitted dataset (train, test, predict)
χ	PBDW hyperparameter
$\psi_n(\mathbf{x})$	Generic basis functions

Glossary**Acronyms**

DA	Data Assimilation
DDROM	Data-Driven Reduced Order Modelling
FOM	Full Order Model
GEIM	Generalised Empirical Interpolation Method
GPR	Gaussian Process Regression
ML	Machine Learning
MSFR	Molten Salt Fast Reactor
PBDW	Parameterised-Background Data-Weak
POD	Proper Orthogonal Decomposition
RBF	Radial Basis Function
RF	Random Forest
ROM	Reduced Order Modelling
SL	Supervised Learning
ULOFF	Unprotected Loss of Fuel Flow

Latin Symbols

\mathcal{D}	Subset of the parameter space
E	Average absolute error in L^2 -norm
\mathcal{F}	GPR input-output map
$\mathbf{g}(\mathbf{x})$	PBDW data coefficients
$\mathcal{I}_M[u]$	TR-GEIM interpolant for the field u
\mathcal{K}	GPR kernel
M	Number of sensors
N	Dimension of the ROM
\mathcal{N}_h	Dimension of the FOM
p	Dimension of the parameter space
\mathcal{P}	log-marginal likelihood loss function
\mathbf{q}	TR-GEIM magic functions
r	Reconstruction error
\mathcal{R}_M	Reconstruction operator for TR-GEIM
s	Sensor point-spread
\mathbb{T}	TR-GEIM regularisation matrix
$u_{FOM}(\mathbf{x}, \boldsymbol{\mu})$	Full order solution (snapshot)
$\hat{u}_{DDROM}(\mathbf{x}, \boldsymbol{\mu}^*)$	Output of the ROM
\mathbf{x}	Spatial coordinate
\mathcal{X}	GPR training dataset
$\mathbf{y}(u_{true}(\mathbf{x}, \boldsymbol{\mu}^*))$	Measurement vector
$z_N(\mathbf{x}, \boldsymbol{\mu})$	PBDW model knowledge

Greek Symbols

$\boldsymbol{\alpha}(\boldsymbol{\mu}_n)$	PBDW model coefficients
$\beta(\boldsymbol{\mu})$	Modal coefficients for TR-GEIM
$\epsilon \sim \mathcal{N}(0, \sigma^2)$	Uncorrelated Gaussian random noise
κ	Measurement drift
λ	TR-GEIM regularisation parameter
$\boldsymbol{\mu}_n$	Vector of parameters
ρ	Measurement spike
$\eta_M(\mathbf{x}, \boldsymbol{\mu})$	PBDW measurement knowledge
σ^2	Measurement noise level

References

1. T. Lassila, A. Manzoni, A. Quarteroni, G. Rozza, *Model Order Reduction in Fluid Dynamics: Challenges and Perspectives* (Springer International Publishing, 2014)
2. G. Rozza et al., *Model Order Reduction: Volume 2: Snapshot-Based Methods and Algorithms* (De Gruyter, 2020). <https://doi.org/10.1515/9783110671490>
3. A. Quarteroni, A. Manzoni, F. Negri, *Reduced Basis Methods for Partial Differential Equations: An Introduction*, 1st edn., (UNITEXT, Springer Cham, 2015)
4. S.L. Brunton, J.N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control* (Cambridge University Press, 2022)
5. A. Carrassi, M. Bocquet, L. Bertino, G. Evensen, Data assimilation in the geosciences: An overview of methods, issues, and perspectives, *WIREs Climate Change* **9**, e535 (2018). <https://doi.org/10.1002/wcc.535>
6. S. Riva, C. Intorini, A. Cammi, Applied Mathematical Modelling Multi-physics model bias correction with data-driven reduced order techniques: Application to nuclear case studies, *Appl. Math. Modell.* **135**, 243 (2024). <https://doi.org/10.1016/j.apm.2024.06.040>
7. N. Baker et al., *Workshop Report on Basic Research Needs for Scientific Machine Learning: Core Technologies for Artificial Intelligence*, Tech. rep., USDOE Office of Science (SC), Washington, D.C. (United States), 2019. <https://doi.org/10.2172/1478744>
8. Y. Maday, A.T. Patera, J.D. Penn, M. Yano, A parameterized-background data-weak approach to variational data assimilation: formulation, analysis, and application to acoustics, *Int. J. Numer. Methods Eng.* **102**, 933 (2019). <https://doi.org/10.1002/nme.4747>
9. Y. Maday, O. Mula, in *A Generalized Empirical Interpolation Method: Application of Reduced Basis Techniques to Data Assimilation* (Springer, 2013), pp. 221–235
10. H. Gong, J.P. Argand, B. Bouriquet, Y. Maday, The empirical interpolation method applied to the neutron diffusion equations with parameter dependence, in *Physics of Reactors 2016, PHYSOR 2016: Unifying Theory and Experiments in the 21st Century*, 1 (May) (2016), pp. 54–63
11. J.-P. Argand, B. Bouriquet, F. de Caso, H. Gong, Y. Maday, O. Mula, Sensor placement in nuclear reactors based on the generalized empirical interpolation method, *J. Comput. Phys.* **363**, 354 (2018). <https://doi.org/10.1016/j.jcp.2018.02.050>
12. H. Gong, Data assimilation with reduced basis and noisy measurement: Applications to nuclear reactor cores, Ph.D. thesis, Sorbonne Université, 2018
13. C. Intorini, S. Riva, S. Lorenzi, S. Cavalleri, A. Cammi, Non-intrusive system state reconstruction from indirect measurements: A novel approach based on hybrid data assimilation methods, *Ann. Nucl. Energy* **182**,

- 109538 (2023). <https://doi.org/10.1016/j.anucene.2022.109538>
14. A. Cammi, S. Riva, C. Introini, L. Loi, E. Padovani, Data-driven model order reduction for sensor positioning and indirect reconstruction with noisy data: Application to a circulating fuel reactor, Nucl. Eng. Des. **421**, 113105 (2024). <https://doi.org/10.1016/j.nucengdes.2024.113105>
 15. H. Gong, Z. Chen, Q. Li, Generalized empirical interpolation method with H1 regularization: application to nuclear reactor physics, Front. Energy Res. **9**, 4018 (2022). <https://doi.org/10.3389/fenrg.2021.804018>
 16. C. Introini, S. Cavalleri, S. Lorenzi, S. Riva, A. Cammi, Stabilization of generalized empirical interpolation method (geim) in presence of noise: A novel approach based on tikhonov regularization, Comput. Methods Appl. Mech. Eng. **404**, 115773 (2023). <https://doi.org/10.1016/j.cma.2022.115773>
 17. F. Cannarile, P. Baraldi, P. Colombo, E. Zio, A novel method for sensor data validation based on the analysis of wavelet transform scalograms, Int. J. Progn. Health Manage. **9**, (2020). <https://doi.org/10.36001/ijphm.2018.v9i1.2670>
 18. V. Rao, A. Sandu, M. Ng, E.D. Nino-Ruiz, Robust data assimilation using l_1 and huber norms, SIAM J. Sci. Comput. **39**, B548 (2017). <https://doi.org/10.1137/15M1045910>
 19. B. Peherstorfer, K. Willcox, Dynamic data-driven model reduction: adapting reduced models from incomplete data, Adv. Model. Simul. Eng. Sci. **3**, 11 (2016). <https://doi.org/10.1186/s40323-016-0064-x>
 20. A. Hossein Abolmasoumi, M. Netto, L. Mili, Robust dynamic mode decomposition, IEEE Access **10**, 65473 (2022). <https://doi.org/10.1109/ACCESS.2022.3183760>
 21. S. Riva, C. Introini, E. Zio, A. Cammi, Impact of malfunctioning sensors on data-driven reduced order modelling: Application to molten salt reactors, EPJ Web Conf. **302**, 17003 (2024). <https://doi.org/10.1051/epjconf/202430217003>
 22. C.E. Rasmussen, C.K.I. Williams, Gaussian Processes for Machine Learning (MIT Press, 2006)
 23. T.K. Ho, The random subspace method for constructing decision forests, IEEE Trans. Pattern Anal. Mach. Intell. **20**, 832 (1998). <https://doi.org/10.1109/34.709601>
 24. M. Brovchenko et al., Design-related studies for the preliminary safety assessment of the molten salt fast reactor, Nucl. Sci. Eng. **175**, 329 (2013). <https://doi.org/10.13182/NSE12-70>
 25. W. Haik, Y. Maday, L. Chamoin, A real-time variational data assimilation method with data-driven model enrichment for time-dependent problems, Comput. Methods Appl. Mech. Eng. **405**, 115868 (2023). <https://doi.org/10.1016/j.cma.2022.115868>
 26. Y. Maday, O. Mula, G. Turinici, Convergence analysis of the generalized empirical interpolation method, SIAM J. Numer. Anal. **54**, 1713 (2016). <https://doi.org/10.1137/140978843>
 27. T. Taddei, Model order reduction methods for data assimilation; state estimation and structural health monitoring, Ph.D. thesis, MIT, 2016. <https://doi.org/10.13140/RG.2.2.16001.45928>
 28. Y. Maday, T. Taddei, Adaptive PBDW approach to state estimation: Noisy observations; user-defined update spaces, SIAM J. Sci. Comput. **41**, B669 (2019). <https://doi.org/10.1137/18M116544X>
 29. O.A. Martin, R. Kumar, J. Lao, *Bayesian Modeling and Computation in Python* (CRC Press, Boca Raton, 2021)
 30. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, J. Mach. Learn. Res. **12**, 2825 (2011)
 31. S. Riva, S. Deanesi, C. Introini, S. Lorenzi, A. Cammi, Neutron flux reconstruction from out-core sparse measurements using data-driven reduced order modelling, in *Proceedings of the International Conference on Physics of Reactors (PHYSOR24)*, 2024, pp. 1632–1641.
 32. M. Aufiero, Development of Advanced Simulation Tools for Circulating Fuel Nuclear Reactors, Ph.D. thesis, Politecnico di Milano, 2014. <https://doi.org/10.13140/2.1.4455.1044>
 33. F. Casenave, A. Ern, T. Lelièvre, Variants of the empirical interpolation method: Symmetric formulation, choice of norms and rectangular extension, Appl. Math. Lett. **56**, 23 (2016). <https://doi.org/10.1016/j.aml.2015.11.010>
 34. S.L. Brunton, J.N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*, 2nd edn. (Cambridge University Press, USA, 2022)

Cite this article as: Stefano Riva, Carolina Introini, Enrico Zio, Antonio Cammi. Data-driven reduced order modelling with malfunctioning sensors recovery applied to the molten salt reactor case, EPJ Nuclear Sci. Technol. 11, 55 (2025). <https://doi.org/10.1051/epjn/2025054>

Appendix A Problem reducibility and singular value decomposition

When considering a reduction problem, it is important to assess how well the high-fidelity dataset can be approximated by a reduced subspace: an estimate on the optimal size of this reduced subspace can be found through the Singular Value Decomposition (SVD) [3]: given any complex-value matrix $\mathbb{X} \in \mathbb{C}^{\mathcal{N}_h \times \mathcal{N}_s}$, the SVD provides a unique decomposition:

$$\mathbb{X} = \mathbb{U} \mathbb{\Sigma} \mathbb{V}^*, \quad (\text{A.1})$$

in which $\mathbb{U} = \{\varphi_i\}_{\mathcal{N}_h} \in \mathbb{C}^{\mathcal{N}_h \times \mathcal{N}_h}$ contains the left singular vectors of \mathbb{X} , $\mathbb{\Sigma} = [\text{diag}(\sigma_1, \dots, \sigma_{\mathcal{N}_f}, 0, \dots, 0)] \in \mathbb{R}^{\mathcal{N}_h \times \mathcal{N}_s}$ contains the singular values and $\mathbb{V}^* = \{v_i^*\}_{\mathcal{N}_s} \in \mathbb{C}^{\mathcal{N}_s \times \mathcal{N}_s}$ contains the right singular vectors of \mathbb{X} . If $\mathcal{N}_h > \mathcal{N}_s$, the diagonal matrix $\mathbb{\Sigma}$ has at most \mathcal{N}_s non-zero elements so that the *economy* SVD can be retrieved [34]:

$$\mathbb{X} = \hat{\mathbb{U}} \hat{\mathbb{\Sigma}} \mathbb{V}^*, \quad (\text{A.2})$$

In practice, the SVD provides a hierarchy of low-rank approximations, due to the fact that the singular values are ranked from the most to the least important: a simple and interpretable way of approximating \mathbb{X} , and hence the

associated snapshots, is therefore:

$$\mathbb{X} = \sum_{i=1}^{\mathcal{N}_s} \sigma_i \varphi_i v_i^*. \quad (\text{A.3})$$

Since each subsequent term is less important in capturing the most dominant features of \mathbb{X} , a good approximation can be obtained by truncating at some rank $r \ll \mathcal{N}_h$:

$$\mathbb{S} \simeq \tilde{\mathbb{U}} \tilde{\mathbb{\Sigma}} \tilde{\mathbb{V}}^* = \sum_{i=1}^r \sigma_i \varphi_i v_i^*, \quad (\text{A.4})$$

allowing for the discovery of dominant low-dimensional patterns in the data matrix \mathbb{X} . The truncated SVD basis $\tilde{\mathbb{U}}$ then provides the basis functions that span the reduced space. Moreover, the SVD comes also with another important property related to the energy captured by the truncation: the partial sum, till $k \leq r$, captures as much of *energy* of the matrix \mathbb{S} and the SVD basis represents the optimal rank- r approximation. In the end, some insights about what the columns of $\tilde{\mathbb{U}}$, i.e. the modes or basis functions, and the rows of $\tilde{\mathbb{V}}$ represent can be given. The former describes the most dominant/energetic spatial features of the data; whereas the latter embeds the parametric and temporal dependences (sometimes referred to as latent dynamics).